Exact Channel Decoupling via Joint Diagonalization and Uniform Splicing for Diffusion Transformer Quantization

Bingyao Yu Yijin Liu Qinkai Xu Li Li† Yuxiang Fu†
Nanjing University
{bingyaoyu, yijinliu, qinkaixu}@smail.nju.edu.cn,
{lili, yuxiangfu}@nju.edu.cn
†Corresponding Author.

Abstract

Diffusion Transformers (DiTs) have emerged as the state-of-the-art architecture in high-fidelity generative modeling, but their massive computational demands and multi-step inference process limit their deployment in practical scenarios. Post-training quantization (PTQ) offers a promising solution for accelerating inference, but DiTs suffer from severe activation outlier issues, which can lead to catastrophic accuracy loss in integer formats. Although some recent outlier mitigation methods employ orthogonal transformations or diagonal scaling to smooth the activation distribution, they either solve the transformation matrix solely for activations or fail to fully balance the inter-channel variance between activations and weights. Additionally, the construction of the calibration set by taking the weighted average of the activation values for each time step introduces spurious temporal correlations. To address these issues, we propose a joint diagonalization of the activation-weight covariance matrix based on the solution of a generalized eigenvalue problem (GEVP). We also transform the feature fusion of the calibration set's temporal dimension into probabilistic sampling and assembly in the spatial dimension using Uniform Token Splicing (UTS). Specifically, compared to suboptimal baseline methods, our approach reduces the FID by 16.06 and 6.49 on both the W3A4-quantized DiT-XL/2 and PixArt-Σ models, respectively. This demonstrates that our method successfully suppresses outliers in activations and weights, maintaining DiT's performance under low-bit quantization conditions. Code is available at `https://github.com/KdxftXr7/JDUS-DiT`.

1 Introduction

Diffusion Transformers (DiTs) [24] have replaced U-Nets [25] in image generation, yet their large scale and iterative denoising hinder edge and real-time deployment. Quantization is a promising compression method [12, 1]. DiTs exhibit substantial spatial and temporal distribution shifts, and fine-tuning demands thousands of GPU-hours. Post-training quantization [34, 4, 22] uses only a small calibration set to tune quantization parameters, achieving efficient compression without quality loss and thus ideal for DiT deployment.

Although ultra-low-bit quantization is highly attractive, extreme activation outliers often lead to severe accuracy collapse. Existing PTQ paradigms primarily including second-order error compensation [8, 20] and equivalent transformations [35, 1, 17, 38] attempt to mitigate this. However, a major challenge remains: current methods fail to simultaneously and effectively reduce the quantization difficulty of both activations and weights. Methods relying on second-order information often suffer

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

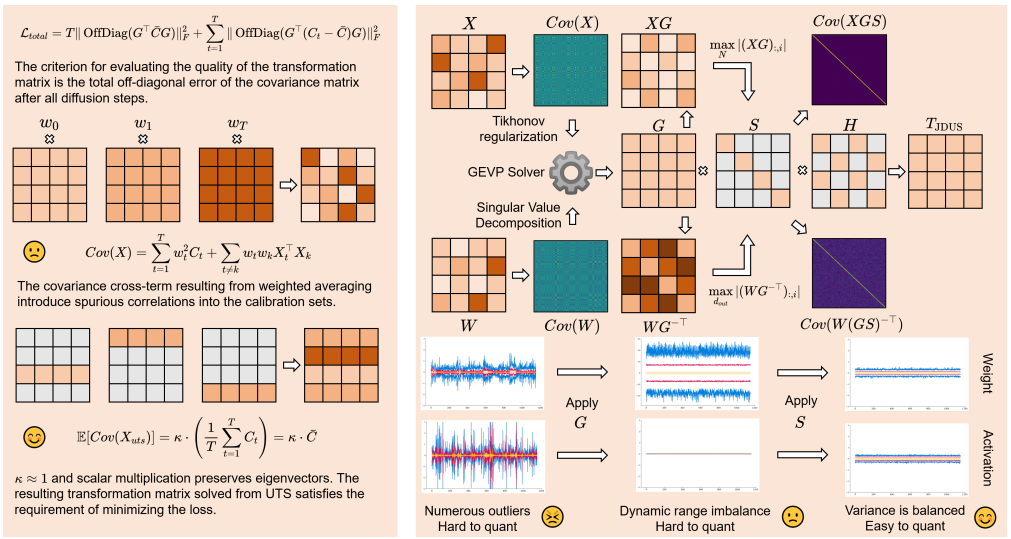


Figure 1: This figure illustrates the algorithmic process described in this paper. It uses the UTS method to obtain a calibration set with a covariance expectation of $\kappa\bar{C}$, employs the solution to GEVP to simultaneously diagonalize the covariance matrices of activations and weights, and addresses the dynamic range imbalance issue using a lossless diagonal scaling method. And get variance equalization through the fusion of Hadamard matrices. Here, S denotes the diagonal scaling matrix, and H denotes the Hadamard matrix.

from numerical instability induced by long-tail outliers. Conversely, equivalent transformation approaches typically involve compromises: they either constrain their transformation space to simple scaling or strict orthogonality, preventing thorough channel decorrelation in complex distributions, or only optimize for activation variance while neglecting the subsequent impact on weights, leading to severe performance degradation under low-bit weight constraints. A second critical challenge lies in the suboptimal construction of calibration sets for dynamic distributions. Unlike large language models, the activation distributions in DiTs fluctuate significantly across iterative denoising steps [4]. Existing attempts to aggregate diverse time steps [38] rely on weighted averages of activations. It introduces spurious cross correlations in the covariance calculation, inadvertently offsetting necessary outliers and limiting the fidelity of the results.

To address these issues, as shown in Figure 1, we propose a novel method for calculating transformation matrices and a method for collecting calibration sets without cross-terms. We utilize the mathematical model of the generalized eigenvalue problem (GEVP) to find the transformation matrix G such that both the transformed activation covariance matrix and the weight covariance matrix become diagonal matrices. Additionally, we introduce truncated singular value decomposition to avoid the rank-deficiency trap in certain weight covariance matrices. We apply lossless scaling to the weight and activation matrices after the basic transformation solved by GEVP, balancing the dynamic ranges of activation and weights without destroying the diagonalized covariance matrix. Finally, apply the Hadamard transform to fully balance the inter-channel variances of weight and activation. By constructing a calibration set capable of aligned temporal mean covariances through Uniform Token Splicing (UTS), we eliminate cross-term contamination caused by feature stacking with minimal GPU memory overhead. The calculation of the transformation matrix based on the calibration set is performed offline, so it does not affect inference speed.

In summary, the core contributions of this paper are as follows:

1. We achieved joint diagonalization of the covariance matrices of activations and weights by solving the GEVP. We avoid the rank-deficiency trap and balance the dynamic ranges through truncated singular value decomposition and lossless scaling. Achieving variance balancing by fusing Hadamard matrices.
2. We propose the UTS method for constructing calibration sets, which avoids spurious cross correlations while aligning the temporal mean covariance.
3. We evaluated our method on the DiT-XL/2 and PixArt-Σ models, achieving significantly better performance than existing baseline models under extremely low-bit quantization settings.

2 Background and Related Work

2.1 Diffusion Transformers

The shift from convolution-based U-Net backbones [25] to Transformer architectures [31] has significantly enhanced the scalability of generative models. The Diffusion Transformer (DiT) [24] introduced a modular design utilizing Vision Transformer blocks [7], where the DiT-XL/2 has become a widely adopted strong baseline due to its impressive performance-scale trade-off. To address the high training costs of DiTs, PixArt-α [3] introduced an efficient transformer-based T2I architecture with a three-stage training decomposition. Its advanced version, PixArt-Σ [2], subsequently pioneered a weak-to-strong training strategy, achieving 4K generation with significantly reduced FLOPs. However, as DiT models become the backbone of high-fidelity generation, their massive memory footprint and attention-driven computational complexity create a pressing need for efficient deployment.

2.2 Model Quantization

Model quantization[18] is crucial for deploying neural networks on resource-constrained devices. By compressing high-precision weights and activations into low-bit formats, it substantially reduces memory footprint, latency, and energy consumption. Quantization methods primarily fall into two paradigms: quantization-aware training (QAT)[40, 32, 11] and post-training quantization (PTQ)[8, 35, 39, 34, 4, 19]. Although QAT mitigates quantization errors by simulating noise during training, it requires full dataset access and incurs considerable computational overhead. Conversely, PTQ quantizes pre-trained models directly using a small calibration set without retraining. This efficiency and ease of deployment make PTQ highly favorable for practical applications.The core of PTQ relies on a unified affine quantization scheme that maps a floating-point value $x \in \mathbb{R}$ to an integer x_q:

$$x_q = \text{clamp}\left(\left\lfloor \frac{x}{s} \right\rceil + z, q_{\min}, q_{\max}\right), \tag{1}$$

where $s > 0$ and $z \in \mathbb{Z}$ denote the scaling factor and zero-point, respectively. $\lfloor \cdot \rceil$ rounds to the nearest integer, and clamp bounds the result within the quantization range $[q_{\min}, q_{\max}]$.

PTQ4DiT [34] is one of the pioneering works dedicated to PTQ for DiTs by introducing channel significance balancing and Spearman ρ-guided temporal calibration, achieving W4A8 quantization but relying on labeled calibration data and exhibiting bias under sharp temporal distribution shifts. DiTAS [6] eliminates calibration data with temporal aggregation smoothing and a training-free low-rank correction scheme, yet its static smoothing factor limits adaptation to evolving activation dynamics, degrading low-bit fidelity. LRQ-DiT [37] employs twin-log quantization and adaptive rotation to accommodate long-tailed weight distributions; however, reliance on non-uniform compute units hinders hardware compatibility. HQ-DiT [22] leverages FP4 and Hadamard transforms to suppress outliers, but lacks widespread native support on commodity and edge hardware, impeding practical deployment. Q-DiT [4] mitigates cross-channel and temporal variance via automatic granularity allocation and dynamic activation quantization, though per-batch uniform quantization remains insufficient for heavy-tailed outliers and granularity search incurs high calibration cost. ViDiT-Q [39] improves generation fidelity by detecting high-sensitivity layers and combining decoupling metrics, yet sensitivity distribution is model- and data-dependent, requiring architecture-specific recalibration and lacking a unified end-to-end optimization framework. VETA-DiT [38] balances inter-channel variance through a KLT-enhanced Hadamard transform and incoherence-aware adaptation for temporally varying data distributions, but excludes weights from rotation optimization, leading to variance imbalance after weight rotation as shown in Figure 3, and weighted averaging introduces covariance cross-terms that degrade the representativeness of the calibration set.

