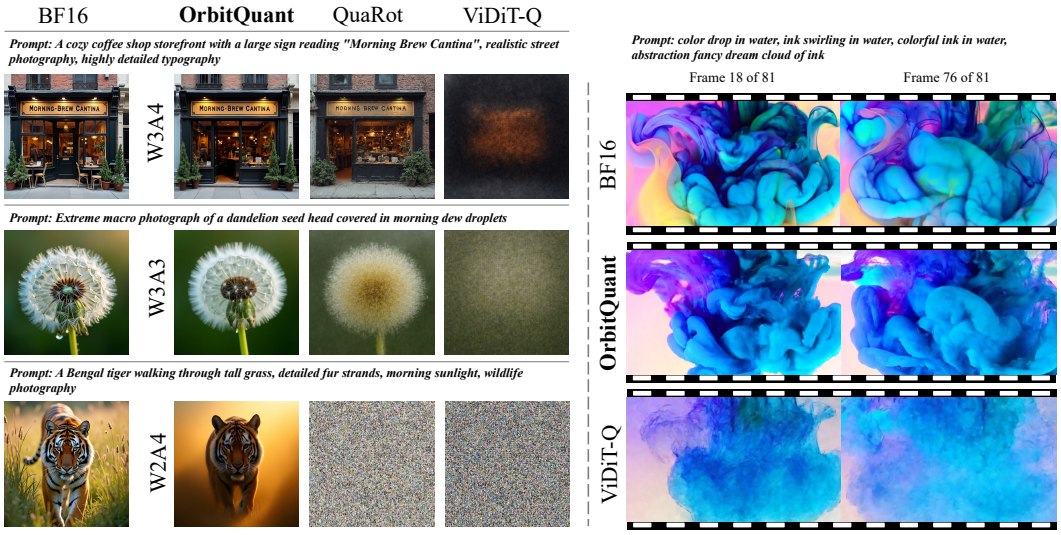
ORBITQUANT: DATA-AGNOSTIC QUANTIZATION FOR IMAGE AND VIDEO DIFFUSION TRANSFORMERS

Anonymous authors
Paper under double-blind review



(a) Image generation on FLUX.1-dev (b) W4A4 Video generation on Wan 14B

Figure 1: Qualitative comparison of OrbitQuant against QuaRot (Ashkboos et al., 2024) and ViDiT-Q (Zhao et al., 2024) under low-bit quantization, with BF16 full-precision output. (a) Image generation on FLUX.1-dev at W3A4, W3A3, and W2A4. (b) Video generation on Wan 14B at W4A4.

ABSTRACT

Diffusion transformers (DiTs) achieve state-of-the-art image and video generation, but their multi-step sampling and growing parameter count make inference expensive. Post-training quantization (PTQ) is the natural remedy, yet DiT activations shift across timesteps, prompts, and guidance branches, forcing prior methods to re-fit calibration data for every new checkpoint or modality. We present OrbitQuant, a data-agnostic weight-activation quantizer that bypasses range estimation by quantizing in a normalized, rotated basis. In this basis, a randomized permuted block-Hadamard (RPBH) rotation balances coordinate variances and maps activations toward a stable, known marginal across inputs, so a single Lloyd–Max codebook serves all timesteps, prompts, and layers of a given input dimension. We extend the same quantizer to weight rows offline, absorbing the rotation into the weights so that it cancels inside each linear layer and only a forward rotation on the activations remains at runtime. The same recipe transfers from image to video without per-modality tuning. Across FLUX.1, Z-Image-Turbo, Wan 2.1, and CogVideoX, it sets the state of the art for PTQ at several low-bit settings. It also pushes PTQ of image DiTs to W2A4 with usable generation quality.

1 INTRODUCTION

Diffusion models have become the dominant paradigm for high-fidelity image and video generation. Traditionally, these models employ a convolutional U-Net as the denoising backbone (Rombach et al., 2022; Saharia et al., 2022; Singer et al., 2022; Ho et al., 2020). More recently, however, the field has shifted to transformer-based denoisers. Diffusion Transformers (DiTs) (Peebles & Xie, 2023; Bao et al., 2023) replace the U-Net with a stack of attention blocks that scales favorably with model size and data, and now underpin state-of-the-art image (Chen et al., 2024; Esser et al., 2024; Labs et al., 2025; Xie et al., 2024; Wu et al., 2025; Cai et al., 2025) and video generators (OpenAI, 2024; Yang et al., 2025c; Kong et al., 2024; Wan et al., 2025).

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.

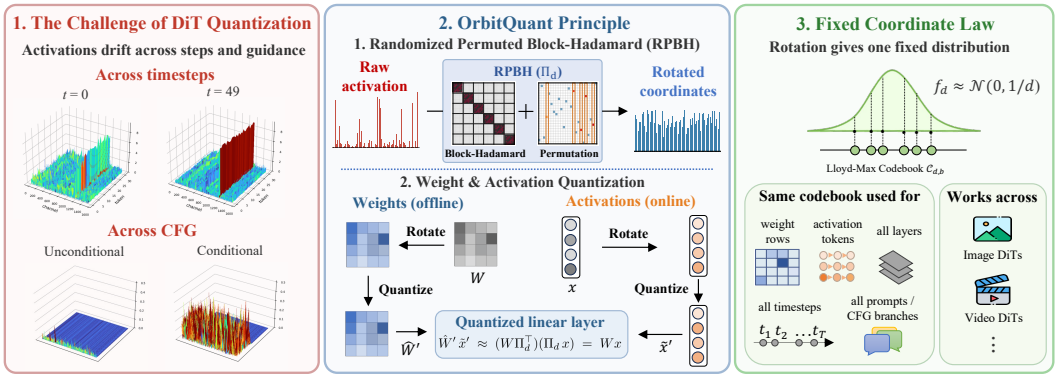


Figure 2: **Overview of OrbitQuant.** **(1)** DiT activations drift across timesteps and CFG branches, invalidating calibrated scales. **(2)** The RPBH rotation Π_d maps them to well-behaved coordinates and, once folded into the weights, cancels layerwise ($\hat{W}'\hat{x}' \approx Wx$). **(3)** Rotated coordinates approximately follow a fixed marginal $f_d \approx \mathcal{N}(0, 1/d)$, enabling one calibration-free Lloyd–Max codebook $\mathcal{C}_{d,b}$ per dimension across layers, timesteps, prompts, and image and video DiTs.

Despite their quality, DiTs are expensive to run at inference for two reasons. First, the transformer trunk is evaluated repeatedly across many sequential denoising timesteps. Second, unlike LLM decoding, where latency is dominated by weight loading (Lin et al., 2024b; Frantar et al., 2022), DiT inference is compute-bound even at a single batch, limiting the acceleration achievable with weight-only quantization (Li et al., 2024). Low-bit post-training quantization (PTQ) of both weights and activations is therefore the natural remedy, since it compresses both the memory footprint and the compute of every step without any retraining.

PTQ is most mature for large language models (LLMs), where activation outliers are handled by scaling them into the weights (Xiao et al., 2023) or using rotations to improve quantizability (Ashkboos et al., 2024; Liu et al., 2025; Chee et al., 2023; Tseng et al., 2024). However, directly transferring these techniques to DiTs is challenging: DiT activations exhibit channel-wise outliers (Wu et al., 2024) and shift across timesteps, prompts, and classifier-free guidance (CFG) branches (Chen et al., 2025; Zhao et al., 2024). Many existing DiT PTQ methods address this drift through calibration (Zhao et al., 2024; Li et al., 2024; Zhang et al., 2026), so each new checkpoint, resolution, or modality requires collecting a new calibration set and refitting the quantizer.

We propose **OrbitQuant**, a rotation-based PTQ framework for diffusion transformers. A DiT activation offers no stable range to calibrate against, since it moves with every input. Rather than repeatedly estimating input-dependent ranges or clipping thresholds, OrbitQuant rotates it away. A random rotation turns a normalized activation into coordinates that closely approximate a fixed, known distribution regardless of the input (Zandieh et al., 2025), so a single Lloyd–Max codebook built offline quantizes every activation and is shared across all denoising steps. OrbitQuant realizes this as a randomized permuted block-Hadamard (RPBH) rotation, and we find that a uniform random permutation suffices to keep the rotated marginal well-behaved at low bit-width on DiT activations. The same rotation is folded into the weight rows offline, so it cancels inside each linear layer, with weights and activations quantized in one shared basis, leaving only one forward RPBH transform per projection at inference. The main contributions of our work are as follows:

- We cast low-bit DiT activation quantization as a distributional codebook problem, replacing per-timestep range calibration with a single Lloyd–Max codebook fit to a fixed post-rotation marginal and shared across all denoising steps.
- We extend the same quantizer to the weight rows with a shared-rotation design that quantizes weights and activations in one common basis.
- We propose the RPBH rotation, an efficient rotation whose uniform random permutation keeps activations well-quantizable at low bit-width without calibration.
- We evaluate OrbitQuant on image and video DiTs, achieving state-of-the-art PTQ on GenEval and VBench without calibration data. At W2A4, where prior PTQ baselines collapse to noise, OrbitQuant is the only method producing usable images, shown in Figure 1.

2 Related Work

LLM Quantization. In LLMs, Weight-only methods quantize weights alone and suit the memory-bound decoding regime (Frantar et al., 2022; Lin et al., 2024b; Kim et al., 2023; Dettmers et al., 2023; Li et al., 2025a), while quantizing activations requires handling outlier channels, either by scaling them into the weights (Xiao et al., 2023) or by rotating them away. Rotation-based methods (Ashkboos et al., 2024; Liu et al., 2025; Chee et al., 2023; Tseng et al., 2024; Sun et al., 2024; Hu et al., 2025; Lee et al., 2026) fold a Hadamard or learned rotation into the weights by computational invariance, leaving the output unchanged while making activations easy to quantize.

Recent block rotations pair the rotation with a calibrated permutation. DuQuant (Lin et al., 2024a) orders channels by outlier magnitude, and PeRQ (Sanjeet et al., 2026) fits a permutation that balances per-block mass governing block-Hadamard outlier suppression. Random rotations also enable calibration-free vector quantization. PolarQuant (Han et al., 2026) quantizes KV embeddings in polar coordinates, building a codebook from the analytically known angle distribution after random preconditioning. TurboQuant (Zandieh et al., 2025) brings this distributional codebook to Cartesian coordinates with a dense Haar rotation and a Beta-marginal Lloyd–Max codebook. Both are standalone KV-cache vector quantizers that rotate back to reconstruct. OrbitQuant instead applies the rotation-plus-codebook idea inside DiT projections, where the shared rotation cancels rather than being inverted, and replaces the dense Haar with an efficient RPBH rotation. Unlike prior permuted rotations, it draws the permutation uniformly at random, with a probabilistic guarantee on coordinate-variance concentration.

