DSAQuant: Denoising-Stage-Aligned Quantization-Aware Training for Video Generation

Shuaiting Li[1,2] **Zelin Gao**[1,4] **Haibin Shen**[2] **Yujun Shen**[1]
Haotong Qin[3†] **Yinghao Xu**[4,1†]
[1]Robbyant [2]ZJU [3]PolyU [4]HKUST
[†]Corresponding Authors

Abstract

Video diffusion models (VDMs) have achieved impressive progress in text-to-video generation, but their high memory and computational costs hinder practical deployment. Quantization-aware training (QAT) is an effective solution for compressing and accelerating advanced generative models without runtime overhead at inference. However, existing QAT methods suffer from a distinctive challenge in VDMs: while they often preserve prompt semantics, global layout, and coarse motion, the quantized model severely degrades visual details, texture fidelity, and sharpness. In this paper, we trace this degradation to the timestep-agnostic design of conventional quantization pipelines, which overlooks the stage-wise functionality of video denoising. In VDMs, early denoising steps mainly establish global structure and motion, whereas middle and late steps refine local appearance and high-frequency details. Based on this insight, we propose **DSAQuant**, a **D**enoising-**S**tage-**A**ligned **Quant**ization-aware training framework for VDMs. During training, *Denoising-Stage Oriented Supervision* preserves teacher distillation in early steps for stable structure planning, while shifting later steps toward target-driven optimization to enhance detail reconstruction. During inference, *Denoising-Stage Gated Guidance* disables CFG in the final denoising steps to prevent it from amplifying quantization-induced errors into high-frequency artifacts. Extensive experiments on the Wan and CogVideoX families under W4A4 and W3A3 settings show that DSAQuant consistently outperforms the SOTA QAT baseline, improving the VBench average score by up to 6.60 under aggressive W3A3 quantization while preserving strong text-video alignment. These results demonstrate that effective VDM quantization requires not only reducing quantization error, but also aligning quantization training and inference with the stage-wise nature of video diffusion.

Project page: `https://robbyant-research.github.io/DSAQuant/`
Code: `https://github.com/robbyant-research/DSAQuant`

1 Introduction

Video diffusion models (VDMs) [34, 31, 15] produce state-of-the-art text-to-video at compute and memory costs that block real deployment, making quantization a critical path to practical use. Existing post-training quantization (PTQ) methods [38, 16, 1] rely on online transformations whose per-layer runtime overhead on high-dimensional video tokens largely offsets the low-bit speedup, leaving advanced VDMs with limited practical benefit from low-bit PTQ.

Quantization-aware training (QAT) sidesteps this overhead by directly optimizing the model under low-bit constraints, with no auxiliary activation transformation at inference time. However, the

Preprint.

only existing QAT framework tailored for VDMs [11] employs a timestep-agnostic MSE distillation pipeline, which applies the same teacher-imitation objective and the same classifier-free guidance (CFG) schedule across every denoising step. Yet video denoising itself is not timestep-agnostic: early steps establish global layout and motion, whereas middle and late steps refine textures and frame-level sharpness. Figure 1 makes the consequence concrete: a conventional quantized VDM still follows the prompt and reproduces the global layout (Fig. 1a), yet its imaging-quality and sharpness scores collapse while prompt-alignment scores stay close to FP (Fig. 1a), and a stage-wise substitution experiment localizes the challenge to the middle and late timesteps (Fig. 1b). A timestep-agnostic quantization strategy is thus structurally mismatched to VDMs.

We trace this mismatch to two timestep-agnostic design choices. **On the training side**, a low-bit student forced to imitate the FP teacher in late steps may spend scarce discrete capacity matching the teacher's continuous-space fitting residual, without improving its own target prediction for detail reconstruction. **On the inference side**, the FP-level conditional and unconditional predictions converge in late steps, so the true guidance signal vanishes, while the two branches are quantized through independent activation grids whose errors are only weakly aligned and tend to accumulate rather than cancel; CFG then multiplicatively amplifies this uncorrelated quantization noise into the high-frequency artifacts we observe.

Building on this two-sided diagnosis, we propose **DSAQuant**, a denoising-stage-aligned QAT framework with two complementary components, one for each diagnosed failure. **Denoising-Stage Oriented Supervision** addresses the training-side capacity drain: we replace uniform MSE distillation with an exponential schedule that smoothly transfers supervision from a pure KD objective at $t = 0$ to a pure target-loss objective at $t = 1$. Early steps, therefore, inherit KD's variance reduction against the noisy STE gradient and stably anchor global structure, while late steps decouple the quantized student from the FP teacher's continuous-space fitting residual, freeing discrete capacity for genuine detail reconstruction. **Denoising-Stage Gated Guidance** addresses the inference-side noise amplification: we disable CFG in the final few denoising steps, removing the multiplicative amplifier that converts weakly aligned cond/uncond quantization errors into high-frequency artifacts, while preserving the strong text-video alignment that CFG provides at earlier steps. The two changes are orthogonal and complementary: the training-side change gives the quantized model the capacity to produce correct details, while the inference-side change ensures those details survive guidance. We validate DSAQuant across diverse VDMs (CogVideoX-2B/1.5-5B, Wan2.1-1.3B/14B, Wan2.2-5B) under both W4A4 and W3A3, achieving consistent and substantial improvements over the SOTA QAT baseline QVGen.

Our contributions are summarized as follows:

- We identify and empirically localize a *stage-dependent* failure mode in quantized VDMs: timestep-agnostic quantization preserves prompt semantics but severely degrades the detail-refinement capability of middle and late denoising steps.
- We propose **DSAQuant**, a denoising-stage-aligned QAT framework with two complementary components: *Denoising-Stage Oriented Supervision* with an exponential schedule that decouples the quantized student from the FP teacher's continuous-space residual and frees discrete capacity for late-step detail; and *Denoising-Stage Gated Guidance*, supported by a noise-amplification analysis showing that CFG multiplicatively amplifies the weakly aligned cond/uncond quantization errors in late timesteps.
- Extensive experiments on diverse VDMs (CogVideoX-2B/1.5-5B, Wan2.1-1.3B/14B, Wan2.2-5B) under W4A4 and W3A3 settings demonstrate consistent and substantial improvements over SOTA QAT baselines, validating the effectiveness of our framework.