3 Method

3.1 Joint Diagonalization of the Covariance Matrix

By analyzing the distribution characteristics of activation values across layers in DiT-XL/2, it can be observed that certain channels exhibit excessively large outliers, as shown in Figure 2. There are significant differences in variance between channels [1], which can lead to substantial quantization

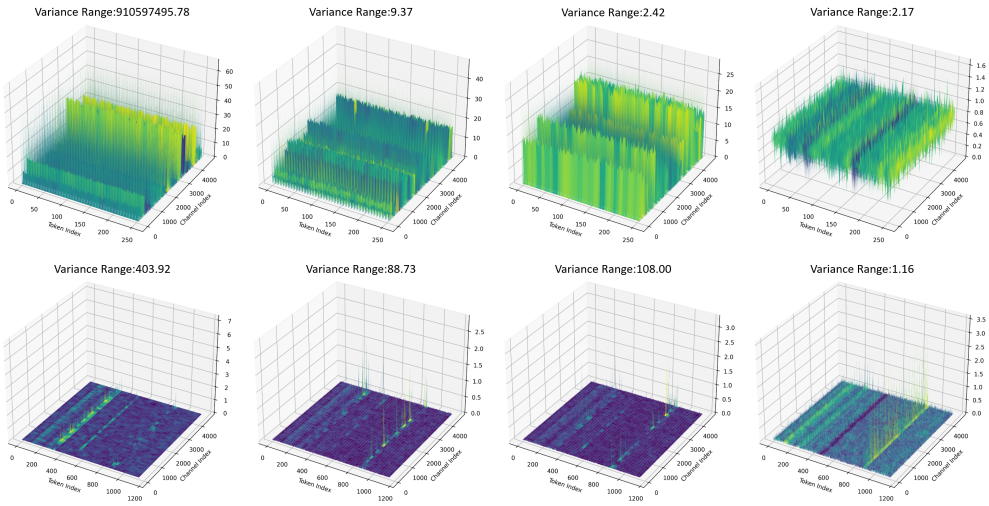


Figure 2: From left to right: activation (**Up**) and weight (**Down**) distributions and corresponding inter-channel variance ranges of blocks.27.mlp.fc2 for the original, Hadamard transform, KLT-enhanced Hadamard transform, and our method, respectively.

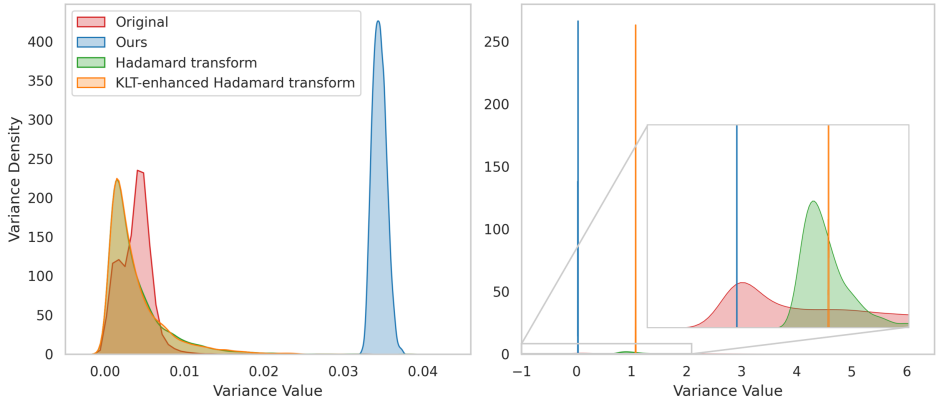


Figure 3: **(Left)** blocks.27.mlp.fc2 weight variance distribution between four methods. **(Right)** blocks.27.mlp.fc2 activation variance distribution between four methods. Variance density reflect the concentration of variance across channels. Our method successfully balances the variance across activation channels while also balancing the variance across weight channels.

clipping errors and result in unacceptable quantization quality. Furthermore, these extreme outliers are often not noise but structural outliers. They frequently carry critical information or extremely important global features.

Previous work [38] has demonstrated that diagonalizing the covariance matrix of a matrix using a transformation and then applying the Hadamard transform can balance the variance across channels. Since the transformation is derived solely from the activation calculation, a corresponding transformation must also be applied to the weights to ensure computational correctness. However, this transformation of the weights was not carefully designed and is highly likely to cause an imbalance in the variance across weight channels. As shown in Figure 3, this increases the difficulty of weight quantization, leading to quantization failure under extremely low bit widths such as W3A4.

Consider a bias-free linear layer defined by $Y = XW^\top$, where $X \in \mathbb{R}^{N\times d}$ is the activation matrix and $W \in \mathbb{R}^{d_{out}\times d}$ is the weight matrix. An equivalent transformation introduces a non-singular matrix $G \in \mathbb{R}^{d\times d}$ to reshape the feature space while preserving the output: $Y = (XG)(G^{-1}W^\top) = \tilde{X}\tilde{W}^\top$. Here, $\tilde{X} = XG$ and $\tilde{W} = WG^{-\top}$ denote the transformed activations and weights. Existing methods often restrict G to an orthogonal matrix[1, 17, 38]. However, orthogonal transformations

merely rotate the vector space, which fails to eliminate the inter-channel correlations in both activation and weight. To decouple the channels, we relax G to a general invertible matrix. We define C as the symbol for the matrix covariance; Our objective is to find an optimal G that diagonalizes the transformed covariances $C_{\tilde{X}}$ and $C_{\tilde{W}}$ simultaneously. Considering the GEVP [33]:

$$C_W^{-1}\mathbf{v} = \lambda C_X \mathbf{v}. \tag{2}$$

Since a covariance matrix is a real symmetric matrix, and Tikhonov regularization [30] can guarantee its positive definiteness, we are able to solve for d eigenvalues and eigenvectors. Let $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_d)$ be the diagonal matrix of the generalized eigenvalues, and let the columns of $G = [\mathbf{v}_1, \ldots, \mathbf{v}_d]$ be the corresponding generalized eigenvectors. The GEVP can be compactly written in matrix form as:

$$C_W^{-1}G = C_X G\Lambda. \tag{3}$$

By the spectral theorem for symmetric positive-definite matrix pencils, the eigenvector matrix G are C_X-orthonormalized. Consequently, the transformation matrix naturally satisfies the weighted orthogonality condition, directly yielding

$$C_{\tilde{X}} = G^\top C_X G = I, \tag{4}$$

$$C_{\tilde{W}} = G^{-1} C_W G^{-\top} = \Lambda^{-1}. \tag{5}$$

This analytical closed-form solution forces $C_{\tilde{X}}$ into the identity matrix and $C_{\tilde{W}}$ into a diagonal matrix, thereby eradicating all off-diagonal cross-channel correlations in both spaces simultaneously.

However, in the actual model layers, there is a problem of mismatched projection dimensions. When the output dimension of the weight matrix is smaller than the input dimension, let the rank of W be $r \leq d_{out} < d$. Accordingly, the covariance matrix of the weights $C_W \in \mathbb{R}^{d\times d}$ inherits such a rank bound and is consequently rank-deficient.

The eigenvalue decomposition of C_W can be expressed as $C_W = \sum_{i=1}^{d} \lambda_i u_i u_i^\top$, where for $i > r$ the eigenvalues λ_i are close to zero, manifesting as floating-point noise ϵ. During the solution of the GEVP involving C_W^{-1}, the condition numbers on these noise dimensions are amplified to $\mathcal{O}(\frac{1}{\epsilon})$, resulting in an excessively large norm for the transformation matrix G, which leads to a collapse of the quantization dynamic range. Therefore, we introduce truncated singular value decomposition [10, 9]. We set a relative truncation threshold τ and construct the truncated covariance matrix $\tilde{C}_W$, where $\lambda_i^* = \max(\lambda_i, \tau \cdot \lambda_{max}), U = [u_1, \ldots, u_d]$:

$$\tilde{C}_W = U\text{diag}(\lambda_1^*, \lambda_2^*, \ldots, \lambda_d^*)U^\top. \tag{6}$$

This operation reduces the norm of the transformation matrix G to a reasonable level, thereby stabilizing the solution process for the GEVP.

After solving for the basic transformation matrix G via GEVP, although we have achieved decoupling between channels, the issue of dynamic range imbalance still persists in practice. This is because, when solving GEVP, we do not impose any constraints on the dynamic range of the matrix after transformed. This will reduce the resolution allocation of the quantizer. To overcome this limitation, we propose a lossless diagonal balancing method. We introduce a diagonal suppression matrix $S = \text{diag}(s_1, s_2, \ldots, s_d)$ and a control factor $\alpha \in (0, 1)$ to perform asymmetric dynamic range allocation. For the ith channel, the analytical solution for its balancing coefficient s_i is:

$$s_i = \frac{\max_{d_{out}} |(WG^{-\top})_{:,i}|^\alpha}{\max_N |(XG)_{:,i}|^{1-\alpha}}. \tag{7}$$

Since the product of diagonal matrices is still a diagonal matrix, the decorrelation property is preserved without loss of information.

By applying the GS transformation, we can eliminate the linear correlation between the activation and weight channels. By combining this with the Hadamard transform, we obtain the final transformation matrix $T_{\text{JDUS}} = G \cdot S \cdot H$. This achieves complete variance balancing between the activation and weight channels.

3.2 Uniform Token Splicing

The highly dynamic generation sequence of DiTs poses a significant challenge for constructing the calibration covariance. The inference of diffusion models is a dynamic process spanning T timesteps. The criterion for evaluating the quality of the transformation matrix G is the total off-diagonal error of the covariance matrix after T diffusion steps. We define the total error loss as the sum of errors across all timesteps. Let C_t denote the covariance matrix of activations or weights at timestep t, and $\text{OffDiag}(X)$ denote setting the diagonal elements of matrix X to 0, retaining only the off-diagonal elements:

$$\mathcal{L}_{total} = \sum_{t=1}^{T} \|\text{OffDiag}(G^\top C_t G)\|_F^2. \tag{8}$$

Let the timestep-averaged covariance be $\bar{C} = \frac{1}{T}\sum_{t=1}^{T} C_t$. According to the matrix variance decomposition identity, we obtain:

$$\mathcal{L}_{total} = T\| \text{OffDiag}(G^\top \bar{C} G)\|_F^2 + \sum_{t=1}^{T} \| \text{OffDiag}(G^\top (C_t - \bar{C}) G)\|_F^2. \tag{9}$$

The second term is associated with individual timesteps during the diffusion process, making it difficult to minimize with a single transformation matrix. Therefore, we consider minimizing the first term to suppress the global error.

VETA-DiT [38] calculates timestep-level incoherence score weights $w_t \in (0, 1)$ and performs a weighted average of activations across different timesteps to fuse temporal information: $X_{veta} = \sum_{t=1}^{T} w_t X_t$. However, this feature-level superposition introduces cross-terms. When calculating the calibration covariance using X_{veta}, its expansion is:

$$C_{veta} = \left(\sum_{t=1}^{T} w_t X_t\right)^\top \left(\sum_{k=1}^{T} w_k X_k\right) = \sum_{t=1}^{T} w_t^2 C_t + \sum_{t \neq k} w_t w_k X_t^\top X_k. \tag{10}$$

The rotation matrix G_{veta} satisfies: $G_{veta} = \arg\min_G \| \text{OffDiag}(G^\top C_{veta} G)\|_F^2$. Deviating from the optimization target $\bar{C}$. Furthermore, in the authentic DiT inference process, the generation trajectory is a discrete sequence following a Markov chain [15, 28]. The channel features at timestep t are not directly combined with those at timestep k via inner products. The cross-term $X_t^\top X_k$ appearing in the above equation is an additionally introduced spurious correlation. This not only causes the solved rotation matrix G_{veta} to deviate from the global optimum, but also, since $w_t \in (0, 1)$, the term w_t^2 quadratically reduces the covariance contribution of the true timesteps, leading to an overall attenuation of the covariance.

Theoretically, the ideal approach to obtain the globally optimal decorrelated is full concatenation. This method constructs a massive calibration set containing complete timestep information by directly concatenating the full activation tensors of all timesteps along the sequence dimension. Its covariance is algebraically equivalent to the global temporally-averaged covariance $\bar{C}$ of authentic inference. However, full concatenation causes the token dimension of the calibration set tensors to explode, incurring prohibitive computational overhead.