Diffusion Quantization. Most existing DiT quantization methods are calibration-based. SVDQuant (Li et al., 2024) absorbs activation outliers with a high-precision low-rank branch fit on a calibration set, PTQ4DiT (Wu et al., 2024) balances salient channels with block reconstruction, AdaTSQ (Zhang et al., 2026) fits per-channel scales with timestep-sensitive precision allocation, and ViDiT-Q (Zhao et al., 2024) pairs per-channel calibration with mixed precision on both image and video DiTs. LRQ-DiT (Yang et al., 2025b) adds calibrated DuQuant-style (Lin et al., 2024a) rotations on outlier-heavy layers, PermuQuant (Cheng et al., 2026) calibrates channel reordering for per-group quantization, and S^2Q-VDiT (Feng et al., 2025) selects calibration data by Hessian-aware saliency with token-level distillation on video DiTs. QVGen (Huang et al., 2025b) and RobuQ (Yang et al., 2025a) depart from PTQ with quantization-aware training (QAT), the latter reaching ternary weights on ImageNet DiTs. Closer to our setting, DVD-Quant (Li et al., 2025b) is data-free, combining rotated quantization with grid refinement and adaptive bit allocation, but is specifically designed for video DiTs. ConvRot (Huang et al., 2025a) instead applies group-wise regular Hadamard rotations with uniform low-bit quantization, with experiments centered on FLUX. In contrast, OrbitQuant derives its codebook analytically from a fixed post-rotation marginal, requiring neither calibration data nor model evaluation during quantizer construction, and applies the same quantization recipe unchanged to both image and video DiTs.

3 Preliminaries

This section fixes notation and reviews the two ingredients we inherit from TurboQuant (Zandieh et al., 2025), namely a Haar-random orthogonal rotation and a Lloyd–Max scalar codebook designed against the post-rotation coordinate distribution.

3.1 Notation

We write matrices in bold uppercase (e.g., $\mathbf{W}$), vectors in bold lowercase (e.g., $\mathbf{x}$), and scalars in plain type. A DiT block is built from linear projections

$$\mathbf{y} = \mathbf{W}\mathbf{x}, \quad \mathbf{W} \in \mathbb{R}^{m\times d},\ \mathbf{x} \in \mathbb{R}^{d}, \tag{1}$$

applied token-wise to image- or text-token streams. We write $\mathbf{w}_i^\top \in \mathbb{R}^d$ for the i-th row of $\mathbf{W}$ and $r_i = \|\mathbf{w}_i\|_2$ for its ℓ_2 norm. Given weight and activation bit-widths b_w and b_a, our goal is to replace $\mathbf{W}$ and $\mathbf{x}$ with quantized surrogates $\hat{\mathbf{W}}$ and $\hat{\mathbf{x}}$ at b_w and b_a bits per coordinate, so that $\hat{\mathbf{W}}\hat{\mathbf{x}} \approx \mathbf{W}\mathbf{x}$ at every denoising step and for every prompt, without calibration data. We write $\mathcal{L}$ for the set of target linear layers and $\mathcal{D}$ for the distinct input dimensions in $\mathcal{L}$.

3.2 TurboQuant

TurboQuant (Zandieh et al., 2025) is a calibration-free vector quantizer, originally for KV-cache compression, that quantizes a vector in two steps. First, it normalizes $\mathbf{x}$ to $\tilde{\mathbf{x}} = \mathbf{x}/\|\mathbf{x}\|_2$, keeps the

norm, and applies a Haar-random orthogonal rotation $\mathbf{\Phi}_d \in \mathbb{R}^{d\times d}$ (Mezzadri, 2006). Regardless of $\mathbf{x}$, each coordinate of $\mathbf{\Phi}_d\tilde{\mathbf{x}}$ then follows the fixed marginal

$$f_d(t) = \frac{\Gamma(d/2)}{\sqrt{\pi}\,\Gamma((d-1)/2)}(1-t^2)^{(d-3)/2}, \quad t \in [-1,1], \tag{2}$$

where $\Gamma(\cdot)$ is the Gamma function. For $d \geq 64$, this marginal is tightly approximated by $\mathcal{N}(0, 1/d)$, and distinct coordinates are nearly independent. Second, since f_d is known offline, we precompute an MSE-optimal Lloyd–Max codebook (Lloyd, 1982; Max, 1960) for each $(d, b) \in \mathcal{D} \times \{b_w, b_a\}$, giving 2^b centroids $\mathcal{C}^{(d,b)} = \{c_1^{(d,b)}, \ldots, c_{2^b}^{(d,b)}\}$ and the nearest-centroid map

$$\hat{q}_b^{(d)}(t) = \arg\min_{c\,\in\,\mathcal{C}^{(d,b)}} |t - c|, \tag{3}$$

applied coordinate-wise via $\hat{Q}_b^{(d)}(\mathbf{u})_k = \hat{q}_b^{(d)}(u_k)$ for any $\mathbf{u} \in \mathbb{R}^d$. The codebook uses no scales or zero-points and is shared by all layers and rows of the same input dimension d. Dequantization looks up centroids, rotates back by $\mathbf{\Phi}_d^\top$, and rescales by the stored norm.

4 Methodology

4.1 Overview

OrbitQuant replaces per-input range calibration with a distributional quantizer applied in one shared, rotated, normalized basis. Because weights and activations are quantized in the same basis, the rotation cancels in the matrix product and only a forward rotation on the activation remains at runtime. We quantize weights offline (Section 4.2) and activations online (Section 4.3), realizing Π_d as a randomized permuted block-Hadamard (RPBH) transform with a uniform random permutation (Section 4.4). Figure 2 gives an overview.

4.2 Offline Weight Quantization

For a linear layer with input dimension d, OrbitQuant uses the shared rotation $\mathbf{\Pi}_d$ of that dimension. Before inference, we rotate the weight matrix into this basis,

$$\mathbf{W}' = \mathbf{W}\mathbf{\Pi}_d^\top. \tag{4}$$

We split each row of $\mathbf{W}'$ into a magnitude r_i' and a unit direction $\tilde{\mathbf{w}}_i'$,

$$r_i' = \|\mathbf{w}_i'\|_2, \quad \tilde{\mathbf{w}}_i' = \mathbf{w}_i'/r_i', \quad i = 1, \ldots, m. \tag{5}$$

We then quantize the direction with the Lloyd–Max codebook of Section 3.2 and re-attach the magnitude,

$$\hat{\mathbf{W}}' = \mathrm{diag}(\mathbf{r}') \cdot \hat{Q}_{b_w}^{(d)}(\tilde{\mathbf{W}}'). \tag{6}$$

Because $\mathbf{\Pi}_d$ is sampled independently of $\mathbf{w}_i$, the rotated unit directions are well mixed in the shared basis, and their coordinate distributions closely match the target density f_d in practice. We therefore quantize them using the Lloyd–Max codebook designed for f_d. The row-norm vector $\mathbf{r}' \in \mathbb{R}^m$ is stored in BF16, adding $16m$ bits per layer, negligible against the $b_w md$ bits of the quantized direction ($< 0.3\%$). The original weight is replaced by $\hat{\mathbf{W}}'$ in place, so the inference path operates entirely in the rotated basis.

4.3 Online Activation Quantization

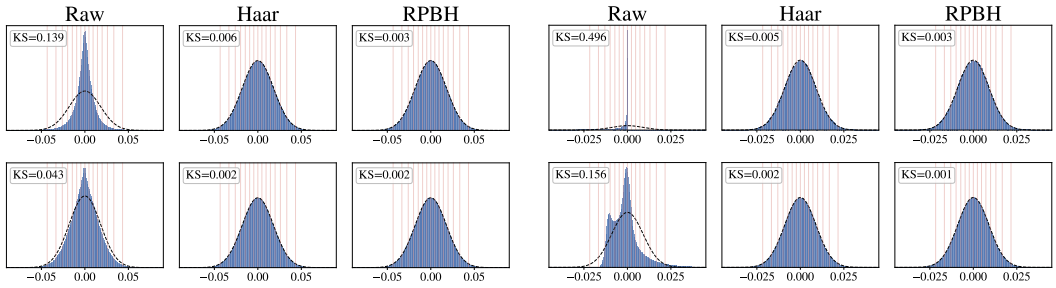
At inference, each incoming activation $\mathbf{x}$ is rotated by $\mathbf{\Pi}_d$ before it enters the layer and split into a magnitude s and a unit direction $\tilde{\mathbf{x}}'$,

$$\mathbf{x}' = \mathbf{\Pi}_d\mathbf{x}, \quad s = \|\mathbf{x}'\|_2, \quad \tilde{\mathbf{x}}' = \mathbf{x}'/(s+\varepsilon), \tag{7}$$

where $\varepsilon = 10^{-10}$ guards against zero norms on padding tokens. For a batch of N tokens, this forward rotation $\mathbf{\Pi}_d\mathbf{x}$ is applied row-wise as $\mathbf{x}\mathbf{\Pi}_d^\top$. We quantize the direction with the Lloyd–Max quantizer $\hat{Q}_{b_a}^{(d)}$ and rescale by s,

$$\hat{\mathbf{x}}' = s \cdot \hat{Q}_{b_a}^{(d)}(\tilde{\mathbf{x}}'). \tag{8}$$

As with the weights, the rotated normalized activations closely follow the target marginal f_d in practice, so the same codebook family applies without refitting. The only input-dependent quantity at inference is the per-token scalar s, while the codebook is fixed and calibration-free. Algorithm 1 collects the offline and online stages. The weight absorbs $\mathbf{\Pi}_d^\top$ and the activation applies $\mathbf{\Pi}_d$, so the two cancel in the product, $\mathbf{W}'\mathbf{x}' = \mathbf{W}\mathbf{\Pi}_d^\top\mathbf{\Pi}_d\mathbf{x} = \mathbf{W}\mathbf{x}$. The quantized layer therefore computes $\hat{\mathbf{W}}'\hat{\mathbf{x}}' \approx \mathbf{W}\mathbf{x}$ with no inverse rotation at runtime.



(a) FLUX.1-schnell $d = 3072$ (b) FLUX.1-schnell $d = 12288$

Figure 3: Rotated activation coordinates follow the dimension marginal f_d. For (a) an attention projection (d=3072) and (b) a feed-forward projection (d=12288) of FLUX.1-schnell, each cell plots the distribution of activation tokens with no rotation (Raw), a dense Haar rotation, and the RPBH. The dashed curve is the target $\mathcal{N}(0, 1/d)$ and the inset reports the Kolmogorov–Smirnov distance to it. The light red vertical ticks mark the bin edges of the shared 4-bit Lloyd–Max codebook, which is fit to f_d and used for both weights and activations in OrbitQuant.