2 Related Work

Efficient video diffusion models. Diffusion transformers (DiTs) [23] underpin recent advances in text-to-video generation, including CogVideoX, Wan, HunyuanVideo, and Open-Sora [34, 31, 15, 41]. Their large parameter counts, long spatio-temporal sequences, and multi-step sampling incur substantial memory and computational costs. Existing acceleration methods address different parts of this workload: step distillation shortens the denoising trajectory [35, 21], sparse attention reduces per-step computation [33], and feature caching reuses intermediate computations across

steps [22, 44, 42]. Quantization [14] complements these approaches by reducing the precision of weights and activations.

Post-training quantization for diffusion models. Early studies primarily investigate image diffusion models. Q-Diffusion [19], PTQD [8], and related approaches [24, 29] address quantization error along the denoising process, where activation distributions vary across timesteps. Temporal Dynamic Quantization [25], TFMQ-DM [10], and Temporal Feature Matters [12] explicitly exploit temporal information to improve calibration or preserve time-dependent features. For transformer backbones, PTQ4DiT [32] and Q-DiT [2] study activation outliers and dynamic activation quantization, while SVDQuant [16] handles difficult outlier components through a low-rank branch. SSVQ [17] and VQ4DiT [3] explore extreme DiT compression with vector quantization, but they rely on specific hardware to achieve speedup. These studies establish the importance of temporal variation and transformer-specific activation distributions for low-bit diffusion inference.

Quantization for video diffusion models. Video generation adds long token sequences, temporal consistency, and motion modeling to the quantization problem. QVD [27] addresses skewed temporal features and inter-channel range disparities through temporal discriminability and scattered channel range integration. ViDiT-Q [38] develops DiT-oriented PTQ and uses mixed precision to allocate capacity across sensitive layers and timesteps. S^2Q-VDiT [6] tackles long-token calibration through Hessian-aware salient data selection and attention-guided sparse token distillation. Nevertheless, low-bit activation quantization can still impair video quality, while online transformations can add runtime cost on long video sequences. Beyond PTQ, Q-DM [20] and EfficientDM [7] finetune quantized image diffusion models with FP-teacher supervision. Other low-bit diffusion frameworks explore more aggressive weight compression and timestep-related feature handling [26, 18]. QVGen [11] extends teacher-distillation-based QAT to VDMs and improves weight quantization with auxiliary modules.

3 Preliminary

3.1 Video Diffusion Sampling

Video diffusion models (VDMs) generate videos through an iterative denoising process. Let x_1 be the clean video latent and $\epsilon \sim \mathcal{N}(0, I)$ be Gaussian noise. We use $t \in [0, 1]$ to denote the sampling progress from noise to data, and write the noising path as

$$x_t = \alpha_t x_1 + \sigma_t \epsilon, \tag{1}$$

where $\alpha_0 = 0$, $\sigma_0 = 1$, $\alpha_1 = 1$, and $\sigma_1 = 0$. This notation includes DDIM-style sampling by choosing $\alpha_t = \sqrt{\bar{\alpha}_{\tau(t)}}$ and $\sigma_t = \sqrt{1 - \bar{\alpha}_{\tau(t)}}$, and flow matching by choosing, e.g., $\alpha_t = t$ and $\sigma_t = 1 - t$.

Under this formulation, the original training target can be written generically as

$$y_t = a_t x_1 + b_t \epsilon, \tag{2}$$

At each step, the denoising network tries to predict a diffusion target $f(x_t, c, t)$, where the target can be noise, velocity, or a data-related quantity.

3.2 Stage-Wise Quantization Analysis for Video Denoising

The denoising process of VDMs is stage-dependent: with signal-to-noise ratio $\mathrm{SNR}_t = \alpha_t^2/\sigma_t^2$, early steps have low SNR and rely on the text condition and generative prior to establish global semantics, scene layout, and coarse motion, while late steps operate on latents that already carry recognizable structure and focus on refining local appearance, high-frequency textures, and frame-level sharpness.

Despite this stage-wise functionality, conventional quantization-aware training (QAT) treats every denoising step uniformly [7, 20, 11]: it optimizes a quantized model f^q to match the full-precision model f^{fp} across all steps indiscriminately. However, as shown in Figure 1(a), directly applying such uniform QAT to VDMs leads to a distinctive degradation pattern: the quantized model preserves prompt semantics, global layout, and coarse motion, but suffers from blurred textures, corrupted details, reduced sharpness, and unstable local motion.

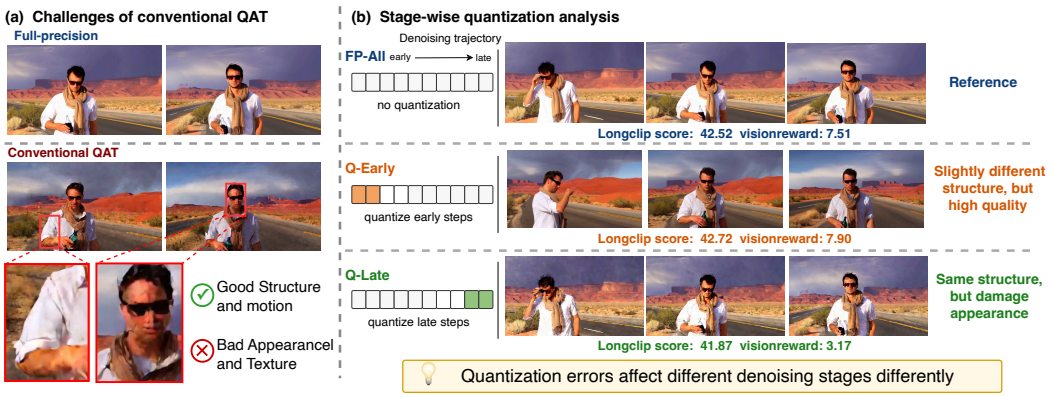

Figure 1: **Analysis of conventional QAT on VDMs (please zoom in for details)**. (a) Conventional methods generally preserve layout and motion, but suffer severe degradation in details and texture. (b) We apply quantization to different denoising stages of VDMs: quantizing early steps slightly perturbs layout but preserves perceptual quality, while quantizing late steps significantly damages visual quality.