To achieve zero cross-term pollution and align with the global temporally-averaged covariance at an extremely low computational cost, we propose Uniform Token Splicing (UTS). Given a total calibration capacity quota N, UTS ignores timestep weight differences based on heuristic metrics and performs equal-share, uniform random sampling without replacement across all T timesteps. Specifically, for each timestep candidate tensor $X_t \in \mathbb{R}^{N \times d}$, we uniformly and randomly sample a subset of $n = \frac{N}{T}$ tokens to form a slice $A_t \in \mathbb{R}^{n \times d}$. Subsequently, the authentic slices from all timesteps are concatenated along the token dimension to construct a minimal and pure final calibration set $X_{uts} = \text{Concat}(A_1, A_2, \ldots, A_T) \in \mathbb{R}^{N \times d}$. Since X_{uts} is vertically concatenated from independent row slices, according to the block matrix multiplication rule, its second moment is the arithmetic sum of the sub-slice moments:

$$X_{uts}^\top X_{uts} = \sum_{t=1}^{T} A_t^\top A_t. \tag{11}$$

Let $\bar{x}_{uts} \in \mathbb{R}^{1\times d}$ be the channel-wise mean of the concatenated tensor X_{uts}. Introducing the finite population sampling theorem to account for finite-sample mean drift across independent slices, the expectation of its covariance $C_{X_{uts}}$ incurs a marginal scalar scaling factor $\kappa = \left(1 - \frac{N-n}{N(N-1)}\right)$:

$$\mathbb{E}[C_{X_{uts}}] = \mathbb{E}\left[\frac{1}{N} X_{uts}^{\top} X_{uts}\right] - \mathbb{E}[\bar{x}_{uts}^{\top} \bar{x}_{uts}] = \kappa \cdot \left(\frac{1}{T}\sum_{t=1}^{T} C_t\right) = \kappa \cdot \bar{C}. \tag{12}$$

Crucially, since N is practically large, $\kappa \approx 1$ and scalar multiplication preserves eigenvectors, the resulting transformation matrix solved from UTS satisfies the requirement of minimizing the loss.

For generative models with Markovian sequential denoising characteristics, if and only if UTS is adopted to construct the calibration set, the mathematical expectation of its calibration covariance can represent the timestep-averaged covariance $\bar{C}$ with a small computational cost. While ensuring zero cross-term pollution, UTS avoids the extreme value neutralization phenomenon caused by feature superposition, providing an unbiased data source for solving the globally optimal decorrelated transformation. Any temporal asymmetric weighting based on the degree of abrupt changes will inevitably cause the transformation matrix to overfit local distributions. Through the UTS method, we extract the optimal physical representation containing the complete temporal covariance characteristics of DiTs at an extremely low computational cost.

4 Experiments

4.1 Experimental Settings

We examine four representative baseline methods: PTQ4DiT[34], Q-DiT[4], ViDiT-Q[39] and VETA-DiT[38], and comprehensively evaluate our approach under a variety of experimental settings. To ensure fair and consistent comparisons, we exclude the mixed-precision quantization strategy of ViDiT-Q, which tends to quantize most linear layers to INT8, as this would compromise the fairness of the comparison under extremely low-bit settings such as W3A4 and W4A4.

We applied our method to the pre-trained model DiT-XL/2[24] to generate 256×256 images on the ImageNet dataset[26]. Sampling was performed using a DDIM sampler[29] with step sizes set to 50 and 100. During sampling, the CFG[16] was set to 1.5. We use FID[14], sFID, IS[27], and Precision as evaluation metrics for image generation quality. The ADM evaluation toolkit[5] is used to compute these metrics. To demonstrate the versatility of our method, we further evaluated it on the PixArt-Σ model[2]. Image generation was performed on the COCO dataset[21] using a 20-step DPM solver[23], with CFG set to 4.5. For the PixArt-Σ model, we additionally employed the ClipScore[13] and ImageReward[36] metrics for evaluation.

We selected two extremely low bit-width settings: W3A4 and W4A4. For the DiT-XL/2 model, we quantized all attention and mlp layers. For the PixArt-Σ model, we quantized all ff.net and attention layers. For both weights and activations, we employed a grouped asymmetric quantization method. We used the same quantization bit-width for all quantized layers.

4.2 Quantization Performance

Table 1 shows the experimental results of DiT-XL/2 on the ImageNet dataset at a resolution of 256×256. Under the low bit-width quantization setting W4A4, our method also performs close to full-precision levels, with FID values at 50 and 100 time steps reduced by 1.67 and 1.20, respectively, compared to the previous state-of-the-art work. Even under the extremely low bit-width quantization setting W3A4, our method still produces distortion-free and visually plausible images, outperforming all baseline models across all metrics. Compared to the second-best method, FID decreased by 16.06, while IS improved by 38.40. Relying solely on solutions for activation values, the previous method[38] was unable to effectively handle scenarios with extremely low weight quantization bitwidth, resulting in degraded image generation quality.

Table 2 reports the experimental results on the PixArt-Σ model. Under the W3A4 and W4A4 settings, our method achieved FID reductions of 6.49 and 5.61, respectively, compared to the strongest baseline model. Using UTS and incorporating weights into the transformation calculation enables our method to demonstrate excellent performance under both extreme bit-width conditions.

Table 1: Performance comparison of DiT-XL/2 on the ImageNet 256×256.

Model	Bit-width	Method	IS(↑)	FID(↓)	sFID(↓)	Precision(↑)
DiT-XL/2 steps=100	W16A16	FP	168.96	4.59	17.54	0.79
	W4A4	PTQ4DiT	2.42	274.59	117.24	0.08
		Q-DiT	2.20	243.03	410.06	0.02
		ViDiT-Q	60.29	37.24	33.18	0.45
		VETA-DiT	137.98	7.22	18.90	0.74
		Ours	**147.49**	**6.02**	**18.55**	**0.75**
	W3A4	PTQ4DiT	2.33	281.14	121.25	0.08
		Q-DiT	1.82	257.31	406.51	0.01
		ViDiT-Q	1.74	288.51	258.22	0.01
		VETA-DiT	76.49	24.39	27.69	0.55
		Ours	**103.86**	**13.00**	**21.60**	**0.65**
DiT-XL/2 steps=50	W16A16	FP	164.70	4.86	17.64	0.80
	W4A4	PTQ4DiT	2.33	281.42	119.44	0.07
		Q-DiT	2.02	251.11	409.00	0.01
		ViDiT-Q	54.37	43.72	37.43	0.40
		VETA-DiT	135.66	7.90	19.11	0.73
		Ours	**145.96**	**6.23**	**18.57**	**0.75**
	W3A4	PTQ4DiT	2.22	276.14	125.79	0.08
		Q-DiT	1.88	250.08	392.15	0.01
		ViDiT-Q	1.63	292.55	281.24	0.01
		VETA-DiT	67.11	29.38	26.56	0.52
		Ours	**105.51**	**13.32**	**21.22**	**0.65**

'(WxAy)' indicates that the weights and activations are quantized to x-bit and y-bit, respectively.

Table 2: Text-to-image generation performance of the PixArt-Σ model on the COCO dataset.

Bit-width	Method	IS(↑)	FID(↓)	sFID(↓)	CLIP(↑)	IR(↑)
W16A16	FP	38.25	59.68	294.69	0.27	0.89
W4A4	Q-DiT	18.78	106.60	**281.09**	0.24	-0.44
	ViDiT-Q	13.77	165.23	332.24	0.22	-0.88
	VETA-DiT	35.56	65.22	305.36	0.26	0.88
	Ours	**38.09**	**59.61**	300.91	**0.27**	**0.94**
W3A4	Q-DiT	7.43	183.73	301.94	0.21	-1.43
	ViDiT-Q	1.48	481.11	405.13	0.12	-2.27
	VETA-DiT	31.71	67.51	**280.77**	0.26	0.61
	Ours	**36.86**	**61.02**	300.98	**0.26**	**0.85**

4.3 Ablation Studies

Table 3: Ablation study on ImageNet 256×256 with W3A4.

Method	IS	FID	sFID	Precision
FP	164.61	4.86	17.64	0.80
Baseline	2.10	242.85	410.35	0.01
+ KLTH-INCO	67.11	29.38	26.56	0.52
+ KLTH-IPTS	55.38	36.60	62.29	0.45
+ KLTH-UTS	70.81	25.43	26.93	0.54
+ GEVP-INCO	79.02	23.18	26.79	0.56
+ GEVP-IPTS	93.37	17.83	24.18	0.61
+ GEVP-UTS	105.51	13.32	21.22	0.65

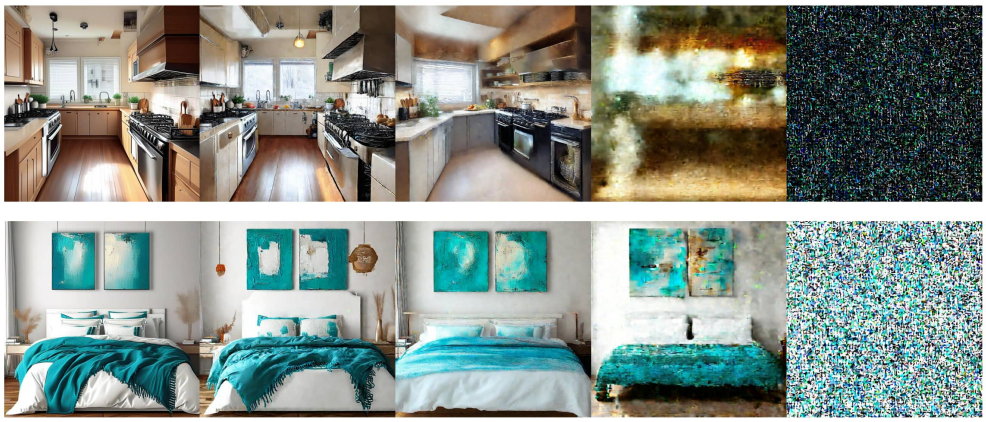


Figure 4: A qualitative comparison of image quality generated by various methods under W3A4 quantization on the PixArt-Σ Model

To thoroughly analyze the impact of different transformation matrix computation methods and calibration set construction paradigms on quantization performance, we conducted ablation experiments using the DiT-XL/2 model with the DDIM solver over 50 sampling iterations on the ImageNet 256×256 dataset under W3A4. We focused on isolating and comparing two transformation matrix computation methods: the Karhunen-Loève transform enhanced Hadamard transform (KLTH)[38] and GEVP with three temporal step fusion strategies: the Incoherent-Aware Adaptive Method (INCO)[38], the Incoherent Proportional Token Splicing (IPTS), and the UTS. Among these, IPTS uses the same incoherent weights as the INCO method to allocates a different number of splicing calibration sets for different time steps. Quantitative results are shown in Table 3. Directly performing W3A4 quantization causes the model's generative capability to collapse. After introducing the KLTH orthogonal transformation and the INCO method, the model's FID reduced to 29.38, partially restoring performance. However, the feature space superposition in INCO introduced spurious cross-term across time steps, resulting in incomplete channel decorrelation and limiting improvements in image generation quality. The UTS method proposed in this paper achieved excellent performance under all settings. Combined with KLTH, UTS further reduced the FID to 25.43 and improved the IS to 70.81. When combined with the GEVP method proposed in this paper, it achieved optimal results of FID 13.32 and IS 105.51. GEVP's approach to calculating transformation matrices for both activations and weights balances the variance across activation channels while simultaneously balancing the variance across weight channels. Through token-level assembly, UTS eliminates cross-term contamination in INCO, providing the purest feature inputs for transformation matrix computation. This achieves globally optimal channel decorrelation.

5 Conclusion

This paper re-examines the core challenges in ultra-low-bit quantization of DiT and proposes a method that utilizes the solution to GEVP to simultaneously diagonalize the covariance matrices of weights and activations. By employing truncated singular value decomposition and scaling, it prevents the collapse of the dynamic range of weights and activations. This enables the subsequent application of the Hadamard transform to achieve variance balancing between channels for both weights and activations. Through UTS, we align the temporal mean covariance, while avoiding the introduction of spurious temporal correlations. Extensive experiments demonstrate that our method enables mainstream DiTs models to achieve a significant performance leap under ultra-low-bit quantization settings. Furthermore, the transformation matrix supports zero-overhead integration with existing networks.