4.4 Randomized permuted block-Hadamard

Quantizing all layers of dimension d with one codebook built from f_d works only if the rotated coordinates follow that marginal. A Haar rotation $\mathbf{\Phi}_d$ from Zandieh et al. (2025) makes them follow f_d exactly. Since the rotation cancels in the matrix product for any orthogonal $\mathbf{\Pi}_d$, we are free to choose it for efficiency, as long as it keeps the marginal close to f_d. A dense Haar rotation costs $O(d^2)$ in both time per token and storage, which dominates the per-image activation cost. We instead realize $\mathbf{\Pi}_d$ as a randomized permuted block-Hadamard (RPBH) rotation (Ailon & Chazelle, 2009; Tropp, 2011),

$$\mathbf{\Pi}_d = \mathrm{blkdiag}(\mathbf{H}_h\mathbf{D}_1, \ldots, \mathbf{H}_h\mathbf{D}_{d/h}) \cdot \mathbf{P}_\pi, \tag{9}$$

where $\mathrm{blkdiag}(\cdot)$ places its arguments as diagonal blocks, $\mathbf{H}_h$ is a $h \times h$ Walsh–Hadamard matrix, each $\mathbf{D}_i$ is a Rademacher sign diagonal, and $\mathbf{P}_\pi$ is the matrix of a uniform random permutation π drawn once per dimension. It admits an $O(d \log h)$ transform through a permutation gather and a per-block Fast Walsh–Hadamard Transform, and stores as a sign vector and a permutation array rather than a $d \times d$ matrix. The standard Walsh–Hadamard transform admits a fast $\mathcal{O}(d \log d)$ implementation when d is a power of two (Tropp, 2011). To accommodate arbitrary transformer dimensions with the same construction, RPBH instead applies power-of-two Hadamard transforms block-wise. In practice, we choose h as the largest power of two dividing d, giving $h \in \{128, 512, 1024, 2048, 4096\}$ across all evaluated models.

The leading permutation $\mathbf{P}_\pi$ acts first and keeps the marginal close to f_d at low bit-width. Without it, each block-Hadamard transform mixes only within its block, so an outlier concentrated in one block cannot spread across the others. $\mathbf{P}_\pi$ redistributes coordinates across blocks, allowing each block to receive a balanced share of the input mass with high probability over π. Crucially, this permutation need not be data-dependent. Prior methods construct data-dependent permutations using outlier magnitude (Lin et al., 2024a), column importance (Gu et al., 2026), or blockwise mass (Sanjeet et al., 2026). RPBH instead draws π uniformly at random, and the following proposition shows that this random permutation is sufficient to balance coordinate variances at the Haar scale.

Proposition 1 (RPBH coordinate-variance concentration)**.** *Let $\mathbf{\Pi}_d$ be the RPBH transform of Equation 9 on $d = kh$, and let $\tilde{\mathbf{x}} \in \mathbb{R}^d$ be a fixed unit vector with $\mu_\infty = \|\tilde{\mathbf{x}}\|_\infty^2$. For any $\delta \in (0, 1)$, define $\rho = d\mu_\infty\sqrt{(1/(2h))\log(2k/\delta)}$. Provided that $\rho < 1$, with probability at least $1 - \delta$ over the random permutation π, simultaneously for every coordinate z_i of $\mathbf{z} = \mathbf{\Pi}_d\tilde{\mathbf{x}}$,*

$$\mathbb{E}_{\mathbf{D}}[z_i \mid \pi] = 0, \qquad \mathrm{Var}_{\mathbf{D}}(z_i \mid \pi) \in \left[\frac{1-\rho}{d}, \frac{1+\rho}{d}\right]. \tag{10}$$

Proposition 1 shows that the random permutation equalizes coordinate variances at the Haar scale $1/d$ unless the input has a large coordinate. We prove the proposition in Appendix B.

4.5 Data-agnostic Codebook

Prior PTQ methods recalibrate because activation statistics shift across timesteps and prompts. OrbitQuant removes this dependence at the source. Proposition 1 shows that RPBH balances nor-

Table 1: GenEval results on three image diffusion transformers at W4A4 and W2A4. Values are scores on six compositional sub-tasks and Overall. **Bold** and underlined entries indicate the best and second-best result within each (model, bit-width) group. ↑ means higher is better. † represents our implementation.

Model	Method	Bit	Single object ↑	Two object ↑	Counting ↑	Colors ↑	Position ↑	Color attribution ↑	Overall ↑
FLUX.1-schnell	FP16	16/16	0.997	0.884	0.600	0.742	0.275	0.488	0.664
	Q-DiT	W4A4	0.741	0.424	0.378	0.418	0.073	0.208	0.373
	SmoothQuant	W4A4	0.619	0.293	0.272	0.317	0.043	0.143	0.281
	QuaRot	W4A4	0.819	0.543	0.472	0.519	0.118	0.275	0.458
	ViDiT-Q	W4A4	0.888	0.586	0.516	0.585	0.130	0.268	0.495
	SVDQuant	W4A4	0.994	**0.910**	0.450	0.708	0.260	0.420	0.624
	AdaTSQ	W4A4	**0.997**	0.894	0.622	0.793	0.278	0.498	0.680
	ConvRot†	W4A4	0.991	0.876	0.684	**0.806**	0.310	0.502	0.695
	OrbitQuant	W4A4	0.991	0.881	**0.706**	0.803	**0.323**	**0.512**	**0.703**
	QuaRot†	W2A4	0.006	0.000	0.000	0.000	0.000	0.000	0.001
	SmoothQuant†	W2A4	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	ViDiT-Q†	W2A4	0.006	0.000	0.000	0.000	0.000	0.000	0.001
	ConvRot†	W2A4	0.006	0.000	0.000	0.000	0.000	0.000	0.001
	OrbitQuant	W2A4	**0.972**	**0.697**	**0.575**	**0.766**	**0.198**	**0.420**	**0.604**
FLUX.1-dev	FP16	16/16	0.984	0.823	0.769	0.771	0.203	0.450	0.667
	Q-DiT	W4A4	0.047	0.000	0.009	0.024	0.000	0.003	0.014
	SmoothQuant	W4A4	0.003	0.000	0.003	0.011	0.000	0.000	0.007
	QuaRot	W4A4	0.634	0.106	0.294	0.346	0.025	0.050	0.243
	ViDiT-Q	W4A4	0.709	0.147	0.325	0.410	0.028	0.060	0.280
	SVDQuant	W4A4	0.981	0.710	0.610	0.698	0.140	0.300	0.573
	AdaTSQ	W4A4	0.981	0.770	0.640	0.708	**0.260**	0.350	0.618
	ConvRot†	W4A4	0.988	**0.780**	0.691	0.766	0.183	**0.430**	0.639
	OrbitQuant	W4A4	**0.991**	0.763	**0.706**	**0.769**	0.183	0.428	**0.640**
	QuaRot†	W2A4	0.006	0.000	0.000	0.000	0.000	0.000	0.001
	SmoothQuant†	W2A4	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	ViDiT-Q†	W2A4	0.006	0.000	0.000	0.000	0.000	0.000	0.001
	ConvRot†	W2A4	0.003	0.000	0.000	0.000	0.000	0.000	0.001
	OrbitQuant	W2A4	**0.956**	**0.424**	**0.481**	**0.678**	**0.110**	**0.203**	**0.475**
Z-Image-Turbo	FP16	16/16	1.000	0.907	0.709	0.859	0.468	0.583	0.754
	SmoothQuant	W4A4	0.003	0.000	0.000	0.000	0.000	0.000	0.000
	QuaRot	W4A4	0.906	0.505	0.416	0.692	0.250	0.343	0.519
	ViDiT-Q	W4A4	0.972	0.705	0.584	0.777	0.435	0.533	0.668
	SVDQuant	W4A4	0.994	0.843	0.633	0.833	0.485	0.520	0.718
	AdaTSQ	W4A4	0.994	**0.891**	0.681	0.872	**0.520**	**0.613**	0.762
	ConvRot†	W4A4	**1.000**	0.864	0.772	0.872	0.450	0.598	0.755
	OrbitQuant	W4A4	0.997	0.889	**0.781**	**0.888**	0.450	0.573	**0.767**
	QuaRot†	W2A4	0.006	0.000	0.000	0.000	0.000	0.000	0.001
	SmoothQuant†	W2A4	0.006	0.000	0.000	0.000	0.000	0.000	0.001
	ViDiT-Q†	W2A4	0.003	0.000	0.000	0.003	0.000	0.000	0.001
	ConvRot†	W2A4	0.125	0.000	0.000	0.053	0.000	0.000	0.003
	OrbitQuant	W2A4	**0.703**	**0.194**	**0.275**	**0.500**	**0.128**	**0.113**	**0.319**

malized activation variances around the Haar scale $1/d$, while the resulting coordinates empirically follow the same dimension-dependent marginal f_d. We therefore run Lloyd–Max on f_d offline to obtain a codebook $\mathcal{C}_{d,b}$ for each dimension and bit-width, reducing quantization to normalization, rotation, and nearest-centroid mapping without collecting input statistics. One $\mathcal{C}_{d,b}$ serves all timesteps, prompts, layers of dimension d, and their weight rows, making OrbitQuant calibration-free.

This codebook follows TurboQuant (Zandieh et al., 2025), which quantizes randomly rotated vectors using a codebook derived from their known post-rotation marginal. TurboQuant is a standalone vector quantizer for KV-cache and vector-database compression. It uses a dense $O(d^2)$ Haar rotation and operates as a quantize-dequantize codec, reconstructing each vector before use. OrbitQuant instead pairs the codebook with the structured RPBH rotation and absorbs it into the weights. The quantized operands then feed each linear layer directly, with no reconstruction and only a forward rotation at inference. Figure 3 confirms marginal matching at an attention projection (d=3072) and a feed-forward layer (d=12288). Raw activations deviate sharply from f_d, whereas RPBH-rotated activations closely match $f_d \approx \mathcal{N}(0, 1/d)$, similarly to a dense Haar rotation, supporting the use of the same dimension-dependent codebook across inputs.

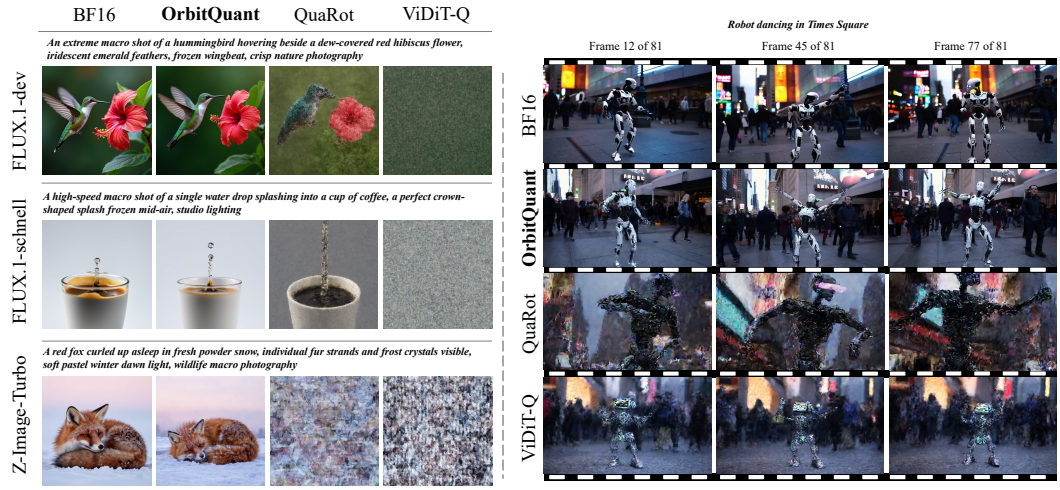
5 Experiments

5.1 Setup

Models and bit-widths. We evaluate OrbitQuant on three image DiTs and two video DiTs. For image generation we report FLUX.1-schnell (4-step, guidance 0.0), FLUX.1-dev (50-step, guidance 3.5), and Z-Image-Turbo (10-step, guidance 0.0) at W4A4 and W2A4. For video generation we report Wan 2.1-1.3B (81 frames, 480×832, 50 steps, CFG 5.0) and CogVideoX-2B (49 frames, 480×720, 50 steps, CFG 6.0) at W4A6 and W4A4. We quantize all transformer-block projections with OrbitQuant and keep the adaptive layer normalization (AdaLN) (Peebles & Xie, 2023) mod-

Table 2: VBench PTQ video-generation results on Wan 2.1-1.3B and CogVideoX-2B. Scores are percentages. **Bold** and underlined entries indicate the best and second-best result within each (model, bit-width) group. † represents our implementation.