To identify the source of this degradation, we perform a stage-wise quantization analysis (Figure 1(b)), in which different sampling stages are quantized separately while the remaining stages are kept full-precision. Quantizing only the early steps slightly perturbs global layout but preserves perceptual quality, while quantizing the middle and late steps reproduces the full failure mode. Consistent with the stage-wise picture above, conventional QAT fails not because it destroys semantic generation, but because its timestep-agnostic objective is inadequate for the detail-sensitive late stages. These observations motivate a denoising-stage-aligned framework that adapts both training supervision and inference guidance to the role each sampling stage plays.

4 Method

The failure mode identified in Sec. 3.2 traces to two timestep-agnostic design choices: a uniform distillation objective during training, and a fixed CFG scale during inference. To address them, we present **DSAQuant**, a **D**enoising-**S**tage-**A**ligned QAT framework with two stage-aware components: **Denoising-Stage Oriented Supervision** (Sec. 4.1) and **Denoising-Stage Gated Guidance** (Sec. 4.2).

4.1 Training Stage: Denoising-Stage Oriented Supervision

The standard target-driven loss used in pretraining of diffusion models is:

$$\mathcal{L}_{\text{Target}}(t) = \|f^q(x_t, c, t) - y_t\|_2^2 . \tag{3}$$

Instead, traditional QAT paradigms [7, 11, 20] usually train the quantized model by imitating the full-precision model, using distillation loss:

$$\mathcal{L}_{\text{KD}}(t) = \|f^q(x_t, c, t) - f(x_t, c, t)\|_2^2 . \tag{4}$$

In early high-noise steps, the noisy latent x_t contains limited information about the clean target x_1. In this regime, teacher imitation is useful in two complementary ways. First, constraining the quantized model to follow the full-precision teacher preserves functional similarity in semantic control, spatial layout, and coarse motion, thereby anchoring the global denoising trajectory. Second, the distillation target provides a lower-variance supervision signal than the sample-wise native target: it replaces the uncertain target y_t with the teacher's estimate of the conditional denoising direction, which stabilizes STE-based QAT updates. We analyze this variance-reduction effect in Appendix A.

However, uniform distillation becomes less suitable in late denoising steps. At this stage, the global structure has largely been formed, and the model mainly refines local appearance, high-frequency textures, object boundaries, and frame-level sharpness. Under low-bit quantization, forcing it to imitate the teacher at every step can therefore waste its limited capacity on matching the teacher

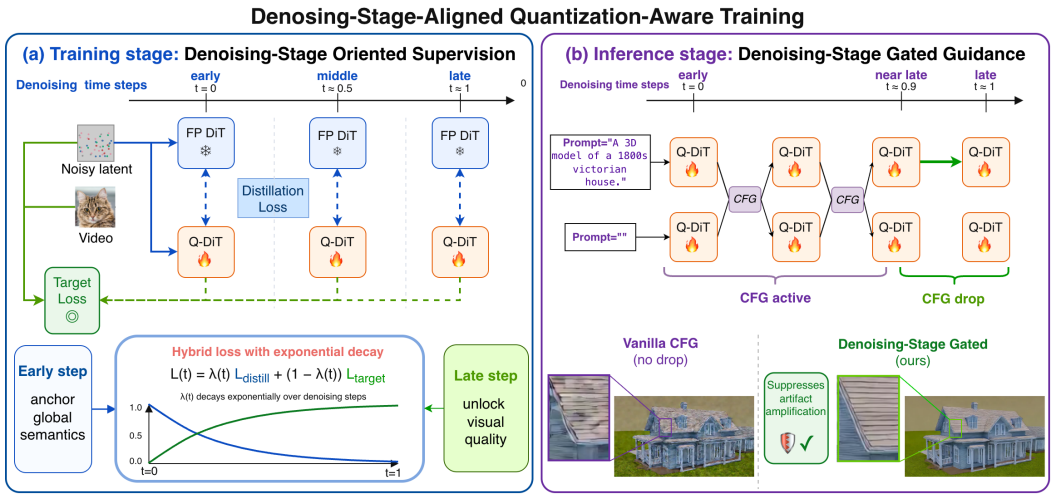

Figure 2: Overview of **DSAQuant**. (a) During training, **Denoising-Stage Oriented Supervision** smoothly transfers supervision from KD in early steps to the target loss in late steps via an exponential schedule, freeing late-step capacity for detail reconstruction. (b) During inference, **Denoising-Stage Gated Guidance** disables CFG in the final few steps, removing the multiplicative amplifier of cond/uncond quantization mismatch.

residuals rather than minimizing the true diffusion or flow target error. Since late-stage errors directly manifest as blur, texture corruption, and loss of sharpness, the supervision should be oriented toward detail refinement.

We therefore propose *Denoising-Stage Oriented Supervision*, a smooth denoising-stage hybrid objective that preserves teacher distillation in early steps and gradually shifts later steps toward the original diffusion or flow target. Formally, we define the overall training objective as a time-dependent weighted sum of the distillation loss and the native target loss:

$$\mathcal{L}_{\text{mix}}(t) = \lambda_{\text{KD}}(t)\mathcal{L}_{\text{KD}}(t) + \lambda_{\text{Target}}(t)\mathcal{L}_{\text{Target}}(t) \tag{5}$$

We design a normalized exponential scheduler to smoothly govern the transition. The weighting functions are formulated as:

$$\lambda_{\text{KD}}(t) = \frac{\exp(\alpha(1-t)) - 1}{\exp(\alpha) - 1}, \quad \lambda_{\text{Target}}(t) = 1 - \lambda_{\text{KD}}(t) \tag{6}$$

Here, $\alpha > 0$ controls the strength of the exponential decay. By construction, $\lambda_{\text{KD}}(0) = 1$ and $\lambda_{\text{Target}}(1) = 1$: early steps are fully driven by KD, anchoring layout and stabilizing the STE gradient, while late steps are driven by the target loss, freeing discrete capacity for high-frequency detail.