6 Acknowledgements

This work was supported in part by the National Key Research and Development Program of China under Grant No. 2023YFB2806802 and in part by Suzhou's "Jiebang Guashuai" Project for Key Core Technologies under Grant No. SYG2024134. The authors are grateful for the help from the Interdisciplinary Research Center for Future Intelligent Chips (Chip-X) and Yachen Foundation.

References

[1] Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian 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.

[2] 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*, pages 74–91, 2024.

[3] Junsong Chen, Jincheng YU, Chongjian GE, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-\alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 57611–57640, 2024.

[4] 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. *arXiv preprint arXiv:2406.17343*, 2024.

[5] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[6] 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)*, pages 4606–4615. IEEE, 2025.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, G Heigold, S Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[8] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan-Adrian Alistarh. Optq: Accurate post-training quantization for generative pre-trained transformers. In *11th International Conference on Learning Representations*, 2023.

[9] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.

[10] Per Christian Hansen. *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*. SIAM, 1998.

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

[12] Yefei He, Luping Liu, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Ptqd: Accurate post-training quantization for diffusion models. In *Advances in Neural Information Processing Systems 2023*, 2023.

[13] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528, 2021.

[14] 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.

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 6840–6851, 2020.

[16] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[17] 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.

[18] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713, 2018.

[19] Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Junxian Guo, Xiuyu Li, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. Svdquant: Absorbing outliers by low-rank component for 4-bit diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[20] 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.

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755, 2014.

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

[23] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: a fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 5775–5787, 2022.

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

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241, 2015.

[26] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[27] 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.

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

[29] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[30] Andrei Nikolaevich Tikhonov et al. Regularization of incorrectly posed problems. Soviet Mathematics Doklady, 1963.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[32] Haoxuan Wang, Yuzhang Shang, Zhihang Yuan, Junyi Wu, Junchi Yan, and Yan Yan. Quest: Low-bit diffusion model quantization via efficient selective finetuning. *arXiv preprint arXiv:2402.03666*, 2024.

[33] J.H. Wilkinson. *The Algebraic Eigenvalue Problem*. Monographs on numerical analysis. Clarendon Press, 1965.

[34] Junyi Wu, Haoxuan Wang, Yuzhang Shang, Mubarak Shah, and Yan Yan. Ptq4dit: Post-training quantization for diffusion transformers. In *Advances in neural information processing systems*, 2024.

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

[36] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: learning and evaluating human preferences for text-to-image generation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 15903–15935, 2023.

[37] 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*, 2025.

[38] Lin Yang, Li Li, Yuxiang Fu, et al. Veta-dit: Variance-equalized and temporally adaptive quantization for efficient 4-bit diffusion transformers. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[39] 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.

[40] Xingyu Zheng, Xianglong Liu, Haotong Qin, Xudong Ma, Mingyuan Zhang, Haojie Hao, Jiakai Wang, Zixiang Zhao, Jinyang Guo, and Michele Magno. Binarydm: Accurate weight binarization for efficient diffusion models. *arXiv preprint arXiv:2404.05662*, 2024.

A Additional Experimental Details

For all baseline methods, we used the same random seed, calibration set, and quantization bit-width. For both activations and weights, we adopted a quantization scheme with a group size of 128, using static quantization for weights and dynamic quantization for activations. In addition, we employed the GPTQ[8] method for offline weight calibration. All experiments were conducted on Nvidia A800 GPUs.

B Additional Experimental Results

B.1 Additional Experiment on PixArt-Σ

Table 4 reports more experimental results on the PixArt-Σ model. We evaluated DiTAS [6], LRQ-DiT [37], and SVDQuant [19] on PixArt-Σ under a unified protocol: the same PixArt-Σ checkpoint, 1,024 COCO prompts, seed 42, 20-step DPM-Solver, CFG 4.5, evaluated linear-layer scope, and evaluator. Retained each method's core algorithm and introduced only the interface-level changes needed to support PixArt-Σ and the requested bit widths.

Table 4: Text-to-image generation performance of the PixArt-Σ model on the COCO dataset.

Bit-width	Method	IS(↑)	FID(↓)	sFID(↓)	CLIP(↑)	IR(↑)
W16A16	FP	38.25	59.68	294.69	0.27	0.89
W4A4	DiTAS	1.23	473.26	438.82	0.13	-2.28
	LRQ-DiT	31.10	72.06	**283.60**	0.26	0.74
	SVDQuant	34.70	64.53	302.68	0.26	0.86
	Ours	**38.09**	**59.61**	300.91	**0.27**	**0.94**
W3A4	DiTAS	1.22	487.71	443.45	0.13	-2.28
	LRQ-DiT	17.36	119.37	303.38	0.24	-0.28
	SVDQuant	13.47	148.94	317.18	0.21	-0.93
	Ours	**36.86**	**61.02**	**300.98**	**0.26**	**0.85**

B.2 Sensitivity to τ , ϵ and α

The main experiments use the same $\tau = 10^{-3}$ and regularization multiplier $\epsilon = 10^{-5}$ for DiT-XL/2 and PixArt-Σ, with no model- or bit-width-specific tuning. We performed a one-factor-at-a-time sweep on PixArt-Σ W3A4 while holding all other settings fixed.

Table 5: Sensitivity analysis of τ and ϵ.

τ	ϵ	IS(↑)	FID(↓)	sFID(↓)	CLIP(↑)	IR(↑)
10^{-3}	10^{-6}	34.75	61.78	299.45	0.26	0.80
10^{-3}	10^{-5}	36.86	61.02	300.98	0.26	0.85
10^{-3}	10^{-4}	34.94	**59.85**	299.76	**0.27**	0.87
10^{-4}	10^{-5}	**37.12**	61.23	300.35	0.26	**0.88**
10^{-2}	10^{-5}	37.08	**59.85**	**294.83**	**0.27**	**0.88**

Table 5 reports experimental results. Across the tested two-order-of-magnitude ranges, FID varies from 59.85 to 61.78 and IS from 34.75 to 37.12; every setting remains better than VETA-DiT [38] at W3A4. Thus, the end-to-end generation quality is robust over this practical range, although the weakest regularization produces a modest degradation.

α is a model-level scalar controlling how diagonal scaling allocates dynamic range between transformed activations and weights; $\alpha = 0.5$ is the symmetric default. We use $\alpha = 0.5$ for DiT-XL/2. For PixArt-Σ, we performed a coarse W3A4 sweep over $\alpha \in \{0.50, 0.55, 0.59, 0.61\}$ and selected 0.61 by FID. The selected value is shared by all quantized layers and is not tuned layer by layer or separately for W3A4 and W4A4.

Table 6: Sensitivity analysis of α.

α	IS(↑)	FID(↓)	sFID(↓)	CLIP(↑)	IR(↑)
0.50	31.56	66.65	295.70	**0.26**	0.71
0.55	35.21	62.88	**294.90**	**0.26**	0.81
0.59	34.00	61.48	298.49	**0.26**	0.83
0.61	**36.86**	**61.02**	300.98	**0.26**	**0.85**

Results in Table 6 show that performance is moderately sensitive near the symmetric default: moving from 0.50 to 0.55 improves FID by 3.77. Within $[0.55, 0.61]$, however, FID varies by only 1.86 and no setting collapses. The secondary metrics favor nearby values (e.g., $\alpha = 0.55$ gives the best sFID), while $\alpha = 0.61$ gives the best FID, IS, and ImageReward.

B.3 UTS calibration seed variance

We repeated PixArt-Σ W3A4 with three calibration seeds while keeping the model, calibration budget, prompts, sampler, quantization configuration, and evaluator fixed. We report mean $\pm$ sample standard deviation:

Table 7: Calibration seed variance of UTS.

Method	IS(↑)	FID(↓)	sFID(↓)	CLIP(↑)	IR(↑)
Ours	36.52 ± 0.46	60.73 ± 0.26	300.59 ± 0.48	0.263 ± 0.003	0.870 ± 0.018

Results in Table 7 show that UTS itself has low calibration seed variance. The mean FID remains 6.78 points below the reported INCO-based VETA-DiT result of 67.51 under the same PixArt-Σ W3A4 evaluation protocol.

C Memory and Latency Optimization

All weights can be transformed offline and stored in a low-bit format. For activation, some layers can fuse into the weights from the previous layer and perform joint diagonalization to ensure accuracy without incurring additional cost. We implemented a custom CUDA kernel for W3A4 and W4A4 matrix multiplication to accelerate the computations. Therefore, layers that require online transform still offer advantages. As shown in Table 8, under the W3A4 bitwidth with online transform, we still achieved a memory savings of 4.062$\times$ and a 1.932$\times$ speedup.

Table 8: Comparison of latency, speedup, and memory reduction.

Bit-Width	Memory Opt.	Latency Opt.
FP	-	-
W4A4	3.997$\times$	2.124$\times$
W3A4	5.328$\times$	2.238$\times$
W4A4 + Online Transform	3.240$\times$	1.933$\times$
W3A4 + Online Transform	4.062$\times$	1.932$\times$

D Proofs

This section provides a detailed derivation of the relevant formulas:

D.1 Proof of Variance Equalization via Covariance Matrix Diagonalization Combined with Hadamard Transform

Since the weight matrix and the splicing activation matrix are not layer-normalized, we can neither use the second raw moment in place of variance nor the second cross moment in place of covariance.

Let matrix $X \in \mathbb{R}^{n\times m}$ and Hadamard matrix $H \in \mathbb{R}^{m\times m}$. $\mathbf{x}_k$ is the k-th column of X, and $\mathbf{z}_j$ is the j-th column of $Z = XH$. For each element z_{ij} in Z, we have:

$$z_{ij} = \sum_{k=1}^{m} x_{ik} h_{kj}. \tag{13}$$

The variance and covariance are defined as:

$$\mathrm{Var}(\mathbf{z}_j) = \frac{1}{n}\sum_{i=1}^{n}(z_{ij} - \bar{z}_j)^2, \tag{14}$$

$$\mathrm{Cov}(\mathbf{x}_k, \mathbf{x}_l) = \frac{1}{n}\sum_{i=1}^{n}(x_{ik} - \bar{x}_k)(x_{il} - \bar{x}_l), \tag{15}$$

where $\bar{z}_j = \frac{1}{n}\sum_i z_{ij}$ and $\bar{x}_k = \frac{1}{n}\sum_i x_{ik}$.

From the linear combination:

$$\bar{z}_j = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{m} x_{ik}h_{kj} = \sum_{k=1}^{m}\left(\frac{1}{n}\sum_{i=1}^{n} x_{ik}\right) h_{kj} = \sum_{k=1}^{m} \bar{x}_k h_{kj}. \tag{16}$$

Centered columns:

$$z_{ij} - \bar{z}_j = \sum_{k=1}^{m}(x_{ik} - \bar{x}_k)h_{kj}. \tag{17}$$

Thus

$$\begin{aligned}
\mathrm{Var}(\mathbf{z}_j) &= \frac{1}{n}\sum_{i=1}^{n}\Big(\sum_{k=1}^{m}(x_{ik} - \bar{x}_k)h_{kj}\Big)^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{m}\sum_{l=1}^{m} h_{kj}h_{lj}(x_{ik} - \bar{x}_k)(x_{il} - \bar{x}_l) \\
&= \sum_{k=1}^{m}\sum_{l=1}^{m} h_{kj}h_{lj}\underbrace{\frac{1}{n}\sum_{i=1}^{n}(x_{ik} - \bar{x}_k)(x_{il} - \bar{x}_l)}_{\mathrm{Cov}(\mathbf{x}_k, \mathbf{x}_l)} \\
&= \sum_{k=1}^{m}\mathrm{Var}(\mathbf{x}_k)h_{kj}^2 + \sum_{k\neq l}\mathrm{Cov}(\mathbf{x}_k, \mathbf{x}_l)h_{kj}h_{lj}.
\end{aligned} \tag{18}$$

Therefore, when a matrix has a diagonal covariance matrix, the second term in the above equation becomes zero. The Hadamard transform can still guarantee balanced variances across channels, even in the absence of normalization.