Model	Method	Bit	Imaging Quality ↑	Aesthetic Quality ↑	Motion Smoothness ↑	Dynamic Degree ↑	Background Consistency ↑	Subject Consistency ↑	Scene ↑	Overall Consistency ↑
Wan 2.1-1.3B	Full Prec.	16/16	64.30	58.21	97.37	70.28	95.94	93.84	28.05	24.67
	SmoothQuant†	W4A6	53.51	49.19	**98.01**	34.44	94.89	92.66	12.81	22.15
	QuaRot†	W4A6	56.92	50.36	96.94	54.17	95.36	91.65	14.88	22.65
	ViDiT-Q	W4A6	56.24	50.18	94.81	52.43	89.67	82.53	13.45	19.58
	SVDQuant	W4A6	58.16	51.27	97.05	49.44	93.74	91.71	14.18	23.26
	OrbitQuant	W4A6	**61.25**	**56.08**	97.76	**59.78**	**95.51**	**94.23**	**24.88**	**24.35**
	SmoothQuant†	W4A4	46.32	36.33	96.39	51.94	95.85	90.39	2.79	15.05
	QuaRot†	W4A4	51.42	40.49	96.21	52.78	95.76	88.80	5.31	17.98
	ViDiT-Q†	W4A4	44.51	36.43	96.16	58.06	**95.92**	89.59	1.85	13.11
	SVDQuant	W4A4	57.57	46.30	94.21	**72.22**	93.16	77.96	12.73	21.91
	OrbitQuant	W4A4	**58.58**	**53.41**	**97.42**	53.89	95.30	**92.98**	**18.81**	**23.86**
CogVideoX-2B	Full Prec.	16/16	59.15	54.49	97.43	67.78	94.79	92.82	36.24	25.06
	SmoothQuant†	W4A6	51.50	49.70	**97.20**	30.00	94.70	91.10	21.90	23.20
	QuaRot†	W4A6	54.11	52.25	96.92	49.72	94.60	91.82	30.73	24.03
	ViDiT-Q	W4A6	54.72	43.01	92.18	43.22	90.76	81.02	26.25	20.41
	SVDQuant	W4A6	**58.27**	47.06	95.28	40.83	92.41	87.45	27.69	21.34
	OrbitQuant	W4A6	55.59	**54.42**	97.02	**57.50**	**94.78**	**92.56**	**32.51**	**24.55**
	SmoothQuant†	W4A4	39.90	35.50	**97.80**	1.90	**95.90**	**92.90**	3.60	12.80
	QuaRot†	W4A4	49.60	47.10	96.90	9.20	94.80	90.20	19.70	21.70
	ViDiT-Q†	W4A4	44.80	42.10	97.30	4.40	95.60	90.30	10.50	18.40
	SVDQuant	W4A4	51.60	49.40	97.69	42.22	94.03	91.78	25.67	22.89
	OrbitQuant	W4A4	**52.62**	**51.66**	96.99	**42.78**	94.50	91.65	**28.53**	**23.86**



(a) Image generation on different models at W3A3

(b) Video generation on Wan 14B at W4A4

Figure 4: Qualitative comparison of OrbitQuant against QuaRot (Ashkboos et al., 2024) and ViDiT-Q (Zhao et al., 2024), with the BF16 full-precision output shown for reference. **(a)** Image generation at W3A3 on FLUX.1-dev, FLUX.1-schnell, and Z-Image-Turbo, with one prompt per model. **(b)** Video generation at W4A4 on Wan 14B, showing three sampled frames per method.

ulation projections, where present, at INT4 weight round-to-nearest (RTN) (Li et al., 2024). This AdaLN treatment is identical across all methods we implement. Wan 2.1-1.3B has no AdaLN modulation, so only its transformer-block projections are quantized. All experiments are conducted on NVIDIA H100 GPUs. The full list of quantized and skipped layers is given in Appendix C.

Baselines. Image baselines are the calibration-based SVDQuant (Li et al., 2024), AdaTSQ (Zhang et al., 2026), and ViDiT-Q (Zhao et al., 2024), together with Q-DiT (Chen et al., 2025), QuaRot (Ashkboos et al., 2024), SmoothQuant (Xiao et al., 2023), and ConvRot (Huang et al., 2025a). We reproduce ConvRot at W4A4 with its official NVFP4 setup. Since NVFP4 is a fixed 4-bit format, we instantiate ConvRot below 4 bits with uniform INT quantization following QuaRot. Video baselines are ViDiT-Q, SVDQuant, QuaRot, and SmoothQuant. Baseline numbers are taken primarily from AdaTSQ (Zhang et al., 2026) for image and QVGen (Huang et al., 2025b) for video.

5.2 Image Generation: GenEval

Table 1 reports GenEval (Ghosh et al., 2023) Overall and per-task scores. At W4A4, OrbitQuant is near-lossless, exceeding FP16 on Overall on FLUX.1-schnell and Z-Image-Turbo and trailing it by only 0.027 on FLUX.1-dev, while outperforming the strongest PTQ baselines across all three

Table 3: End-to-end PTQ cost on FLUX.1-schnell and Wan 2.1-1.3B. Cal., App., and Total denote calibration, application, and total runtime in seconds. Rel. is normalized to OrbitQuant. "–" indicates that no calibration is required.

Model	Method	Cal. (s)	App. (s)	Total (s)	Rel.	Peak (GiB)
FLUX.1-schnell	**OrbitQuant**	–	60.23	**60.23**	**1.00×**	32.64
	QuaRot	–	62.06	62.06	1.03×	**32.28**
	SmoothQuant	42.28	62.24	104.52	1.74×	34.43
	ViDiT-Q	42.25	61.26	103.51	1.72×	48.56
Wan-1.3B	**OrbitQuant**	–	0.30	**0.30**	**1.00×**	13.89
	QuaRot	–	0.40	0.40	1.34×	**13.78**
	SmoothQuant	644.82	0.41	645.22	2139.87×	16.16
	ViDiT-Q	649.92	0.64	650.57	2157.59×	16.66

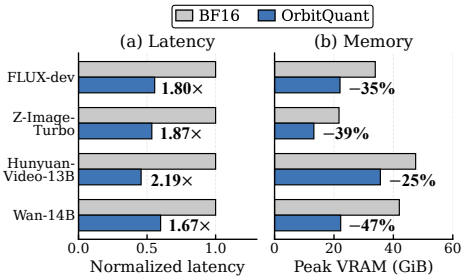


Figure 5: End-to-end efficiency on four DiTs. (a) Latency normalized to BF16 for each model. (b) Peak inference VRAM.

models. The advantage widens at W2A4, where the rotation and smoothing baselines collapse to near-zero on all three backbones while OrbitQuant remains functional, retaining strong quality on the FLUX models and producing the only meaningful score on Z-Image-Turbo. Results at W3A3 and W2A3 are presented in Appendix D.1.

5.3 Video Generation: VBench

OrbitQuant applies to Wan 2.1-1.3B (Wan et al., 2025) and CogVideoX-2B (Yang et al., 2025c) with the same recipe used for the image experiments, and Table 2 reports the full VBench (Huang et al., 2024) comparison. At W4A6, OrbitQuant achieves the best Overall score among PTQ methods on both backbones and remains competitive across the VBench dimensions. The advantage persists at W4A4, where OrbitQuant again ranks first among PTQ methods in Overall on both models while maintaining strong scores across quality and consistency metrics. Results on larger models, including Wan 14B (Wan et al., 2025) and HunyuanVideo (Kong et al., 2024), and comparisons against quantization-aware training (QAT) are provided in Appendix D.2 and D.3, respectively.

5.4 Qualitative Comparison

Figure 4 shows generations from OrbitQuant, QuaRot (Ashkboos et al., 2024), and ViDiT-Q (Zhao et al., 2024) alongside the BF16 reference. For images at W3A3, OrbitQuant stays close to BF16 on FLUX.1-dev, FLUX.1-schnell, and Z-Image-Turbo, retaining fine structure and color, while the other methods lose fidelity and collapse to noise on Z-Image-Turbo. For Wan 14B video at W4A4, OrbitQuant preserves scene layout and stays consistent across frames, whereas the other methods drift in color and structure.

5.5 Efficiency Analysis

Quantization cost. Table 3 reports the one-time cost of producing the quantized model. OrbitQuant requires no calibration pass: it quantizes FLUX.1-schnell in 60.2 s and Wan-1.3B in 0.3 s, with the cost spent entirely on offline model conversion. The FLUX cost is dominated by INT4 quantization of the AdaLN modulation weights (∼60 s), which Wan does not require. QuaRot is similarly calibration-free but slightly slower (1.03× and 1.34×), while the calibration-based SmoothQuant and ViDiT-Q take up to 2157× longer on Wan, dominated by their calibration passes.

Inference efficiency. We measure end-to-end inference efficiency using optimized GPU kernel implementations on an NVIDIA H100 GPU. Figure 5 reports generation latency and peak inference memory at W4A4 against BF16 on four DiTs. OrbitQuant delivers 1.67–2.19× speedup, with the largest gain on HunyuanVideo-13B, and reduces peak memory by 25–47%. In particular, it reduces the latency of Wan-14B from 482.9 s to 289.2 s and keeps the peak memory of all evaluated models below 36 GiB. A fake-quantization overhead comparison against prior PTQ baselines is provided in Appendix D.5.

6 Ablations

6.1 Comparison between Rotation Matrices

The forward identity of Section 4.3 holds for any orthogonal rotation, but codebook compatibility requires the rotated marginal to match f_d. Table 4 compares four rotations inside an otherwise identical pipeline on FLUX.1-schnell. At W4A4 all four perform similarly, but they separate at lower bit-widths, where RPBH leads at both W3A3 and W2A4. The gap over Block-RHT, which

Table 4: Rotation-class ablation on FLUX.1-schnell. GenEval Overall (mean over three seeds) and per-image activation-rotation latency at 1024^2 on an H100.