4.2 Inference Stage: Denoising-Stage Gated Guidance

Classifier-free guidance (CFG) is commonly used during inference to improve text-video alignment. We write CFG in the equivalent form

$$\hat{f}_\omega(x_t, c, t) = f(x_t, \emptyset, t) + \omega\left(f(x_t, c, t) - f(x_t, \emptyset, t)\right), \tag{7}$$

where ω is the guidance strength. Let $y_t = y_t(x_1, \epsilon, t)$ denote the sample-wise original diffusion or flow target. The target is shared by the conditional and unconditional branches. We denote the optimal conditional and unconditional predictors as

$$m_c(t) = \mathbb{E}[y_t \mid x_t, c], \qquad m_\emptyset(t) = \mathbb{E}[y_t \mid x_t]. \tag{8}$$

For the quantized model, we decompose the two predictions as

$$f_c^q(t) = m_c(t) + \eta_c^q(t), \qquad f_\emptyset^q(t) = m_\emptyset(t) + \eta_\emptyset^q(t), \tag{9}$$

where $\eta_c^q(t)$ and $\eta_\emptyset^q(t)$ denote quantization-induced deviations from the ideal predictors.

Substituting Eq. (9) into Eq. (7), the guided prediction error with respect to the true target becomes

$$\hat{f}^q_\omega(t) - y_t = \underbrace{m_\emptyset(t) - y_t}_{\text{target residual}} + \underbrace{\eta^q_\emptyset(t)}_{\text{uncond quant error}} + \omega\Big(\underbrace{m_c(t) - m_\emptyset(t)}_{\text{ideal guidance direction}} + \underbrace{\eta^q_c(t) - \eta^q_\emptyset(t)}_{\text{branch quantization mismatch}} \Big). \quad (10)$$

This decomposition shows that CFG amplifies not only the ideal guidance direction, but also the mismatch between the conditional and unconditional quantization errors. In quantized VDMs, the two branches can induce different activation distributions, leading to different clipping and rounding with dynamic activation quantization. Therefore, the quantization errors from the two branches are only weakly correlated; their difference tends to accumulate rather than cancel, and CFG amplifies this mismatch (analysis in Appendix B).

Early denoising benefits from CFG for prompt alignment and global structure formation. As the denoising process progresses, x_t encapsulates increasingly rich video details, leading to a relative decrease in shared semantic information; consequently, the ideal guidance gradually diminishes. However, as x_t deviates further from the initial Gaussian distribution, the quantization error continuously amplifies. As a result, the branch mismatch paradoxically emerges as the dominant factor in the trailing timesteps, which could lead to high-frequency artifacts. As evidenced in Fig. 3, the mismatch error rises rapidly in those late timesteps.

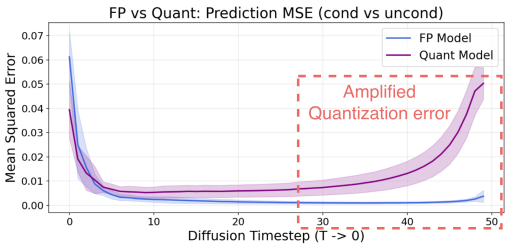

Figure 3: **Cond/uncond prediction MSE across denoising timesteps for FP vs. quantized models.** While the FP model's MSE stays near zero throughout, the quantized model's MSE rises sharply in late timesteps (boxed region), showing that branch quantization mismatch dominates the (vanishing) ideal guidance signal.

We therefore propose a gated guidance scheduler:

$$\hat{f}^q_\omega(t) = \begin{cases} f^q_\emptyset(t) + \omega\big(f^q_c(t) - f^q_\emptyset(t)\big), & t \le \tau, \\ f^q_c(t), & t > \tau. \end{cases} \quad (11)$$

which removes the amplification of the branch quantization mismatch in the final few steps. (Computationally, the unconditional branch can also be skipped during these late steps, since it is no longer used.)

5 Experiments

5.1 Implementation details

Models and Baselines. Following QVGen [11], we conduct experiments on advanced open-source DMs, including CogVideoX-2B, CogVideoX1.5-5B [34], Wan2.1 1.3B and Wan2.1 14B. We further include Wan2.2 5B [31], a model with highly compressed VAE for fast generation. We adopt previous powerful PTQ and QAT methods as baselines: ViDiT-Q [38], SVDQuant [16], S2Q-VDIT [6], LSQ [5], Q-DM [20], EfficientDM [7], and QVGen [11]. We primarily compare our method against QVGen because it's the first and only QAT framework tailored for VDMs, to our knowledge.

Quantization settings. Following QVGen [11], we use static per-channel weight quantization and dynamic per-token activation quantization. We use symmetric W4A4 quantization for kernel compatibility [16, 1] and additionally report asymmetric W3A3 results for aggressive low-bit settings. We evaluate released QVGen checkpoints when available and otherwise reproduce it with the same data, iterations, and quantizers as ours. For Wan, we additionally use a matched KD-only Base with shared flow-matching training settings (Appendix I.1). Following [18, 26], we exclude timestep-conditioned linear layers from quantization because their finite-set inference outputs can be precomputed as LUTs (Appendix F).

Training. Following [9], we synthesize training data using publicly released VidProM-derived text prompts [30]. Detailed settings can be found in Appendix D.

Evaluation. Our evaluation protocol incorporates multiple benchmarks. Following prior works [38, 6, 11], we evaluate 8 standard dimensions in VBench [13]. We additionally utilize VBench-2.0 [39] to gauge adherence to *physical laws* and *reasoning*. Furthermore, following [43], we measure Vision

Table 1: **Performance comparison across different quantization methods on VBench [13].** "*" indicates results taken from the QVGen [11] paper. "†" denotes QVGen's original asymmetric results from the public checkpoint. "Full Prec." denotes the full-precision (BF16) model. Best results are highlighted in **bold**.