D.2 Proof of Orthogonal Transformations fails to Joint Diagonalization

If the transformation matrix is an orthogonal rotation, then a necessary and sufficient condition for both the activation covariance matrix and the weight covariance matrix to be diagonalized by this transformation is that these two matrices commute. For the distributions of activations or weights encountered in practical models, it is highly unlikely that the covariance matrices will exactly satisfy this commutativity property.

Let C_1, C_2 be real symmetric matrices of order n.

Necessity: Assume there exists an orthogonal matrix Q such that

$$Q^\top C_1 Q = \Lambda_1, \qquad Q^\top C_2 Q = \Lambda_2, \tag{19}$$

where Λ_1, Λ_2 are diagonal matrices. From the property $Q^\top = Q^{-1}$ of orthogonal matrices we obtain

$$C_1 = Q\Lambda_1 Q^\top, \qquad C_2 = Q\Lambda_2 Q^\top. \tag{20}$$

Computing the products:

$$C_1C_2 = Q\Lambda_1Q^\top \cdot Q\Lambda_2Q^\top = Q\Lambda_1\Lambda_2Q^\top, \quad C_2C_1 = Q\Lambda_2\Lambda_1Q^\top. \tag{21}$$

Since diagonal matrices commute ($\Lambda_1\Lambda_2 = \Lambda_2\Lambda_1$), it follows that $C_1C_2 = C_2C_1$.

Sufficiency: Assume $C_1C_2 = C_2C_1$. Because C_1 is real symmetric, there exists an orthogonal matrix Q_1 such that

$$Q_1^\top C_1 Q_1 = \Lambda_1, \tag{22}$$

where Λ_1 is diagonal. By reordering the columns of Q_1 we may group equal eigenvalues together; consequently Λ_1 can be written in block diagonal form

$$\Lambda_1 = \mathrm{diag}(\lambda_1 I_{k_1},\ \lambda_2 I_{k_2},\ \ldots,\ \lambda_r I_{k_r}), \tag{23}$$

where $\lambda_1, \ldots, \lambda_r$ are the distinct eigenvalues and k_i denotes the multiplicity of λ_i. Define $D = Q_1^\top C_2 Q_1$; then D is a real symmetric matrix. From $C_1C_2 = C_2C_1$ we obtain

$$\Lambda_1 D = D\Lambda_1. \tag{24}$$

Now partition D conformably with Λ_1, writing $D = (D_{ij})$ with blocks of size $k_i \times k_j$. The commutativity condition $\Lambda_1 D = D\Lambda_1$ compares the (i, j) blocks:

$$\lambda_i D_{ij} = D_{ij}\lambda_j.$$

For $i \neq j$ we have $\lambda_i \neq \lambda_j$; therefore $D_{ij} = 0$ whenever $i \neq j$. Hence D is block diagonal with the same structure as Λ_1:

$$D = \mathrm{diag}(D_{11},\ D_{22},\ \ldots,\ D_{rr}), \tag{25}$$

and each diagonal block D_{ii} is a real symmetric matrix of size k_i.

Since each D_{ii} is symmetric, it can be orthogonally diagonalized: there exist orthogonal matrices U_i of order k_i such that $U_i^\top D_{ii} U_i = \Lambda_{2,i}$ is diagonal. Assemble these into a block-diagonal orthogonal matrix

$$U = \mathrm{diag}(U_1,\ U_2,\ \ldots,\ U_r). \tag{26}$$

Then U is orthogonal, and

$$U^\top D U = \mathrm{diag}(\Lambda_{2,1},\ \ldots,\ \Lambda_{2,r}) = \Lambda_2, \tag{27}$$

which is a diagonal matrix. Moreover, because each diagonal block of Λ_1 is a scalar matrix $\lambda_i I_{k_i}$ that commutes with any matrix of the same size,

$$U_i^\top(\lambda_i I_{k_i})U_i = \lambda_i U_i^\top U_i = \lambda_i I_{k_i},$$

and therefore

$$U^\top \Lambda_1 U = \mathrm{diag}(\lambda_1 I_{k_1},\ \ldots,\ \lambda_r I_{k_r}) = \Lambda_1. \tag{28}$$

Finally, set $Q = Q_1U$. Then Q is orthogonal and we obtain

$$\begin{aligned} Q^\top C_1 Q &= U^\top(Q_1^\top C_1 Q_1)U = U^\top \Lambda_1 U = \Lambda_1, \\ Q^\top C_2 Q &= U^\top(Q_1^\top C_2 Q_1)U = U^\top D U = \Lambda_2, \end{aligned} \tag{29}$$

which shows that C_1 and C_2 are simultaneously orthogonally diagonalized by Q.

D.3 Proof of the Implementation of Joint Diagonalization by Solving GEVP

C_X-orthonormalized: Since a covariance matrix is a real symmetric matrix, and Tikhonov regularization [30] can guarantee C_X's positive definiteness, we are able to solve the GEVP:

$$C_W^{-1}\mathbf{v} = \lambda C_X \mathbf{v}. \tag{30}$$

For the i-th and j-th eigenvalues and eigenvectors and $i \neq j$, we have:

$$\begin{aligned} C_W^{-1}\mathbf{v_i} &= \lambda_i C_X \mathbf{v_i}, \\ C_W^{-1}\mathbf{v_j} &= \lambda_j C_X \mathbf{v_j}. \end{aligned} \tag{31}$$

Take the transpose of the first equation. By symmetry $C_W^{-\top} = C_W^{-1}, C_X^\top = C_X$:

$$\mathbf{v_i}^\top C_W^{-1} = \lambda_i \mathbf{v_i}^\top C_X. \tag{32}$$

Right-multiply by $\mathbf{v_j}$:

$$\mathbf{v_i}^\top C_W^{-1} \mathbf{v_j} = \lambda_i \mathbf{v_i}^\top C_X \mathbf{v_j}. \tag{33}$$

Left-multiply the second eigenvalue equation by $\mathbf{v_i}^\top$:

$$\mathbf{v_i}^\top C_W^{-1} \mathbf{v_j} = \lambda_j \mathbf{v_i}^\top C_X \mathbf{v_j}. \tag{34}$$

Subtract equation 34 from equation 33:

$$(\lambda_i - \lambda_j)\, \mathbf{v_i}^\top C_X \mathbf{v_j} = 0. \tag{35}$$

Since $\lambda_i \neq \lambda_j$, we have $\mathbf{v_i}^\top C_X \mathbf{v_j} = 0$ for $i \neq j$.

$\mathbf{C_X}$-normalization: For the ith eigenvector $\mathbf{v}_i$, since C_X is a positive-definite matrix, its quadratic form must be a real number greater than 0.

$$\mathbf{v}_i^\top C_X \mathbf{v}_i = k_i > 0. \tag{36}$$

Generalized eigenvectors's length can be scaled arbitrarily. If $\mathbf{v}_i$ is an eigenvector, then $c \cdot \mathbf{v}_i$ is also an eigenvector. For normalization, the underlying solver automatically divides each vector by $\sqrt{k_i}$ before returning the result. After this scaling, all eigenvectors satisfy:

$$\mathbf{v}_i^\top C_X \mathbf{v}_i = 1. \tag{37}$$

For $G = [\mathbf{v}_1, \ldots, \mathbf{v}_d]$. When we compute $G^\top C_X G$, the element in the ith row and jth column of the resulting matrix is $\mathbf{v}_i^\top C_X \mathbf{v}_j$. When $i \neq j$, due to **$\mathbf{C_X}$-orthonormalized**, the element is 0. When $i = j$, due to **$\mathbf{C_X}$-normalization**, the element is 1. Such that:

$$G^\top C_X G = I. \tag{38}$$

Substitute it into the equation 3, we have:

$$G^\top C_W^{-1} G = \Lambda. \tag{39}$$

For $\tilde{X} = XG$. Define: $\mu_X = \mathbb{E}[X]$, $C_X = \mathbb{E}\big[(X - \mu_X)^\top (X - \mu_X)\big]$. Similarly, $\mu_{\tilde{X}} = \mathbb{E}[\tilde{X}]$, $C_{\tilde{X}} = \mathbb{E}\big[(\tilde{X} - \mu_{\tilde{X}})^\top (\tilde{X} - \mu_{\tilde{X}})\big]$.

First, the mean of $\tilde{X}$ is

$$\mu_{\tilde{X}} = \mathbb{E}[XG] = \mathbb{E}[X]G = \mu_X G.$$

Center $\tilde{X}$:

$$\tilde{X} - \mu_{\tilde{X}} = XG - \mu_X G = (X - \mu_X)G.$$

Then the covariance matrix of $\tilde{X}$ is

$$\begin{aligned} C_{\tilde{X}} &= \mathbb{E}\big[(\tilde{X} - \mu_{\tilde{X}})^\top (\tilde{X} - \mu_{\tilde{X}})\big] \\ &= \mathbb{E}\big[((X - \mu_X)G)^\top ((X - \mu_X)G)\big] \\ &= \mathbb{E}\big[G^\top (X - \mu_X)^\top (X - \mu_X) G\big] \\ &= G^\top\, \mathbb{E}\big[(X - \mu_X)^\top (X - \mu_X)\big]\, G \\ &= G^\top\, C_X\, G \\ &= I. \end{aligned} \tag{40}$$

Similarly, for $\tilde{W} = WG^{-\top}$:

$$C_{\tilde{W}} = G^{-1} C_W G^{-\top} = \Lambda^{-1}. \tag{41}$$

D.4 Proof of Introducing Diagonal Scaling does not Destroy the Decorrelation Property

Referring to equation 40, we can obtain:

$$C_{\tilde{X}S} = S^\top C_{\tilde{X}} S, \tag{42}$$

for $W(GS)^{-\top}$:

$$C_{W(GS)^{-\top}} = (GS)^{-1} C_W (GS)^{-\top} = S^{-1} C_{\tilde{W}} S^{-\top}. \tag{43}$$

As can be seen, the transformed covariance matrix remains a diagonal matrix.

D.5 Proof of the General Diagonal Error Decomposition Formula

For any matrices A and B and a scalar a, we have $\mathrm{OffDiag}(A+B) = \mathrm{OffDiag}(A) + \mathrm{OffDiag}(B)$, $\mathrm{OffDiag}(aA) = a\,\mathrm{OffDiag}(A)$.

let $M_t = G^\top C_t G$,$\bar{M} = G^\top \bar{C} G$, By linearity, we have $\bar{M} = \frac{1}{T}\sum_{t=1}^{T} M_t$. The total loss is given by:

$$\mathcal{L}_{\text{total}} = \sum_{t=1}^{T} \|\mathrm{OffDiag}(M_t)\|_F^2 . \tag{44}$$

Substituting $M_t = \bar{M} + (M_t - \bar{M})$ and using the property that the square of the Frobenius norm expands as:

$$\|\mathrm{OffDiag}(M_t)\|_F^2 = \left\|\mathrm{OffDiag}(\bar{M}) + \mathrm{OffDiag}(M_t - \bar{M})\right\|_F^2 . \tag{45}$$

Using the norm identity for inner product spaces, $\|A+B\|_F^2 = \|A\|_F^2 + \|B\|_F^2 + 2\langle A, B\rangle_F$, and summing over all times, we obtain:

$$\begin{aligned}\sum_{t=1}^{T} \|\mathrm{OffDiag}(M_t)\|_F^2 = \sum_{t=1}^{T} \left\|\mathrm{OffDiag}(\bar{M})\right\|_F^2 + \sum_{t=1}^{T} \left\|\mathrm{OffDiag}(M_t - \bar{M})\right\|_F^2 \\ + 2\sum_{t=1}^{T} \left\langle \mathrm{OffDiag}(\bar{M}), \mathrm{OffDiag}(M_t - \bar{M})\right\rangle_F .\end{aligned} \tag{46}$$

Cross terms can be simplified:

$$\sum_{t=1}^{T} \mathrm{OffDiag}(M_t - \bar{M}) = \mathrm{OffDiag}\left(\sum_{t=1}^{T}(M_t - \bar{M})\right) = \mathrm{OffDiag}\left(\sum_{t=1}^{T} M_t - T\bar{M}\right) = 0. \tag{47}$$

Therefore, the sum of the cross terms is 0.