Rotation	W4A4	W3A3	W2A4	Latency(s)
Haar	**0.696**	0.669	0.591	11.65
Full RHT	0.691	0.672	0.587	0.452
Block-RHT	0.678	0.642	0.558	0.381
RPBH (ours)	0.690	**0.674**	**0.595**	**0.451**

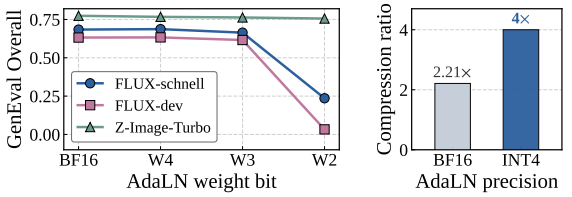


Figure 6: AdaLN bit-width ablation, main model fixed at W4A4. Left: GenEval Overall as the AdaLN weight bit drops. Right: compression ratio on FLUX.

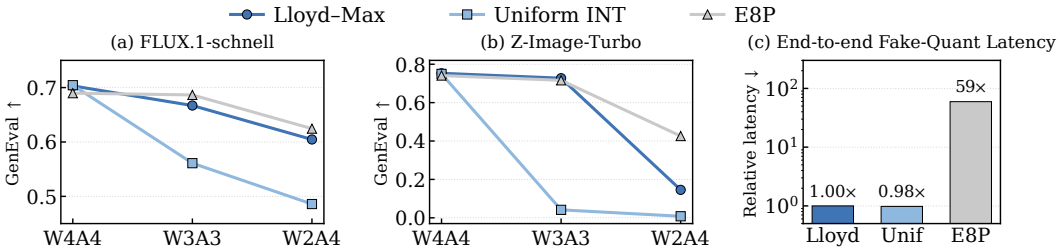


Figure 7: Quantizer ablation. (a–b) GenEval across bit-widths on FLUX.1-schnell and Z-Image-Turbo. (c) End-to-end fake-quantization latency relative to Lloyd–Max.

is RPBH without the permutation, highlights the role of the random permutation: it redistributes input mass across blocks, improving the match between the rotated coordinates and f_d. On latency, the structured rotations admit fast Hadamard kernels unavailable to dense Haar ($26\times$ faster), and RPBH is no slower than the Full RHT used in rotation-based LLM quantizers (Ashkboos et al., 2024; Tseng et al., 2024), while accommodating every dimension in our study, including $d{=}1920$ in CogVideoX-2B where a standard size-d FWHT cannot be directly applied.

6.2 AdaLN Bit-width

OrbitQuant fixes the AdaLN modulation projections at INT4 weight RTN, since their timestep-dependent scale-and-shift cannot be folded into neighboring weights. Holding the main model at W4A4 and varying only the AdaLN weight bit with activations in BF16, Figure 6 shows that INT4 nearly matches BF16 on all three models. W3 remains competitive, while W2 causes severe degradation on both FLUX models, though Z-Image-Turbo stays robust. INT4 therefore provides a favorable accuracy–compression trade-off. These projections account for 27% of the weights, so keeping them in BF16 would reduce FLUX compression from $4\times$ to $2.21\times$, as shown in Figure 6, right.

6.3 Quantizer Choice

To clarify the quantizer choice, we compare three representative activation quantizers on top of the same RPBH rotation: Uniform INT, Lloyd–Max, and E8P. Uniform INT follows the QuaRot setting (Ashkboos et al., 2024), while E8P uses the 8-D lattice codebook (Tseng et al., 2024). Uniform INT is competitive at W4A4 but degrades sharply at lower bit-widths, particularly on Z-Image-Turbo. E8P is more robust than Uniform INT at low precision, but its 8-D E_8 decoding incurs about $59\times$ higher latency than Lloyd–Max, making it costly for online activation quantization. In contrast, Lloyd–Max uses a lightweight scalar codebook with per-element nearest-centroid lookup, retaining strong low-bit accuracy with overhead comparable to Uniform INT. Lloyd–Max therefore provides the best balance between low-bit accuracy and online quantization efficiency for OrbitQuant.

7 Conclusion

We present OrbitQuant, a calibration-free weight-activation quantizer for diffusion transformers that replaces per-timestep range calibration with a single distributional codebook applied in a shared, rotated, normalized basis. The rotation is absorbed into the weights offline and cancels inside each linear layer, leaving only a forward RPBH rotation on the activations at runtime. Its random permutation balances input mass across blocks and improves marginal matching at low bit-width. Across FLUX.1, Z-Image-Turbo, Wan 2.1, and CogVideoX, the same recipe transfers from image to video without per-modality tuning, achieves state-of-the-art PTQ performance across low-bit GenEval and VBench settings, and supports usable 2-bit weights where prior PTQ methods collapse.

AI USE STATEMENT

In this work, generative AI tools were used to aid and polish the writing, support the retrieval and discovery of related work, draft portions of the manuscript, and assist in proving mathematical claims, including suggesting intermediate derivation steps and refining proof structure. The authors checked all retrieved references against the original sources and independently reviewed and verified all AI-assisted mathematical arguments. All AI-generated text was revised and approved by the authors. The authors take full responsibility for the final content of this work.

REFERENCES

Nir Ailon and Bernard Chazelle. The fast johnson–lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on computing*, 39(1):302–322, 2009.

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.

Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22669–22679, 2023.

Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Xin Jin, Liangchen Li, et al. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 28306–28315, 2025.

Yongsen Cheng, Kai Liu, Kaiwen Tao, Junxian Li, Zhixin Wang, Zhikai Chen, Renjing Pei, and Yulun Zhang. Permuquant: Lowering per-group quantization error by reordering channels for diffusion models. *arXiv preprint arXiv:2605.09503*, 2026.

Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. Spqr: A sparse-quantized representation for near-lossless llm weight compression. *arXiv preprint arXiv:2306.03078*, 2023.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Weilun Feng, Haotong Qin, Chuanguang Yang, Xiangqi Li, Han Yang, Yuqi Li, Zhulin An, Libo Huang, Michele Magno, and Yongjun Xu. S^2 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.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Hongyaoxing Gu, Lijuan Hu, Liye Yu, Haowei Li, and Fangfang Liu. Lopro: Enhancing low-rank quantization via permuted block-wise rotation. *arXiv preprint arXiv:2601.19675*, 2026.

Insu Han, Praneeth Kacham, Amin Karbasi, Vahab Mirrokni, and Amir Zandieh. Polarquant: Vector quantization with polar transformation. In *The 29th International Conference on Artificial Intelligence and Statistics*, 2026.

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

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

Xing Hu, Yuan Cheng, Dawei Yang, Zukang Xu, Zhihang Yuan, Jiangyong Yu, Chen Xu, Zhe Jiang, and Sifan Zhou. Ostquant: Refining large language model quantization with orthogonal and scaling transformations for better distribution fitting. *arXiv preprint arXiv:2501.13987*, 2025.

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, Yifu Ding, Chengtao Lv, Haotong Qin, and Jun Zhang. Qvgen: Pushing the limit of quantized video generative models. *arXiv preprint arXiv:2505.11497*, 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, 2024.

Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W Mahoney, and Kurt Keutzer. Squeezellm: Dense-and-sparse quantization. *arXiv preprint arXiv:2306.07629*, 2023.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL https://arxiv.org/abs/2506.15742.

Donghyun Lee, Yuhang Li, Ruokai Yin, and Priyadarshini Panda. Kronq: Llm quantization via kronecker-factored hessian. *arXiv preprint arXiv:2607.07964*, 2026.

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.

Yuhang Li, Ruokai Yin, Donghyun Lee, Shiting Xiao, and Priyadarshini Panda. Gptaq: Efficient finetuning-free quantization for asymmetric calibration. *arXiv preprint arXiv:2504.02692*, 2025a.

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. *arXiv preprint arXiv:2505.18663*, 2025b.

Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. Duquant: Distributing outliers via dual transformation makes stronger quantized llms. *Advances in Neural Information Processing Systems*, 37:87766–87800, 2024a.

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, 2024b.

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.

Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2): 129–137, 1982.

Joel Max. Quantizing for minimum distortion. *IRE Transactions on Information Theory*, 6(1):7–12, 1960.

Francesco Mezzadri. How to generate random matrices from the classical compact groups. *arXiv preprint math-ph/0609050*, 2006.

OpenAI. Sora: Creating video from text. https://openai.com/sora, 2024. Accessed: 2024-02-15.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

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*, pp. 10684–10695, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

Sai Sanjeet, Ian Colbert, Pablo Monteagudo-Lago, Giuseppe Franco, Yaman Umuroglu, and Nicholas J Fraser. Pushing the limits of block rotations in post-training quantization. *arXiv preprint arXiv:2601.22347*, 2026.

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.

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, et al. Flatquant: Flatness matters for llm quantization. *arXiv preprint arXiv:2410.09426*, 2024.

Joel A Tropp. Improved analysis of the subsampled randomized hadamard transform. *Advances in Adaptive Data Analysis*, 3(01n02):115–126, 2011.

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.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 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.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024.

Kaicheng Yang, Xun Zhang, Haotong Qin, Yucheng Lin, Kaisen Yang, Xianglong Yan, and Yulun Zhang. Robuq: Pushing dits to w1. 58a2 via robust activation quantization. *arXiv preprint arXiv:2509.23582*, 2025a.

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. *arXiv preprint arXiv:2508.03485*, 2025b.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, volume 2025, pp. 83048–83077, 2025c.

Amir Zandieh, Majid Daliri, Majid Hadian, and Vahab Mirrokni. Turboquant: Online vector quantization with near-optimal distortion rate. *arXiv preprint arXiv:2504.19874*, 2025.

Shaoqiu Zhang, Zizhong Ding, Kaicheng Yang, Junyi Wu, Xianglong Yan, Xi Li, Bingnan Duan, Jianping Fang, and Yulun Zhang. Adatsq: Pushing the pareto frontier of diffusion transformers via temporal-sensitivity quantization. *arXiv preprint arXiv:2602.09883*, 2026.

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.