Method	#Bits (W/A)	Sym	Imaging Quality ↑	Aesthetic Quality ↑	Motion Smoothness ↑	Dynamic Degree ↑	Background Consistency ↑	Subject Consistency ↑	Scene Consistency ↑	Overall Consistency ↑	Average ↑
`Wan2.1` 1.3B (`CFG` $= 5.0, 480p,$ `fps` $= 16$)											
Full Prec.	16/16	–	64.22	57.85	97.25	71.11	95.72	93.34	26.07	24.70	66.28
ViDiT-Q*	4/6	×	56.24	50.18	94.81	52.43	89.67	82.53	13.45	19.58	57.36
SVDQuant*	4/6	✓	58.16	51.27	97.05	49.44	93.74	91.71	14.18	23.26	59.85
LSQ*	4/4	×	59.11	49.09	**98.35**	71.11	92.66	91.67	10.38	18.83	61.40
Q-DM*	4/4	×	60.40	52.50	97.22	76.67	93.37	89.26	13.28	21.63	63.04
EfficientDM*	4/4	×	60.70	53.57	96.18	56.39	93.74	91.70	11.77	21.19	60.66
QVGen†	4/4	×	59.36	54.23	96.92	73.89	93.54	90.57	11.66	23.14	62.90
QVGen	4/4	✓	59.88	54.92	97.18	69.72	94.58	90.95	18.05	23.17	63.56
Ours	4/4	✓	**63.52**	**58.01**	96.66	**80.00**	**95.52**	**92.96**	**26.54**	**24.49**	**67.21**
CogVideoX-2B (`CFG` $= 6.0, 480p,$ `fps` $= 8$)											
Full Prec.	16/16	–	58.21	54.37	97.42	65.27	94.62	92.54	37.07	25.10	65.57
ViDiT-Q*	4/6	×	54.72	43.01	92.18	43.22	90.76	81.02	26.25	20.41	56.45
S2Q-VDiT*	4/6	✓	53.71	52.31	98.09	36.11	96.15	93.99	34.23	24.90	61.18
LSQ*	4/4	×	58.73	54.20	97.57	45.00	92.97	92.41	24.06	23.17	61.01
Q-DM*	4/4	×	54.96	52.71	98.00	48.61	93.82	91.86	28.02	23.87	61.48
EfficientDM*	4/4	×	55.96	51.97	98.03	46.67	94.10	91.70	27.76	24.28	61.31
QVGen†	4/4	×	56.41	53.21	97.88	60.83	94.06	91.66	27.82	24.54	63.30
QVGen	4/4	✓	56.14	52.04	**97.50**	**57.77**	93.92	91.23	29.33	24.47	62.80
Ours	4/4	✓	**59.05**	**55.01**	97.49	57.22	**94.32**	**92.47**	**34.26**	**25.54**	**64.42**
`Wan2.2` 5B (`CFG` $= 5.0, 720p,$ `fps` $= 16$)											
Full Prec.	16/16	–	64.31	59.03	98.23	66.11	96.56	94.82	35.18	25.54	67.47
QVGen	4/4	✓	62.85	57.95	97.92	62.50	95.63	93.21	25.52	24.79	65.04
Ours	4/4	✓	**63.46**	**59.86**	**98.66**	**64.17**	**96.10**	**95.02**	**29.08**	**25.33**	**66.46**
CogVideoX1.5-5B (`CFG` $= 6.0, 720p,$ `fps` $= 16$)											
Full Prec.	16/16	–	65.36	59.42	98.39	60.83	96.57	95.00	38.43	26.29	67.54
QVGen	4/4	✓	65.70	56.58	98.81	47.50	96.58	94.05	30.42	25.23	64.35
Ours	4/4	✓	**66.63**	**59.17**	**98.82**	**54.17**	**97.04**	**94.94**	**38.17**	**26.04**	**66.87**

Reward and instruction following using high-motion prompts. Finally, the visual clarity of the videos is quantified via Laplacian Variance (LV).

5.2 Main results

Comparison with baselines. Tab. 1 presents the evaluation results on the standard VBench, alongside a comparison with existing PTQ and QAT frameworks. Under the 4-bit symmetric quantization setting, our method consistently outperforms QVGen and even surpasses its 4-bit asymmetric counterpart. Notably, on the Wan-1.3B model, our approach achieves a +8.49 improvement in scene consistency and a 3.64 gain in image quality. The consistent performance gains observed on the DDIM-based CogVideo further validate the generalizability of our method. Additional W3A3 VBench results are provided in Appendix G.

Results on complex prompts and lower bitwidths. We further evaluate quantized models under augmented prompts, which contain more complex scenes and instructions. Results are summarized in Tab. 2. Across all tested `Wan` backbones, our W4A4 models consistently outperform prior quantization baselines and remain close to the full-precision models. For example, our method improves the average score over QVGen by 1.40–2.14 points at 4-bit, while reducing the gap to full precision to less than 0.6 points on all models. Moreover, our method remains robust under more aggressive 3-bit quantization, achieving 2.63–6.60 higher average scores than QVGen. Furthermore, we assess our models on VBench-2.0. As shown in Tab. 3, on Wan2.1-14B, our method maintains high performance even at 3-bit settings. These results demonstrate that our method preserves strong generation capability even for challenging prompts and lower bit-width settings. Representative qualitative comparisons are shown in Appendix N, with full video comparisons provided in the supplementary material.

5.3 Ablation study

To demonstrate the effect of each design, we employ W4A4 quantization on Wan 1.3B for ablation studies. Besides VBench, we also evaluate VisionReward and LV for a comprehensive comparison.