In the first term, $\left\|\mathrm{OffDiag}(\bar{M})\right\|_F^2$ is independent of t, so

$$\sum_{t=1}^{T} \left\|\mathrm{OffDiag}(\bar{M})\right\|_F^2 = T\left\|\mathrm{OffDiag}(\bar{M})\right\|_F^2 . \tag{48}$$

Substituting $M_t = G^\top C_t G$ and $\bar{M} = G^\top \bar{C} G$ back into the equation, we finally obtain:

$$\sum_{t=1}^{T} \left\|\mathrm{OffDiag}(G^\top C_t G)\right\|_F^2 = T\left\|\mathrm{OffDiag}\big(G^\top \bar{C} G\big)\right\|_F^2 + \sum_{t=1}^{T} \left\|\mathrm{OffDiag}\big(G^\top (C_t - \bar{C}) G\big)\right\|_F^2 . \tag{49}$$

D.6 Proof of the Covariance Expansion using Incoherence-Aware Adaptation

For the covariance matrix of X_{veta}, since the activation has been normalized, $Cov(X_{veta}) = X_{veta}^\top X_{veta}$, substitute the defining equation of X_{veta} into the covariance formula. To avoid confusion with the indices of the summation symbols during multiplication, we must use different index variables for the transposed term on the left-hand side and the original term on the right-hand side:

$$Cov(X_{veta}) = \left(\sum_{t=1}^{T} w_t X_t\right)^\top \left(\sum_{k=1}^{T} w_k X_k\right). \tag{50}$$

To perform the transpose operation, based on the linear property of matrix transposition and the transposition of scalar multiplication, we can move the transpose symbol inside the summation. Note that w_t is a scalar, so only the matrix X_t is transposed:

$$Cov(X_{veta}) = \left(\sum_{t=1}^{T} w_t X_t^\top\right) \left(\sum_{k=1}^{T} w_k X_k\right). \tag{51}$$

By the distributive law of matrix multiplication, the product of two finite sums can be expanded into a double sum by factoring out the scalars w_t and w_k and multiplying the matrices $X_t^\top$ and X_k in order:

$$Cov(X_{veta}) = \sum_{t=1}^{T}\sum_{k=1}^{T} w_t w_k X_t^\top X_k. \tag{52}$$

We divide all $T \times T$ combinations into two groups based on whether their time step indices are the same, and substitute these into the definition of the independent covariance matrix for each time step to obtain:

$$Cov(X_{veta}) = \sum_{t=1}^{T} w_t^2 Cov(X_t) + \sum_{t\neq k} w_t w_k X_t^\top X_k. \tag{53}$$

D.7 Proof of Isomorphic Covariance Expectation under Finite-Sample Mean Drift

We formally derive the expectation of the strictly centered covariance $C_{X_{uts}}$. Let $X_t \in \mathbb{R}^{N\times d}$ be the centered full-batch activation at timestep t ($\mu_t = 0$), and its full-batch covariance is $C_t = \frac{1}{N} X_t^\top X_t$. UTS extracts n tokens uniformly without replacement to form slice $A_t \in \mathbb{R}^{n\times d}$. We set the total calibration capacity be $N = nT$, such that the size of the calibration set matches the size of the single time step activation X_t. Let $\bar{a}_t \in \mathbb{R}^{1\times d}$ be the sample mean vector of the t-th extracted slice A_t, representing the localized mean drift due to finite sampling from the t-th timestep. Let $\bar{x}_{uts} \in \mathbb{R}^{1\times d}$ be the centroid of the concatenated tensor X_{uts}, which is the arithmetic average of all localized mean vectors: $\bar{x}_{uts} = \frac{1}{T}\sum_{t=1}^{T} \bar{a}_t$. The strictly centered covariance of the spliced tensor X_{uts} is defined as:

$$C_{X_{uts}} = \frac{1}{N} X_{uts}^\top X_{uts} - \bar{x}_{uts}^\top \bar{x}_{uts}. \tag{54}$$

Taking the mathematical expectation on both sides:

$$\mathbb{E}[C_{X_{uts}}] = \mathbb{E}\left[\frac{1}{N} X_{uts}^\top X_{uts}\right] - \mathbb{E}[\bar{x}_{uts}^\top \bar{x}_{uts}]. \tag{55}$$

Because the sub-slice second moment is an unbiased estimator of the full-batch second moment (proof in D.8) and the uncentered empirical second moment of the calibration set is the arithmetic mean of the sub-slice moments (proof in D.9), the first term simplifies to the temporally-averaged covariance $\bar{C}$:

$$\mathbb{E}\left[\frac{1}{N} X_{uts}^\top X_{uts}\right] = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[\frac{1}{n} A_t^\top A_t\right] = \frac{1}{T}\sum_{t=1}^{T} C_t = \bar{C}. \tag{56}$$

For the second term, $\bar{x}_{uts} = \frac{1}{T}\sum_{t=1}^{T} \bar{a}_t$, where $\bar{a}_t$ is the sample mean of slice A_t, and the $\bar{a}_t$ are independent of one another. Expanding the outer product yields cross-terms ($t \neq k$) and diagonal terms ($t = k$):

$$\mathbb{E}[\bar{x}_{uts}^\top \bar{x}_{uts}] = \frac{1}{T^2}\left(\sum_{t\neq k} \mathbb{E}[\bar{a}_t]^\top \mathbb{E}[\bar{a}_k] + \sum_{t=1}^{T} \mathbb{E}[\bar{a}_t^\top \bar{a}_t]\right). \tag{57}$$

Since the mean of each time step X_t is zero ($\mathbb{E}[\bar{a}_t] = \mu_t = 0$), all cross-terms vanish. The remaining diagonal terms $\mathbb{E}[\bar{a}_t^\top \bar{a}_t]$ represent the sampling variance of the mean. Applying the Finite Population Correction factor:

$$\mathbb{E}[\bar{a}_t^\top \bar{a}_t] = Cov(\bar{a}_t) = \frac{N-n}{n(N-1)} C_t. \tag{58}$$

Substituting this back into the expectation of the mean outer product:

$$\mathbb{E}[\bar{x}_{uts}^\top \bar{x}_{uts}] = \frac{1}{T^2}\sum_{t=1}^{T} \frac{N-n}{n(N-1)} C_t = \frac{N-n}{N(N-1)}\left(\frac{1}{T}\sum_{t=1}^{T} C_t\right) = \frac{N-n}{N(N-1)}\bar{C}. \tag{59}$$

Finally, substituting both terms back into the expectation of $C_{X_{uts}}$:

$$\mathbb{E}[C_{X_{uts}}] = \bar{C} - \frac{N-n}{N(N-1)}\bar{C} = \underbrace{\left(1 - \frac{N-n}{N(N-1)}\right)}_{\kappa}\bar{C}. \tag{60}$$

Since N is practically large, $\kappa \approx 1$. More importantly, because κ is a uniform scalar multiplier, the eigenvalue decomposition of $\mathbb{E}[C_{X_{uts}}]$ yields the identical orthogonal eigenvectors as $\bar{C}$.

D.8 Proof of the Sub-slice Second Moment is an Unbiased Estimator of the Full-batch Second Moment

Let $X_t \in \mathbb{R}^{N\times d}$ denote the full-batch activation tensor at timestep t, comprising N spatial tokens. We denote the i-th token as a row vector $x_{t,i} \in \mathbb{R}^{1\times d}$. Since X_t is normalized, the full-batch uncentered second moment C_t can be explicitly formulated as the sum of outer products of all physical tokens:

$$C_t = \frac{1}{N} X_t^\top X_t = \frac{1}{N} \sum_{i=1}^{N} x_{t,i}^\top x_{t,i}. \tag{61}$$

In the UTS paradigm, we construct the sub-slice $A_t \in \mathbb{R}^{n\times d}$ by uniformly sampling n tokens from X_t without replacement. To mathematically formalize this discrete sampling process, we introduce a binomial indicator variable $I_{t,i} \in \{0, 1\}$. Specifically, $I_{t,i} = 1$ if the i-th token is selected into the sub-slice A_t, and $I_{t,i} = 0$ otherwise. Consequently, the empirical second moment of the sampled sub-slice A_t can be equivalently expressed as a weighted sum over the entire population:

$$A_t^\top A_t = \sum_{i=1}^{N} I_{t,i} \cdot x_{t,i}^\top x_{t,i}. \tag{62}$$

We then apply the mathematical expectation operator $\mathbb{E}[\cdot]$ to both sides. In this context, the full-batch token features $x_{t,i}$ are deterministic constants, and the stochasticity strictly originates from the random sampling indicator $I_{t,i}$. According to the linearity of expectation:

$$\mathbb{E}[A_t^\top A_t] = \mathbb{E}\left[\sum_{i=1}^{N} I_{t,i} \cdot x_{t,i}^\top x_{t,i}\right] = \sum_{i=1}^{N} \mathbb{E}[I_{t,i}] \cdot x_{t,i}^\top x_{t,i}. \tag{63}$$

Under the assumption of uniform sampling without replacement, every token shares an identical marginal probability of being selected. Therefore, the expectation of the indicator variable is exactly determined by the sampling ratio:

$$\mathbb{E}[I_{t,i}] = P(I_{t,i} = 1) = \frac{n}{N}. \tag{64}$$

Substituting this probability back into the expectation equation, we can factor out the constant scalar:

$$\mathbb{E}[A_t^\top A_t] = \frac{n}{N} \sum_{i=1}^{N} x_{t,i}^\top x_{t,i} = \frac{n}{N}(X_t^\top X_t). \tag{65}$$

Dividing both sides by the slice length n, the scalar n cancels out perfectly, yielding the proof of unbiasedness:

$$\mathbb{E}\left[\frac{1}{n} A_t^\top A_t\right] = \frac{1}{N} X_t^\top X_t = C_t. \tag{66}$$

This algebraic cancellation demonstrates that the empirical second moment of a uniformly sampled sub-slice is a strictly unbiased estimator of its corresponding full-batch second moment C_t. Finally, by extending this unbiased property across the temporal dimension and applying the global averaging operator $\frac{1}{T}\sum_{t=1}^{T}$, we recover the theoretical globally-averaged covariance oracle $\bar{C}$:

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[\frac{1}{n} A_t^\top A_t\right] = \frac{1}{T} \sum_{t=1}^{T} C_t = \bar{C}. \tag{67}$$

D.9 Proof of the Uncentered Empirical Second Moment of the Calibration Set is the Sum of the Sub-slice Moments

Given that the assembled tensor X_{uts} is constructed by concatenating the feature slices A_t from T time steps along the row dimension, it can be represented as a column-block matrix:

$$X_{uts} = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_T \end{bmatrix}. \tag{68}$$