A ALGORITHM

Algorithm 1 OrbitQuant offline weight patching and online activation quantization

Require: Transformer $\mathcal{T}$, target layers $\mathcal{L}$, input dimensions $\mathcal{D}$, bit-widths (b_w, b_a), clamp ε

▷ **Offline**
for $d \in \mathcal{D}$ **do**
  $\mathbf{\Pi}_d \leftarrow \mathrm{RPBH}(d)$
  $\hat{Q}^{(d)}_{b_w}, \hat{Q}^{(d)}_{b_a} \leftarrow \textsc{LloydMax}(d, b_w), \textsc{LloydMax}(d, b_a)$
end for
for each $\mathbf{W} \in \mathcal{L}$ with input dimension d **do**
  $\mathbf{W}' \leftarrow \mathbf{W}\mathbf{\Pi}_d^\top$
  $r'_i \leftarrow \|\mathbf{w}'_i\|_2, \quad \tilde{\mathbf{w}}'_i \leftarrow \mathbf{w}'_i / r'_i \quad$ for $i = 1, \ldots, m$
  $\hat{\mathbf{W}}' \leftarrow \mathrm{diag}(\mathbf{r}')\, \hat{Q}^{(d)}_{b_w}(\tilde{\mathbf{W}}')$
  Replace $\mathbf{W}$ by $\hat{\mathbf{W}}'$ in $\mathcal{T}$
end for

▷ **Online** on tokens $\mathbf{x} \in \mathbb{R}^{N \times d}$
$\mathbf{x}' \leftarrow \mathbf{x}\mathbf{\Pi}_d^\top$
$s_i \leftarrow \|\mathbf{x}'_i\|_2, \quad \tilde{\mathbf{x}}'_i \leftarrow \mathbf{x}'_i / (s_i + \varepsilon) \quad$ for $i = 1, \ldots, N$
$\hat{\mathbf{x}}'_i \leftarrow s_i \cdot \hat{Q}^{(d)}_{b_a}(\tilde{\mathbf{x}}'_i) \quad$ for $i = 1, \ldots, N$
return $\hat{\mathbf{x}}'$

B RPBH COORDINATE-VARIANCE CONCENTRATION AND INCOHERENCE

Setup. Fix a unit vector $\tilde{\mathbf{x}} \in \mathbb{R}^d$ and write $d = kh$. Let π be a uniformly random permutation, and partition $\mathbf{y} = \mathbf{P}_\pi \tilde{\mathbf{x}}$ into k blocks $\mathbf{y}^{(j)} \in \mathbb{R}^h$. Define the block masses $M_j = \|\mathbf{y}^{(j)}\|_2^2$, so that $\sum_{j=1}^k M_j = 1$. For each block, let $\mathbf{z}^{(j)} = \mathbf{H}_h \mathbf{D}_j \mathbf{y}^{(j)}$, where $(\mathbf{H}_h)_{\ell i} \in \{-1/\sqrt{h}, 1/\sqrt{h}\}$ and the diagonal entries of $\{\mathbf{D}_j\}_{j=1}^k$ are independent Rademacher variables, independent of π. Write $\mathbf{z} = (\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(k)}) = \mathbf{\Pi}_d \tilde{\mathbf{x}}$, $M_{\max} = \max_{j \in [k]} M_j$, and $\mu_\infty = \|\tilde{\mathbf{x}}\|_\infty^2$.

Lemma 1 (Mass balancing)**.** *For any $\delta \in (0, 1)$, with probability at least $1 - \delta$ over π, simultaneously for all $j \in [k]$,*

$$\left| M_j - \frac{1}{k} \right| \le \mu_\infty \sqrt{\frac{h}{2} \log \frac{2k}{\delta}}. \tag{11}$$

Proof. For any fixed j, the uniformly random permutation π assigns to block j a uniformly random subset $S_j \subseteq [d]$ of size h. Therefore, $M_j = \sum_{i \in S_j} \tilde{x}_i^2$ is the sum of h values sampled without replacement from the population $\{\tilde{x}_i^2\}_{i=1}^d \subseteq [0, \mu_\infty]$. The population mean is $1/d$ because $\|\tilde{\mathbf{x}}\|_2^2 = 1$, and hence $\mathbb{E}_\pi[M_j] = h/d = 1/k$.

Hoeffding's inequality for sampling without replacement (Hoeffding, 1963) gives

$$\Pr_\pi \left[\left| M_j - \frac{1}{k} \right| \ge \epsilon \right] \le 2 \exp\left(-\frac{2\epsilon^2}{h \mu_\infty^2} \right). \tag{12}$$

Setting $\epsilon = \mu_\infty \sqrt{(h/2) \log(2k/\delta)}$ in Equation 12 makes the failure probability for each block at most δ/k. A union bound over the k blocks proves Equation 11. □

Proof of Proposition 1. Fix a coordinate $z_\ell^{(j)}$ in block j. Conditioned on π, the block $\mathbf{y}^{(j)}$ is fixed, and

$$\mathbb{E}_{\mathbf{D}} \left[z_\ell^{(j)} \mid \pi \right] = 0, \qquad \mathrm{Var}_{\mathbf{D}} \left(z_\ell^{(j)} \mid \pi \right) = \sum_{i=1}^h (\mathbf{H}_h)_{\ell i}^2 (y_i^{(j)})^2 = \frac{M_j}{h}. \tag{13}$$

The expectation is zero because the diagonal entries of $\mathbf{D}_j$ are mean-zero Rademacher variables. Their independence eliminates the cross terms in the variance.

Let $\rho = d\mu_\infty \sqrt{\frac{1}{2h} \log \frac{2k}{\delta}}$. The right-hand side of Equation 11 equals ρ/k. Hence, on the event of Lemma 1,

$$\frac{1-\rho}{k} \le M_j \le \frac{1+\rho}{k} \qquad \text{for all } j \in [k]. \tag{14}$$

Combining Equations 13 and 14, and using $d = kh$, gives

$$\mathrm{Var}_{\mathbf{D}}\left(z_\ell^{(j)} \mid \pi\right) \in \left[\frac{1-\rho}{d}, \frac{1+\rho}{d}\right] \qquad \text{for all } j \in [k],\ \ell \in [h]. \tag{15}$$

Lemma 1 ensures that this event holds with probability at least $1-\delta$ over π, completing the proof. □

Lemma 2 (Mass-dependent block incoherence)**.** *For any $\delta \in (0,1)$, conditioned on any realization of π, with probability at least $1-\delta$ over $\{\mathbf{D}_j\}_{j=1}^k$,*

$$\|\mathbf{z}\|_\infty \le \sqrt{\frac{2M_{\max}}{h} \log \frac{2d}{\delta}}. \tag{16}$$

Proof. Conditioned on π, each block $\mathbf{y}^{(j)}$ and its mass M_j are fixed. For $j \in [k]$ and $\ell \in [h]$, write $z_\ell^{(j)} = \sum_{i=1}^h a_i \sigma_i^{(j)}$, where $a_i = (\mathbf{H}_h)_{\ell i} y_i^{(j)}$ and $\sigma_i^{(j)}$ is the i-th diagonal entry of $\mathbf{D}_j$. Thus, $z_\ell^{(j)}$ is a mean-zero Rademacher sum.

If $M_j = 0$, then $\mathbf{y}^{(j)} = \mathbf{0}$ and hence $\mathbf{z}^{(j)} = \mathbf{0}$. Suppose that $M_j > 0$. By Hoeffding's inequality (Hoeffding, 1963), $\Pr[|\sum_i a_i \sigma_i^{(j)}| > t] \le 2\exp(-t^2/(2\sum_i a_i^2))$. Since $(\mathbf{H}_h)_{\ell i}^2 = 1/h$, we have $\sum_i a_i^2 = M_j/h$. Therefore,

$$\Pr_{\mathbf{D}_j}\left[|z_\ell^{(j)}| > t \,\middle|\, \pi\right] \le 2\exp\left(-\frac{t^2 h}{2M_j}\right). \tag{17}$$

Set $t_j = \sqrt{2M_j \log(2d/\delta)/h}$. Equation 17 then gives $\Pr_{\mathbf{D}_j}[|z_\ell^{(j)}| > t_j \mid \pi] \le \delta/d$. A union bound over all $d = kh$ output coordinates shows that, with probability at least $1-\delta$, $|z_\ell^{(j)}| \le t_j$ simultaneously for every $j \in [k]$ and $\ell \in [h]$. Since $M_j \le M_{\max}$, taking the maximum over all coordinates gives Equation 16. □

C Additional Experimental Details

C.1 Generation settings

Image models use the sampler and step count of their public checkpoints: FLUX.1-schnell at 4 steps with guidance 0.0, FLUX.1-dev at 50 steps with guidance 3.5, and Z-Image-Turbo at 10 steps with guidance 0.0. Video models use Wan 2.1-1.3B at 81 frames, 480×832, 50 steps, and CFG 5.0, and CogVideoX-2B at 49 frames, 480×720, 50 steps, and CFG 6.0. All experiments are conducted on NVIDIA H100 GPUs.

C.2 Quantized and skipped layers

We quantize every linear projection in the transformer blocks through the OrbitQuant path, including the image- and text-side Q, K, V, and output projections and the feed-forward layers of every block. We also quantize the text-conditioning K and V projections that consume text-encoder hidden states, corresponding to the joint-attention text path in FLUX and Z-Image and the cross-attention projections in Wan and CogVideoX.

AdaLN modulation projections are the one exception. Their outputs parameterize timestep-dependent elementwise scale-and-shift. A static norm affine can be folded into neighboring weights, as in rotation-based LLM quantizers (Ashkboos et al., 2024), but this dynamic modulation cannot. The shared-rotation cancellation of Section 4 therefore cannot be applied to these projections. Their input is a single conditioning token per step, leaving little activation compute to save. We therefore quantize only their weights using INT4 RTN with group size 64 and keep their activations in BF16. Embeddings, the timestep MLP, the final un-patchify head, and the text encoder remain in BF16.

Table 5: GenEval results at the lowest bit-widths, W3A3 and W2A3, on three image diffusion transformers. **Bold** and underlined entries indicate the best and second-best PTQ result within each (model, bit-width) group. ↑ means higher is better.† represents our implementation.

Model	Method	Bit	Single object ↑	Two object ↑	Counting ↑	Colors ↑	Position ↑	Color attribution ↑	Overall ↑
FLUX.1-schnell	FP16	16/16	0.997	0.884	0.600	0.742	0.275	0.488	0.664
	SVDQuant	W3A3	0.820	0.647	0.466	0.560	0.160	0.373	0.504
	AdaTSQ	W3A3	**0.997**	**0.920**	0.530	0.688	**0.230**	0.440	0.634
	ConvRot†	W3A3	0.003	0.000	0.000	0.053	0.000	0.000	0.001
	OrbitQuant	W3A3	0.978	0.861	**0.684**	**0.777**	0.223	**0.542**	**0.678**
	QuaRot†	W2A3	0.003	0.000	0.000	0.000	0.000	0.000	0.001
	SmoothQuant†	W2A3	0.003	0.000	0.000	0.000	0.000	0.000	0.001
	ViDiT-Q†	W2A3	0.009	0.000	0.000	0.003	0.000	0.000	0.002
	ConvRot†	W2A3	0.006	0.000	0.000	0.053	0.000	0.000	0.001
	OrbitQuant	W2A3	**0.947**	**0.573**	**0.431**	**0.691**	**0.140**	**0.318**	**0.517**
FLUX.1-dev	FP16	16/16	0.984	0.823	0.769	0.771	0.203	0.450	0.667
	SVDQuant	W3A3	0.869	0.288	0.425	0.524	0.033	0.123	0.377
	AdaTSQ	W3A3	0.956	0.548	**0.628**	0.656	0.083	0.290	0.527
	ConvRot†	W3A3	0.025	0.000	0.000	0.008	0.000	0.000	0.006
	OrbitQuant	W3A3	**0.981**	**0.684**	0.606	**0.734**	**0.128**	**0.372**	**0.584**
	QuaRot†	W2A3	0.003	0.000	0.000	0.000	0.000	0.000	0.001
	SmoothQuant†	W2A3	0.003	0.000	0.000	0.000	0.000	0.000	0.001
	ViDiT-Q†	W2A3	0.0013	0.000	0.000	0.003	0.000	0.000	0.002
	ConvRot†	W2A3	0.006	0.000	0.000	0.000	0.000	0.000	0.001
	OrbitQuant	W2A3	**0.906**	**0.235**	**0.338**	**0.582**	**0.050**	**0.120**	**0.372**
Z-Image-Turbo	FP16	16/16	1.000	0.907	0.709	0.859	0.468	0.583	0.754
	SVDQuant	W3A3	0.005	0.000	0.000	0.000	0.000	0.000	0.000
	AdaTSQ	W3A3	**0.994**	**0.870**	0.550	**0.885**	**0.410**	0.455	0.694
	ConvRot†	W3A3	0.003	0.000	0.000	0.053	0.000	0.000	0.001
	OrbitQuant	W3A3	**0.994**	0.846	**0.750**	0.859	0.395	**0.598**	**0.740**
	QuaRot†	W2A3	0.013	0.000	0.000	0.000	0.000	0.000	0.002
	SmoothQuant†	W2A3	0.009	0.000	0.000	0.000	0.000	0.000	0.002
	ViDiT-Q†	W2A3	0.000	0.000	0.000	0.003	0.000	0.000	0.000
	ConvRot†	W2A3	0.016	0.003	0.000	0.053	0.000	0.000	0.003
	OrbitQuant	W2A3	**0.272**	**0.023**	**0.028**	**0.269**	**0.018**	**0.023**	**0.105**