Ablation on two key designs. First, we evaluate the two proposed components in a 2-by-2 ablation, as summarized in Tab. 4. Denoising-Stage Oriented Supervision improves perceptual quality by

Table 2: **Performance comparison on VBench [13] with augmented prompts under W4A4 and W3A3 settings.** "†" denotes QVGen's original asymmetric results from the public checkpoint. "Full Prec." denotes the full-precision (BF16) model. Best results are highlighted in **bold**.

Method	#Bits (W/A)	Sym	Imaging Quality↑	Aesthetic Quality↑	Motion Smoothness↑	Dynamic Degree↑	Background Consistency↑	Subject Consistency↑	Scene Consistency↑	Overall Consistency↑	Average↑
`CogVideoX` 2B (`CFG` $= 5.0, 480p,$ `fps` $= 16$)											
Full Prec.	16/16	-	60.18	60.71	97.21	72.22	93.66	91.40	51.10	27.22	69.21
QVGen	4/4	✓	60.53	59.87	97.15	**64.80**	93.43	90.41	50.07	27.13	67.94
Ours	4/4	✓	**61.86**	**62.37**	**97.36**	61.28	**94.12**	**90.73**	**53.34**	**27.34**	**68.55**
QVGen	3/3	×	56.61	56.75	97.17	**47.22**	93.34	88.65	45.45	26.21	63.93
Ours	3/3	×	**59.65**	**59.98**	**97.34**	44.72	**93.69**	**90.39**	**49.21**	**26.87**	**65.23**
`Wan2.1` 1.3B (`CFG` $= 5.0, 480p,$ `fps` $= 16$)											
Full Prec.	16/16	-	64.73	65.12	97.99	70.28	96.69	93.62	46.05	25.77	70.00
QVGen†	4/4	×	62.93	63.25	**98.30**	62.77	95.75	93.13	40.45	25.04	67.71
QVGen	4/4	✓	58.42	59.50	97.21	63.06	94.56	89.43	32.50	24.14	64.85
Ours	4/4	✓	**63.99**	**64.36**	97.95	**73.33**	**96.30**	**93.17**	**44.10**	**25.62**	**69.85**
QVGen	3/3	×	58.00	57.30	**98.32**	35.55	94.63	90.24	32.00	22.57	61.08
Ours	3/3	×	**62.50**	**62.27**	97.96	**68.05**	**95.54**	**92.09**	**37.73**	**25.30**	**67.68**
`Wan2.2` 5B (`CFG` $= 5.0, 720p,$ `fps` $= 16$)											
Full Prec.	16/16	-	66.95	65.96	98.65	57.22	97.68	94.25	48.25	26.17	69.39
QVGen	4/4	✓	64.84	63.68	98.49	52.22	96.75	93.78	44.68	25.43	67.48
Ours	4/4	✓	**66.06**	**66.36**	**98.86**	**55.27**	**97.70**	**94.75**	**46.21**	**25.86**	**68.88**
QVGen	3/3	×	63.44	62.36	98.40	46.11	96.57	93.04	41.48	24.34	65.72
Ours	3/3	×	**64.18**	**65.46**	**98.74**	**55.27**	**97.52**	**94.15**	**45.67**	**25.82**	**68.35**
`Wan2.1` 14B (`CFG` $= 5.0, 720p,$ `fps` $= 16$)											
Full Prec.	16/16	-	67.41	65.85	97.88	73.05	97.08	92.87	45.93	26.03	70.76
QVGen	4/4	✓	66.51	65.35	97.84	62.16	96.14	92.59	41.96	25.15	68.34
Ours	4/4	✓	**67.60**	**65.88**	**98.38**	**68.89**	**97.33**	**93.75**	**43.69**	**26.04**	**70.20**
QVGen	3/3	×	64.31	64.45	97.32	58.33	95.64	92.74	38.11	25.17	67.01
Ours	3/3	×	**66.35**	**65.57**	**98.74**	**64.44**	**97.22**	**94.52**	**45.96**	**25.87**	**69.83**

Table 3: **Performance comparison on VBench-2.0 [39].** "*" indicates results taken from the QVGen [11] paper. "†" denotes QVGen's original asymmetric results from the public checkpoint. "Full Prec." denotes the full-precision (BF16) model. Best results are highlighted in **bold**.

Method	#Bits (W/A)	Sym	Creativity↑	Commonsense↑	Control↑	Human Fidelity↑	Physics↑	Average↑
`Wan2.1` 1.3B (`CFG` $= 5.0, 480p,$ `fps` $= 16$)								
Full Prec.	16/16	-	50.0	63.9	35.2	78.7	59.7	57.5
QVGen†	4/4	×	43.3	61.6	31.0	**81.1**	57.2	54.8
Ours	4/4	×	**48.5**	**63.9**	**35.7**	76.2	**59.0**	**56.7**
QVGen	4/4	✓	**46.5**	57.5	28.0	70.1	57.2	51.9
Ours	4/4	✓	45.0	**62.4**	**35.0**	**74.0**	**57.3**	**54.8**
QVGen	3/3	×	37.3	51.9	23.6	**72.3**	52.6	47.5
Ours	3/3	×	**43.9**	**61.5**	**31.0**	67.1	**58.9**	**52.5**
`Wan2.1` 14B (`CFG` $= 5.0, 720p,$ `fps` $= 16$)								
Full Prec.	16/16	-	55.2	64.0	37.3	81.6	62.8	60.2
QVGen*	4/4	×	48.0	55.5	40.0	80.5	62.2	57.2
QVGen	4/4	✓	52.8	59.3	37.6	79.1	**62.4**	58.2
Ours	4/4	✓	**54.6**	**63.4**	**37.8**	**81.3**	**62.4**	**59.9**
QVGen	3/3	×	47.2	60.7	31.2	77.3	57.6	54.8
Ours	3/3	×	**48.5**	**62.0**	**38.3**	**80.3**	**58.0**	**57.4**

aligning supervision with denoising stages, while Denoising-Stage Gated Guidance (CFG-drop) targets late-step guidance amplification. Combining the two components yields the best overall performance, suggesting that the training-side and inference-side designs are complementary. The full-precision CFG-drop control is reported in Appendix C, with the available VBench dimensions in Table 12. Representative qualitative comparisons are provided in the supplementary material.