Transpose it; according to the rules for transposing block matrices, we obtain the row block matrix:

$$X_{uts}^{\top} = \begin{bmatrix} A_1^{\top} & A_2^{\top} & \cdots & A_T^{\top} \end{bmatrix}. \tag{69}$$

Multiplying the two and applying the rule for block matrix multiplication, we can directly prove that:

$$X_{uts}^{\top} X_{uts} = \begin{bmatrix} A_1^{\top} & A_2^{\top} & \cdots & A_T^{\top} \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_T \end{bmatrix} = \sum_{t=1}^{T} A_t^{\top} A_t. \tag{70}$$

E Detailed Algorithm for our method

The algorithmic details of our method are described in detail in algorithm 1. The algorithm consists of seven stages. **Phase 1**: For each time step, we perform uniform, non-repetitive random sampling from the corresponding full-size candidate feature tensor to extract slices of the true physical features from the original image. **Phase 2**: The unbiased slices extracted in the previous step are vertically concatenated along the sequence dimension to form the activation calibration set. **Phase 3**: Calculate the covariance matrices of the activation calibration set and the weights. **Phase 4**: Perform eigenvalue decomposition on the weight covariance matrix. After extracting the maximum eigenvalue, utilize a preset relative truncation threshold to perform hard truncation on all minuscule eigenvalues and null space eigenvalues. Reconstruct the inverse covariance matrix based on the truncated eigenvalues. **Phase 5**: Introduce a minimal numerical stability term and add it to the main diagonal of the activation covariance matrix to ensure it is positive definite and full rank. Subsequently, solve the GEVP of the matrix pencil. The resulting eigenvector matrix serves as the base transformation matrix. **Phase 6**: First, apply the base transformation to the activation calibration set and the weights to obtain the temporary matrices. Next, calculate the 99.5% quantile maximum absolute value for each channel. Based on the hyperparameter α, analytically calculate the asymmetric dynamic equalization coefficient for each channel, and thereby construct the diagonal stabilization matrix. **Phase 7**: Obtain the Hadamard matrix, merge the fundamental transformation matrix with the Hadamard matrix, and output the final equivalent transformation matrix.

F Limitations and Broader Impacts

In this work, we propose an effective PTQ framework. We obtain a calibration set capable of aligned temporal mean covariances via UTS. By solving the GEVP, we achieve joint diagonalization of weights and activations covariance matrix and balance the dynamic ranges through scaling after apply the transformation; finally, by combining this with the Hadamard matrix, we simultaneously achieve inter-channel variance balancing for both weights and activations.

It is important to note that, unlike orthogonal transformation, non-orthogonal invertible transformation cannot pass the Layernorm and fuse the transformation to the previous layer, as non-orthogonal transformation alter the variance of the original data. For weights where the number of tokens is smaller than the number of channels, since the transformation matrix is already larger than the weight matrix, using an equivalent transformation does not provide any speedup.

To address these issues, in future research we plan to employ a method that groups activation, extracts sub-covariance matrices within each group, and solves the GEVP independently for each sub-matrix. This approach transforms the transformation matrix into a block diagonal matrix, significantly alleviating the storage and computation burden associated with the transformation matrix and ensuring that all online transformations remain advantageous.

For broader impacts, there is a danger of misuse for surveillance, deepfakes, and other sensitive content generation if not properly regulated, like many optimization techniques that enable model deployment in low-resource environments. We advocate that deploying quantized model in real-world applications should be strictly coupled with ethical AI development principles.

G Additional Visualization Results

In this section, to further demonstrate the effectiveness of our method, we present additional visual results. Figures 5 and 6 show images randomly generated using the DiT-XL/2 model under the W4A4 and W3A4 quantization settings. Figures 7 and 8 also provide examples of images generated by the PixArt-Σ model.

Algorithm 1: Joint Diagonalization Quantization with Uniform Token Splicing

Input: Set of full-timestep candidate activations $X \in \mathbb{R}^{T\times N\times d}$, weight matrix $W \in \mathbb{R}^{d_{out}\times d}$
Input: Truncation threshold τ (e.g., 10^{-3}), Scaling factor $\alpha \in (0, 1)$
Input: Numerical stability constant ϵ (e.g., $10^{-5} \cdot \lambda_{max}(C_X)$)
Output: Optimal equivalent transformation matrix T_{JDUS}

Phase 1: Uniform Token Extraction
for $t = 1$ **to** T **do**
  Uniform random sampling without replacement to avoid temporal bias $\Omega_t \sim \mathcal{U}(\{1, 2, \ldots, N\}, \frac{N}{T})$
  Extract the exact physical feature slice via index selection $A_t \leftarrow X_t[\Omega_t, :] \in \mathbb{R}^{\frac{N}{T}\times C}$
  Append the unbiased sub-estimator to the list $\mathcal{A}$.append(A_t)

Phase 2: Spatial Splicing and Activation Calibration Construction
Concatenate all spatial slices vertically to form the compact calibration tensor:
$X_{uts} \leftarrow \text{Concat}(\mathcal{A}, \text{dim} = 0) \in \mathbb{R}^{N\times d}$

Phase 3: Covariance Computation
$C_X \leftarrow Cov(X_{uts}, dim = columns)$
$C_W \leftarrow Cov(W, dim = columns)$

Phase 4: Truncated Singular Value Decomposition
Compute eigenvalue decomposition: $C_W = U\text{diag}(\lambda_1, \ldots, \lambda_d)U^\top$
$\lambda_{max} \leftarrow \max_i(\lambda_i)$
for $i = 1$ **to** d **do**
  $\lambda_i^* \leftarrow \max(\lambda_i, \tau \cdot \lambda_{max})$
$\tilde{C}_W^{-1} \leftarrow U\text{diag}(\frac{1}{\lambda_1^*}, \ldots, \frac{1}{\lambda_d^*})U^\top$

Phase 5: Generalized Eigenvalue Problem Solving
$\epsilon_x \leftarrow \max\left(\epsilon,\ 10\epsilon \cdot \max\left(\text{diag}(C_X)\right)\right)$
$C_X \leftarrow C_X + \epsilon_x I$
Solve GEVP for matrix pencil $(\tilde{C}_W^{-1}, C_X)$ to find eigenvectors G:

Phase 6: Covariance-Preserving Scaling
$\tilde{X} \leftarrow X_{uts}G$
$\tilde{W} \leftarrow WG^{-\top}$
for $i = 1$ **to** d **do**
  $m_{X,i} \leftarrow \text{Percentile}(|\tilde{X}_{:,i}|, 99.5\%)$
  $m_{W,i} \leftarrow \text{Percentile}(|\tilde{W}_{:,i}|, 99.5\%)$
  $s_i \leftarrow \frac{m_{W,i}^{\alpha}}{m_{X,i}^{1-\alpha}}$
$S \leftarrow \text{diag}(s_1, \ldots, s_d)$

Phase 7: Final Transformation Matrix Construction
Construct the Hadamard matrix H;
Form the composite transform: $T_{\text{JDUS}} = G \cdot S \cdot H$;

return T_{JDUS}

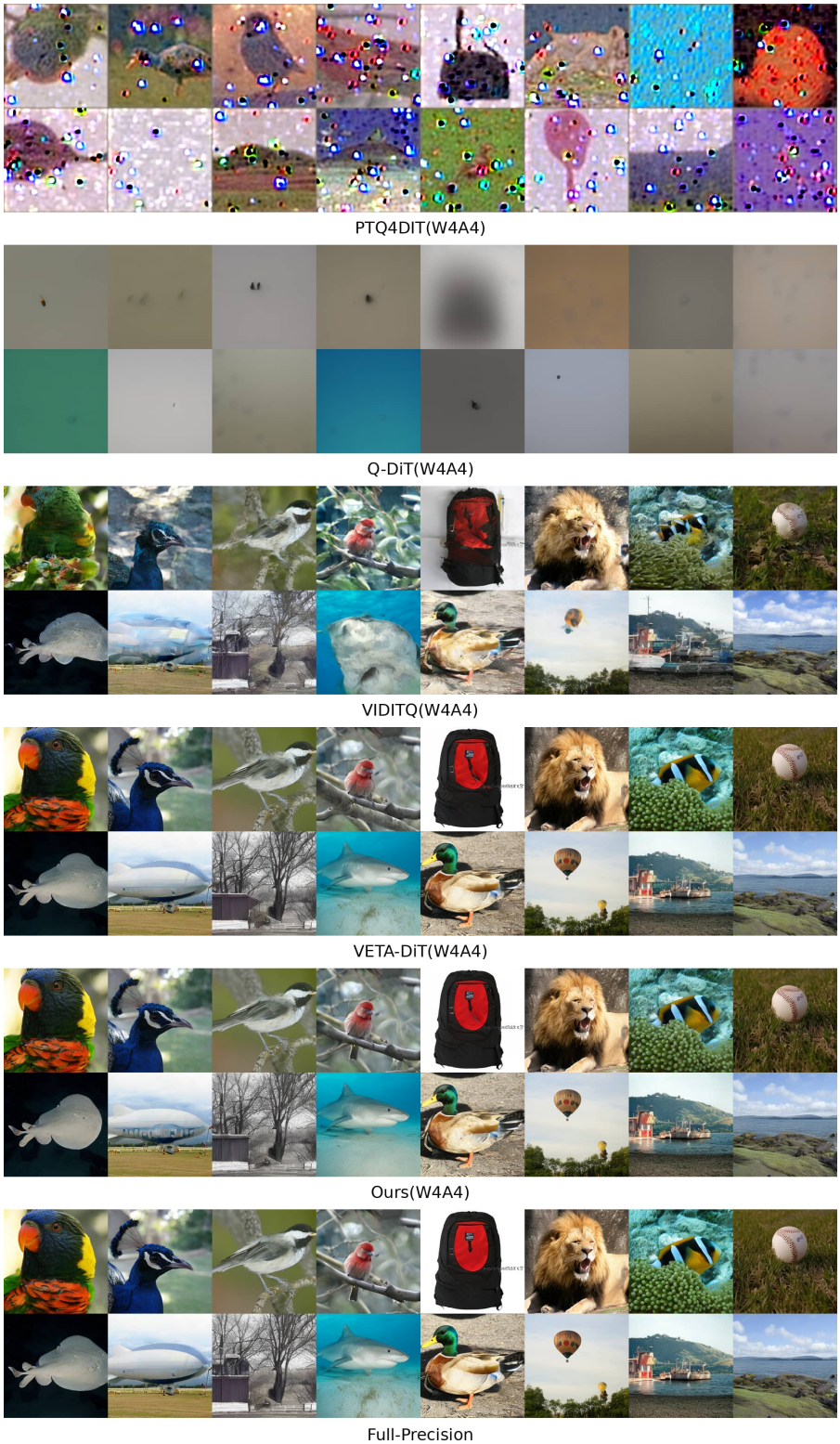


Figure 5: Randomly generated images from the DiT-XL/2 model on ImageNet 256×256, quantized to W4A4 using different PTQ methods.