Table 6: VBench results on Wan 14B and HunyuanVideo at W4A4. Per-dimension scores over eight VBench dimensions. **Bold** and underlined entries indicate the best and second-best PTQ result. QVGen is a QAT method, shown for reference and excluded from the PTQ ranking. ↑ means higher is better. † represents our implementation.

Model	Method	Bit	Imaging Quality ↑	Aesthetic Quality ↑	Motion Smoothness ↑	Dynamic Degree ↑	Background Consistency ↑	Subject Consistency ↑	Scene ↑	Overall Consistency ↑
Wan 14B	BF16	16/16	0.6514	0.6136	0.9738	0.7389	0.9632	0.9365	0.3330	0.2629
	SmoothQuant†	W4A4	0.5971	0.5263	**0.9763**	0.4472	0.9390	0.9171	0.1439	0.2327
	QuaRot†	W4A4	0.6332	0.5686	0.9701	0.5500	0.9504	0.9185	0.2589	0.2541
	ViDiT-Q†	W4A4	0.5948	0.5373	0.9672	0.5417	0.9533	0.9202	0.1849	0.2432
	OrbitQuant	W4A4	**0.6405**	**0.6022**	0.9754	**0.6250**	**0.9559**	**0.9363**	**0.3285**	**0.2615**
HunyuanVideo	BF16	16/16	0.6478	0.6253	0.9930	0.5139	0.9701	0.9605	0.4281	0.2586
	SmoothQuant	W4A4	0.5946	0.4841	0.9879	0.0139	0.9672	0.9497	0.0784	0.2109
	QuaRot	W4A4	0.5430	0.4485	0.9222	**0.8750**	0.9769	0.9264	0.0094	0.1733
	ViDiT-Q	W4A4	0.4010	0.4536	**0.9943**	0.0000	0.9719	**0.9729**	0.0785	0.1966
	DVD-Quant	W4A4	0.6182	**0.6196**	0.9915	0.5694	**0.9782**	0.9661	0.2994	**0.2568**
	OrbitQuant	W4A4	**0.6209**	0.6072	0.9930	0.4417	0.9751	0.9622	**0.3052**	0.2283

D Additional Experiments

D.1 Lowest bit-widths: W3A3 and W2A3

We push to the lowest bit-widths, W3A3 and W2A3, on three image diffusion transformers. Table 5 reports GenEval. At W3A3, we compare against the low-bit image quantizers SVDQuant (Li et al., 2024) and AdaTSQ (Zhang et al., 2026), and at W2A3 against the rotation and smoothing baselines. At W3A3, OrbitQuant has the best Overall score on all three models and stays close to FP16. AdaTSQ is competitive and leads on a few individual dimensions, while SVDQuant collapses entirely on Z-Image-Turbo. W2A3 is the harder test. The rotation and smoothing baselines collapse to near zero on every model, consistent with the difficulty of representing the dense regions of rotated activations with a 3-bit uniform grid. OrbitQuant is the only method that stays functional, remaining usable on the FLUX models. Z-Image-Turbo is the exception, where even OrbitQuant degrades sharply, marking the limit of a calibration-free codebook at this bit-width.

Table 7: Video-generation results on Wan 2.1-1.3B and CogVideoX-2B at W4A4. Scores are percentages. P/Q marks each method as quantization-aware training (QAT) or post-training quantization (PTQ). **Bold** and underlined indicate the best and second-best result across all methods within each model; full-precision rows are references.

Model	Method	P/Q	Bit	Imaging Quality ↑	Aesthetic Quality ↑	Motion Smoothness ↑	Dynamic Degree ↑	Background Consistency ↑	Subject Consistency ↑	Scene ↑	Overall Consistency ↑
Wan 2.1-1.3B	Full Prec.	–	16/16	64.30	58.21	97.37	70.28	95.94	93.84	28.05	24.67
	LSQ	QAT	W4A4	59.11	49.09	**98.35**	71.11	92.66	91.67	10.38	18.83
	Q-DM	QAT	W4A4	60.40	52.50	97.22	76.67	93.37	89.26	13.28	21.63
	EfficientDM	QAT	W4A4	60.70	53.57	96.18	56.39	93.74	91.70	11.77	21.19
	QVGen	QAT	W4A4	**63.08**	**54.67**	98.25	**77.78**	**94.08**	92.57	15.32	23.01
	SmoothQuant	PTQ	W4A4	46.32	36.33	96.39	51.94	95.85	90.39	2.79	15.05
	QuaRot	PTQ	W4A4	51.42	40.49	96.21	52.78	95.76	88.80	5.31	17.98
	ViDiT-Q	PTQ	W4A4	44.51	36.43	96.16	58.06	**95.92**	89.59	1.85	13.11
	SVDQuant	PTQ	W4A4	57.57	46.30	94.21	72.22	93.16	77.96	12.73	21.91
	OrbitQuant	PTQ	W4A4	58.58	53.41	97.42	53.89	95.30	**92.98**	**18.81**	**23.86**
CogVideoX-2B	Full Prec.	–	16/16	59.15	54.49	97.43	67.78	94.79	92.82	36.24	25.06
	LSQ	QAT	W4A4	58.73	54.20	97.57	45.00	92.97	92.41	24.06	23.17
	Q-DM	QAT	W4A4	54.96	52.71	98.00	48.61	93.82	91.86	28.02	23.87
	EfficientDM	QAT	W4A4	55.96	51.97	98.03	46.67	94.10	91.70	27.76	24.28
	QVGen	QAT	W4A4	**60.16**	**54.61**	**98.06**	**67.22**	**94.38**	**93.01**	**31.42**	**24.61**
	SmoothQuant	PTQ	W4A4	39.90	35.50	97.80	1.90	95.90	92.90	3.60	12.80
	QuaRot	PTQ	W4A4	49.60	47.10	96.90	9.20	94.80	90.20	19.70	21.70
	ViDiT-Q	PTQ	W4A4	44.80	42.10	97.30	4.40	95.60	90.30	10.50	18.40
	SVDQuant	PTQ	W4A4	51.60	49.40	97.69	42.22	94.03	91.78	25.67	22.89
	OrbitQuant	PTQ	W4A4	52.62	51.66	96.99	42.78	94.50	91.65	28.53	23.86

Table 8: Seed robustness of OrbitQuant on GenEval. Mean and standard deviation over three random seeds on three image diffusion transformers at W4A4 and W2A4. ↑ means higher is better.

Model	Bit	Single object ↑	Two object ↑	Counting ↑	Colors ↑	Position ↑	Color attribution ↑	**Overall** ↑
FLUX.1-schnell	W4A4	0.991 ± 0.003	0.879 ± 0.019	0.685 ± 0.018	0.793 ± 0.011	0.280 ± 0.039	0.510 ± 0.014	$\mathbf{0.690 \pm 0.012}$
	W2A4	0.963 ± 0.011	0.692 ± 0.013	0.577 ± 0.010	0.754 ± 0.025	0.164 ± 0.038	0.423 ± 0.031	$\mathbf{0.595 \pm 0.008}$
FLUX.1-dev	W4A4	0.990 ± 0.002	0.763 ± 0.005	0.721 ± 0.027	0.761 ± 0.007	0.177 ± 0.010	0.421 ± 0.009	$\mathbf{0.639 \pm 0.004}$
	W2A4	0.943 ± 0.014	0.395 ± 0.032	0.480 ± 0.011	0.668 ± 0.012	0.079 ± 0.027	0.198 ± 0.007	$\mathbf{0.460 \pm 0.014}$
Z-Image-Turbo	W4A4	0.998 ± 0.002	0.880 ± 0.009	0.766 ± 0.022	0.875 ± 0.015	0.464 ± 0.017	0.618 ± 0.020	$\mathbf{0.767 \pm 0.001}$
	W2A4	0.616 ± 0.149	0.165 ± 0.045	0.243 ± 0.090	0.433 ± 0.086	0.100 ± 0.037	0.103 ± 0.029	$\mathbf{0.276 \pm 0.072}$

D.2 Video Generation on Large Models

We evaluate the two largest video diffusion transformers in our study, Wan 14B (Wan et al., 2025) and HunyuanVideo (Kong et al., 2024), at W4A4. Table 6 reports VBench per-dimension scores. On Wan 14B, OrbitQuant achieves the best PTQ result on seven of the eight dimensions and remains close to BF16 across several metrics. The rotation and smoothing baselines degrade substantially on motion and scene-related metrics. On HunyuanVideo, we compare against DVD-Quant (Li et al., 2025b), a method designed specifically for video DiT quantization, with all baseline and DVD-Quant numbers taken from the DVD-Quant paper. Although OrbitQuant is calibration-free and uses the same recipe across all backbones, it remains competitive with DVD-Quant and outperforms it on imaging quality, motion smoothness, and scene. These results further demonstrate that the same distributional codebook transfers to large video DiTs without per-model tuning.

D.3 Comparison with QAT

Table 7 compares OrbitQuant with QAT methods that fine-tune the quantized model. As a calibration-free PTQ method, OrbitQuant generally trails the strongest QAT baseline, QVGen (Huang et al., 2025b), whose fine-tuned objective directly targets video generation quality. Nevertheless, OrbitQuant remains competitive across most VBench dimensions and surpasses all QAT methods on several metrics, achieving the best Subject Consistency, Scene, and Overall Consistency on Wan 2.1-1.3B. These results show that OrbitQuant can remain competitive with fine-tuned quantizers without gradient-based optimization or per-model tuning.