Ablation on schedulers for Denoising-Stage-Oriented Supervision. Table 5 compares linear and exponential schedules for balancing distillation loss and target loss. The exponential schedule with $\alpha = 5$ performs best, achieving the highest VBench average and visual quality scores. This suggests that a moderately sharp decay of the distillation loss better balances teacher guidance at early timesteps and target-loss optimization at later timesteps, while an overly aggressive decay with $\alpha = 10$ harms performance.

Ablation on the threshold of Denoising-Stage-Gated Guidance. Fig. 4 illustrates the impact of the CFG-drop threshold on both visual clarity and VBench scores. We observe that applying CFG-drop

Table 4: **Component ablation on Wan2.1-1.3B (W4A4, sym).** We evaluate Denoising-Stage Oriented Supervision (mixed loss) and Denoising-Stage Gated Guidance (CFG-drop) independently. "Base" denotes the matched KD-only quantized baseline with neither component (Appendix I.1). "LV" denotes Laplacian Variance. Best results are highlighted in **bold**; our full framework is shaded.

Method	Mixed loss	CFG drop	#Bits (W/A)	VBench1↑						Visual Quality↑		
				Imaging Quality	Aesthetic Quality	Dynamic Degree	Scene Consistency	Overall Consistency	Avg	LV	Vision Reward	Instruc Following
Wan 1.3B (CFG $= 5.0, 480p,$ fps $= 16$)												
FP	–	–	16/16	64.22	57.85	71.11	26.07	24.70	66.3	808	7.51	37.5
Base	×	×	4/4	62.86	56.32	71.67	22.94	24.35	65.1	674	1.59	25.8
Base + CFG-drop	×	✓	4/4	61.67	56.75	77.50	25.58	24.20	66.3	842	4.51	28.9
Base + Mixed loss	✓	×	4/4	**64.06**	57.47	76.67	25.16	**24.55**	66.2	806	4.04	29.1
Ours	✓	✓	4/4	63.52	**58.01**	**80.00**	**26.54**	24.49	**67.2**	**1026**	**6.47**	**34.5**

Table 5: **Ablation on the mixing schedule of Denoising-Stage Oriented Supervision** (Wan2.1-1.3B, W4A4, sym). "linear" uses a linear schedule; "expα" uses the exponential schedule of Eq. 6 with parameter α. CFG-drop is applied by default. Our choice (exp5) is shaded; best results are in **bold**.

Method		#Bits (W/A)	Sym	VBench1↑						Visual Quality↑		
				Imaging Quality	Aesthetic Quality	Dynamic Degree	Scene Consistency	Overall Consistency	Avg	LV	Vision Reward	Instruc Following
Wan 1.3B (CFG $= 5.0, 480p,$ fps $= 16$)												
FP		16/16	–	64.22	57.85	71.11	26.07	24.70	66.3	808	7.51	37.5
Mix loss	linear	4/4	✓	62.81	57.13	70.55	20.21	23.84	65.2	862	3.80	29.7
	exp1	4/4	✓	63.42	57.03	73.61	22.60	23.85	65.9	887	4.11	28.4
	exp5	4/4	✓	**63.52**	**58.01**	**80.00**	**26.54**	**24.49**	**67.2**	**1026**	**6.47**	**34.5**
	exp10	4/4	✓	63.50	57.76	74.70	18.79	23.43	65.6	785	4.19	24.8

even to only the final three timesteps yields a substantial boost in both metrics, effectively suppressing the amplification of quantization errors induced by CFG. However, when CFG-drop is extended beyond ten steps, the VBench score begins to decline, suggesting that premature removal of CFG compromises the semantic integrity of the generated video. Consequently, we set the threshold to nine steps, achieving a near-optimal balance between VBench performance and visual clarity.

5.4 Efficiency analysis

Training efficiency. Table 6 reports raw GPU-days together with the GPU type used for each method. In our training setup, H100 is approximately 2.5×–2.8× faster than H20[1], so after normalizing both methods to H100-equivalent GPU-days, DSAQuant reduces training cost by roughly 2×–7×, with the largest gap on Wan 14B, indicating favorable scaling to large video diffusion models.

Inference efficiency. Table 7 shows that our W4A4 model consistently improves inference

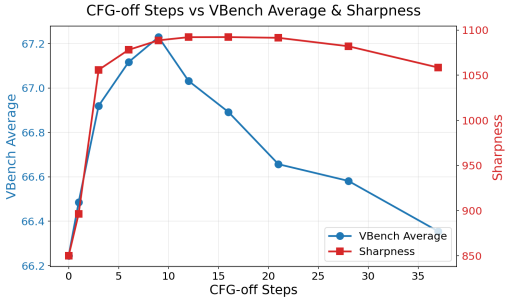

Figure 4: Ablation on the threshold of Denoising-Stage Gated Guidance (CFG-drop).

Table 6: **Training cost on four VDMs, reported in raw GPU-days.** QVGen* times are taken from the QVGen paper [11] (measured on H100); ours are measured on H20 under the settings in Appendix D. The per-row GPU is shown explicitly to avoid hardware-implicit comparison.

Method	GPU	CogVideoX-2B	Wan 1.3B	CogVideoX1.5-5B	Wan 14B
QVGen*	H100	9.44	11.11	∼ 51	∼ 182
Ours	H20	6.07	8.19	∼ 65	∼ 64

[1]The H100 SXM and H20 are rated at 989 and 148 BF16 TFLOPS, respectively, per NVIDIA's official specifications; the practical training-throughput gap is narrower than the ∼6.7× peak ratio because H20's higher HBM bandwidth (4.0 vs. 3.35 TB/s) partially compensates the compute deficit on memory-bound operators (attention, optimizer updates, gradient synchronization).