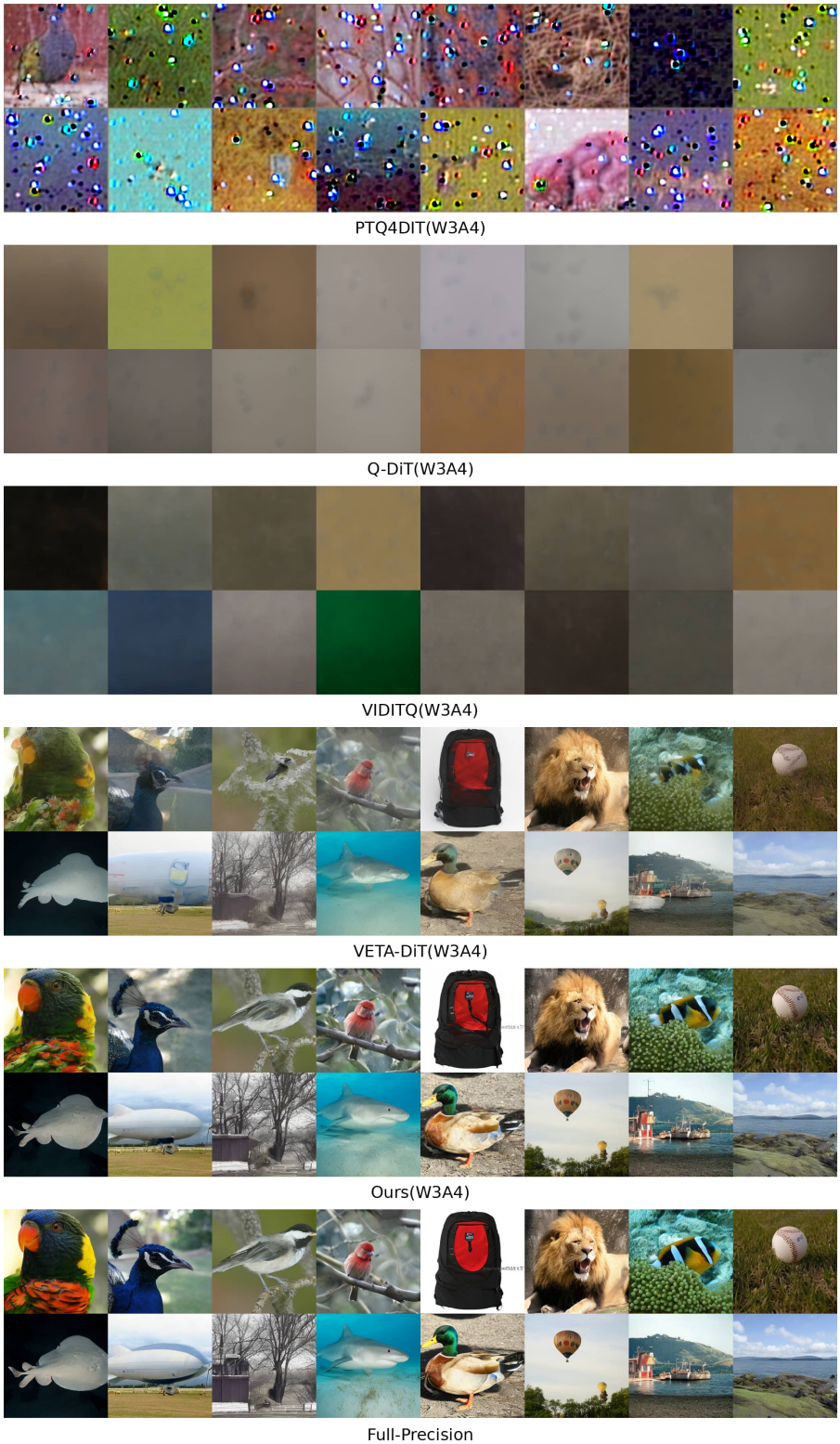


Figure 6: Randomly generated images from the DiT-XL/2 model on ImageNet 256×256, quantized to W3A4 using different PTQ methods.

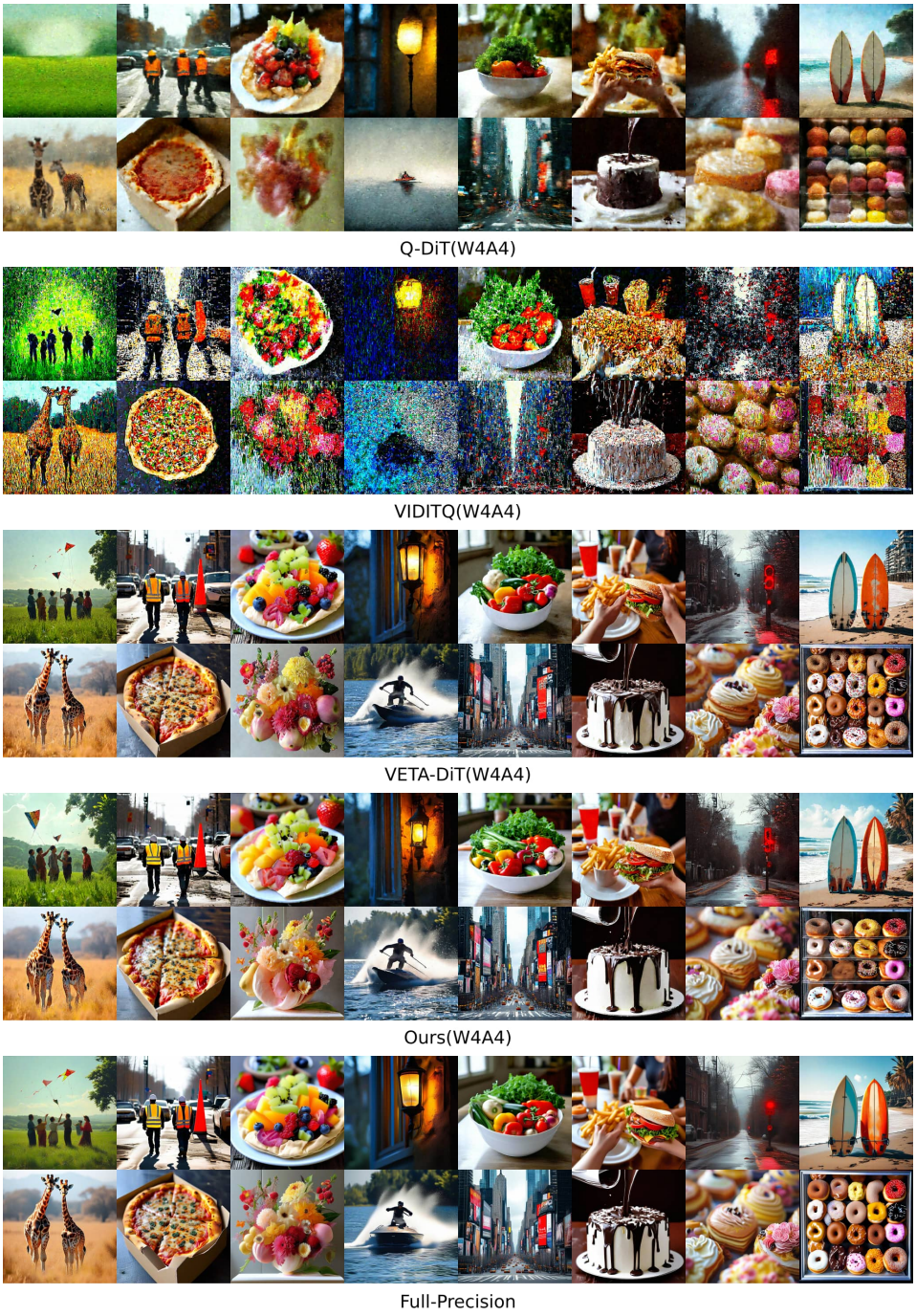


Figure 7: Random images produced by the Pixart-Σ model on COCO using different PTQ methods under W4A4 quantization.

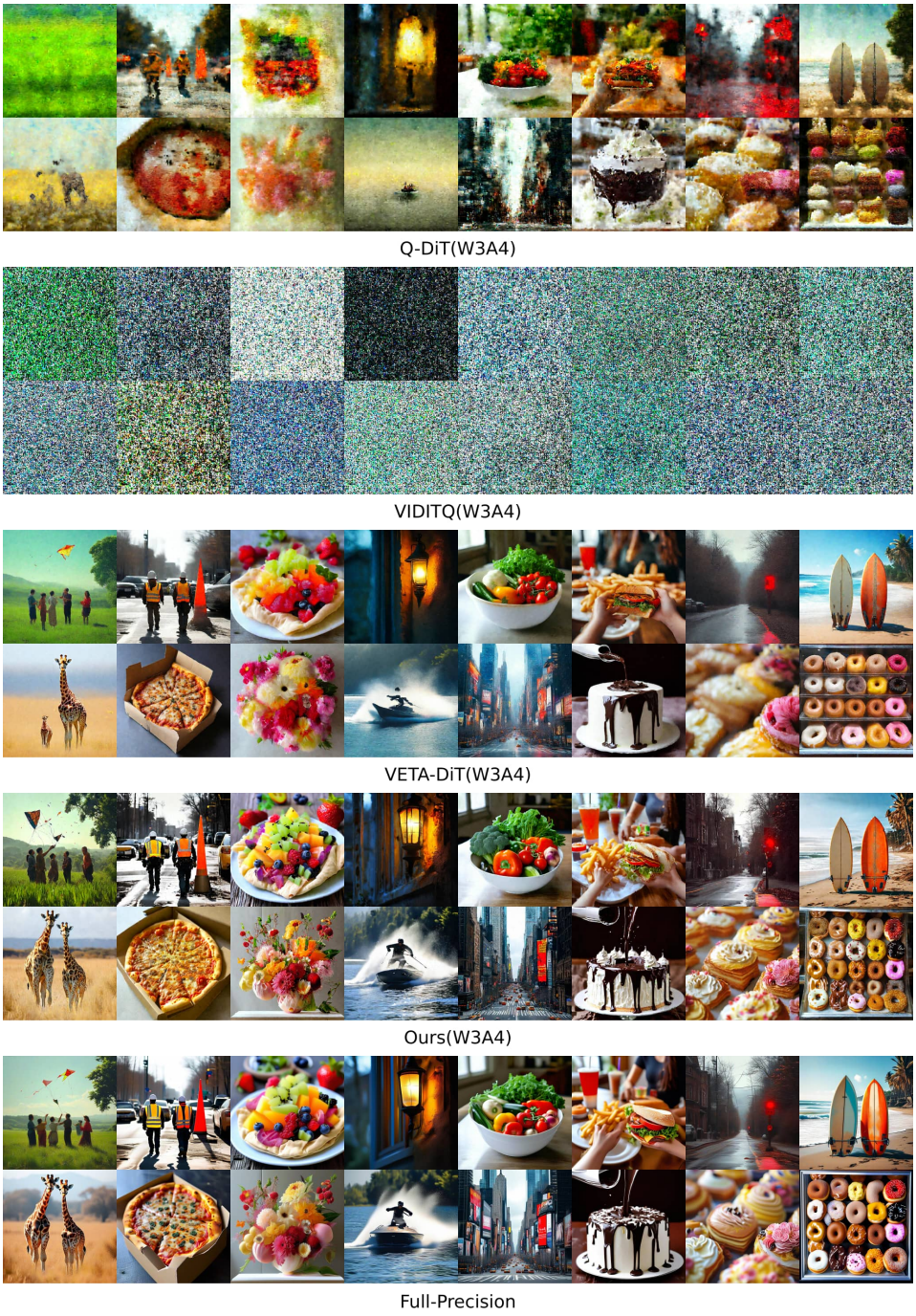


Figure 8: Random images produced by the Pixart-Σ model on COCO using different PTQ methods under W3A4 quantization.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We state the contributions in the abstract and introduction sections.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations of the work are available at Appendix F

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Please refer to Appendix D for related assumptions and proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Experiment details are available at Section 4.1 and Appendix A. We have released our code at: `https://github.com/KdxftXr7/JDUS-DiT`

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have released our code at: `https://github.com/KdxftXr7/JDUS-DiT`

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Please refer to Section 4.1 and Appendix A for experimental settings.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We perform multiple independent runs with different calibration seeds and report mean $\pm$ sample standard deviation at B.3.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments were conducted on Nvidia A800 GPUs, as mentioned in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impacts are provided at Appendix F

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all datasets and models utilized in our experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core methods of our research do not involve the use of LLMs.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.