D.4 Robustness

OrbitQuant is calibration-free, so the only stochastic parts of the pipeline are the RPBH rotation and sampling noise. To confirm that the results are not an artifact of a single random draw, we rerun the full pipeline with three random seeds and report the mean and standard deviation of GenEval. Ta-

ble 8 covers the three image models at W4A4 and W2A4. At W4A4, the Overall standard deviation is at most 0.012 across all three models. The FLUX models remain similarly stable at W2A4, with Overall standard deviations below 0.015, while Z-Image-Turbo shows larger variance. Overall, the analytic codebook and random rotation yield consistent results across seeds.

D.5 Fake-Quantization Overhead Comparison

We measure end-to-end latency and peak memory on FLUX.1-dev (NVIDIA H100, 1024^2, 50 steps, guidance 3.5) and Wan 2.1-1.3B (480×832, 81 frames, 50 steps, CFG 5.0). All methods run under fake quantization with the matmul in BF16, so the comparison measures quantization overhead rather than realized speedup. As shown in Figure 8, OrbitQuant has the lowest overhead on both modalities. SmoothQuant (Xiao et al., 2023), QuaRot (Ashkboos et al., 2024), and ViDiT-Q (Zhao et al., 2024) run $1.09\times$, $1.28\times$, and $1.40\times$ slower on image, respectively, with the same ordering on video. OrbitQuant uses a fixed shared-codebook lookup for activation quantization, avoiding additional input-dependent range estimation or rescaling at runtime. On video, the lookup indices and gather tensors increase peak memory slightly over QuaRot and SmoothQuant (20.3 vs. 19.3 GB), while remaining below ViDiT-Q.

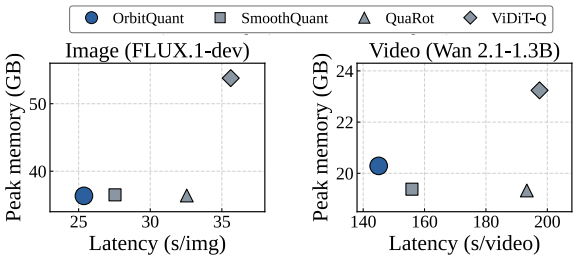


Figure 8: Fake-quantization overhead on FLUX.1-dev and Wan 2.1-1.3B.

E Limitations and Future Work

OrbitQuant incurs the online rotation cost common to rotation-based quantization, although this overhead is substantially reduced by our fused RPBH implementation. Another challenge is preserving the non-uniform Lloyd–Max centroids during integer execution. We find that representing the centroids in INT8 preserves generation quality, while lower-precision centroid representations can cause substantial degradation. This creates a gap between low-bit code storage and efficient low-bit arithmetic, since the non-uniform centroids must still be represented with sufficient precision during execution. Developing kernels that more directly exploit low-bit Lloyd–Max codebooks without sacrificing generation quality is therefore an important direction for future work.

F Additional Qualitative Results

We provide additional qualitative comparisons on FLUX.1-dev across W4A4, W3A3, and W2A4 settings. All methods are evaluated using the same prompts and generation settings. As the precision decreases, quantization artifacts become increasingly pronounced for the baselines, whereas OrbitQuant better preserves visual quality, prompt alignment, and overall image structure.

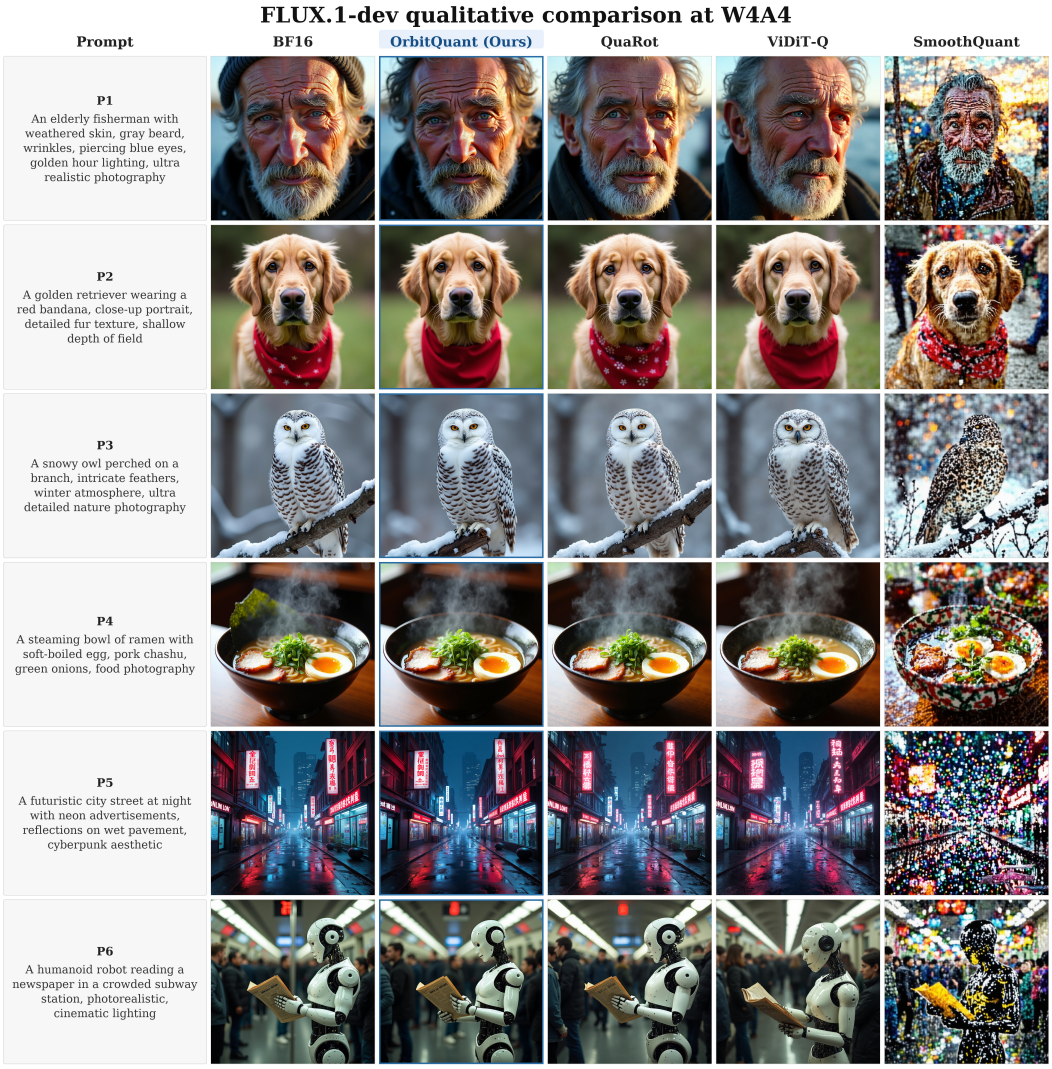


Figure 9: Qualitative comparison on FLUX.1-dev at W4A4. Each row uses the same prompt, while the columns compare BF16, OrbitQuant, QuaRot, ViDiT-Q, and SmoothQuant under identical generation settings.

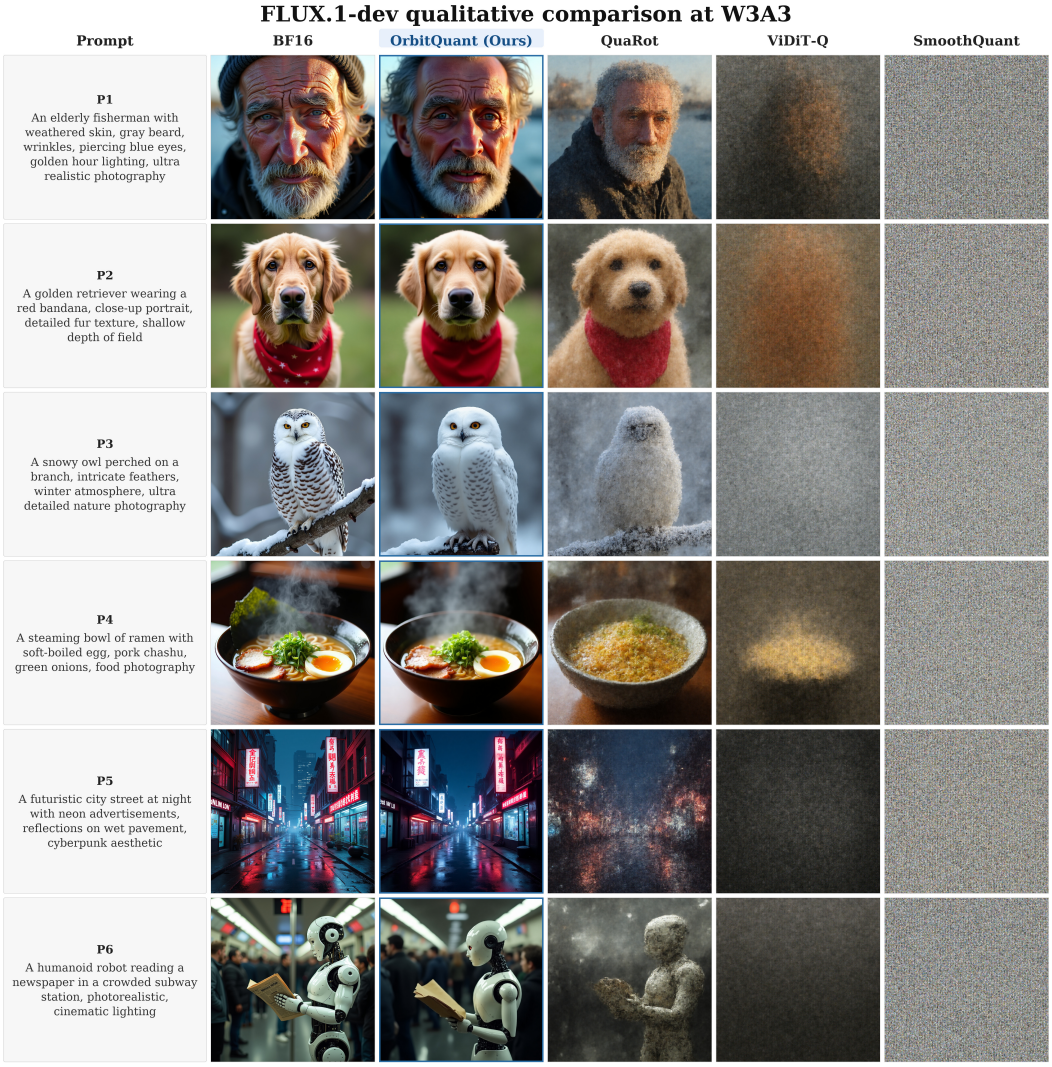


Figure 10: Qualitative comparison on FLUX.1-dev at W3A3.



Figure 11: Qualitative comparison on FLUX.1-dev at W2A4.