Table 7: **Memory and inference latency on the Wan series, measured under 81 frames at 480p.**

Model	Methods	Storage (GB)	Peak inference memory (GB)	A6000 latency (s)	4090D latency (s)
Wan 2.1-1.3B	FP16	2.6	8.7	313	246
	Ours w4a4	0.7 (-73.1%)	7.0 (-19.5%)	264 (-15.7%)	193 (-21.5%)
Wan 2.1-14B	FP16	26.6	37.7	1845	OOM
	Ours w4a4	6.7 (-74.8%)	20.7 (-45.1%)	1268 (-31.3%)	930
Wan 2.2-5B	FP16	9.5	16.9	129	100
	Ours w4a4	2.4 (-74.7%)	10.9 (-35.5%)	80 (-38.0%)	58 (-42.0%)

efficiency on the Wan series. Compared with FP16, it reduces storage by over 73%, peak memory by up to 45.1%, and A6000 latency by up to 38.0%. Notably, our quantized Wan 14B can run on 4090D, while the FP16 model leads to OOM. Moreover, our technique is orthogonal to attention acceleration methods, and combined results are provided in Appendix M.

5.5 Additional comparisons and generalization

Beyond the component ablations, we examine whether the benefits of stage alignment persist under different supervision objectives, sampling configurations, and model representations. For fast comparison, we sample one video per prompt for experiments in this subsection.

Preserving full-precision model behavior. A target-only objective can improve fitting to the training data while shifting the quantized model away from its FP teacher. With the same CFG-drop setting, DSAQuant achieves LPIPS/FVD of $0.44/975$ relative to FP outputs, compared with $0.63/1623$ for target-only QAT and $0.43/962$ for KD-only QAT (Appendix J). Together with the quality gains in Table 4, this supports using early-stage distillation to preserve model behavior while introducing target supervision for detail refinement.

Robustness to sampling choices. On Wan2.1-1.3B, the two components consistently improve VBench scores under 20-step FlowMatch and FlowMatch-UniPC sampling with guidance scales of 3.5, 5, and 7. On Wan2.2-5B, dropping CFG in the last 6–12 steps of a 50-step trajectory also improves over the no-drop setting. These results support interpreting the intervention as a late-stage window whose benefit persists across sampling configurations, rather than a single fixed step threshold (Appendix K).

Transfer across numerical formats and sampling trajectories. We further test whether the stage-aware design extends beyond the main INT quantization settings. With NVFP4 weights and activations, DSAQuant raises augmented-prompt VBench from 68.3 to 70.0 on Wan2.1-1.3B. On the four-step rCM model under W4A8 quantization, it improves VBench from 64.8 to 66.2 while retaining the native four-step trajectory during QAT. These results demonstrate transfer to a floating-point quantization format and a step-distilled video generator (Appendix L).

6 Conclusion

We present DSAQuant, a denoising-stage aligned quantization-aware training framework for video diffusion models. Our key observation is that conventional QAT preserves coarse semantics but disrupts the detail-refinement capability of later denoising steps, resulting in degraded visual fidelity. DSAQuant addresses this issue through Denoising-stage-Oriented Supervision during training and Denoising-stage-Gated Guidance during inference, aligning both supervision and guidance with the stage-wise behavior of video denoising. Experiments demonstrate that DSAQuant improves visual quality, sharpness, and temporal stability while maintaining text-video alignment, highlighting the importance of stage-aware design for high-fidelity VDM quantization.

Acknowledgments

We would like to thank Yichong Lu, Jiahao Wang, Yufen Yuan, Nan Zhou, Jiahao Shao, and Yudong Jin for their insightful discussions.

References

[1] Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms, 2024.

[2] 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, 2024.

[3] Juncan Deng, Shuaiting Li, Zeyu Wang, Hong Gu, Kedong Xu, and Kejie Huang. Vq4dit: Efficient post-training vector quantization for diffusion transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 16226–16234, 2025.

[4] DiffSynth-Studio Contributors. DiffSynth-Studio. `https://github.com/modelscope/DiffSynth-Studio`. Public diffusion-model training and inference framework.

[5] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization. In *International Conference on Learning Representations*, 2020.

[6] Weilun Feng, Haotong Qin, Chuanguang Yang, Xiangqi Li, Han Yang, Yuqi Li, Zhulin An, Libo Huang, Michele Magno, and Yongjun Xu. S^2Q-VDiT: Accurate quantized video diffusion transformer with salient data and sparse token distillation. *arXiv preprint arXiv:2508.04016*, 2025.

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

[8] Yefei He, Luping Liu, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Ptqd: Accurate post-training quantization for diffusion models, 2023.

[9] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

[10] Yushi Huang, Ruihao Gong, Jing Liu, Tianlong Chen, and Xianglong Liu. Tfmq-dm: Temporal feature maintenance quantization for diffusion models, 2024.

[11] Yushi Huang, Ruihao Gong, Jing Liu, Yifu Ding, Chengtao Lv, Haotong Qin, and Jun Zhang. QVGen: Pushing the limit of quantized video generative models. In *The Fourteenth International Conference on Learning Representations*, 2026.

[12] Yushi Huang, Ruihao Gong, Xianglong Liu, Jing Liu, Yuhang Li, Jiwen Lu, and Dacheng Tao. Temporal feature matters: A framework for diffusion model quantization, 2025.

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

[14] 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, 2017.

[15] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. Hunyuanvideo: A systematic framework for large video generative models, 2025.

[16] 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.

[17] Shuaiting Li, Juncan Deng, Chengxuan Wang, Kedong Xu, Rongtao Deng, Hong Gu, Haibin Shen, and Kejie Huang. Ssvq: Unleashing the potential of vector quantization with sign-splitting. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 23710–23719. IEEE, 2025.

[18] Shuaiting Li, Juncan Deng, Zeyu Wang, Kedong Xu, Rongtao Deng, Hong Gu, Haibin Shen, and Kejie Huang. Efficiency meets fidelity: A novel quantization framework for stable diffusion, 2025.

[19] Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models, 2023.

[20] Yanjing Li, Sheng Xu, Xianbin Cao, Xiao Sun, and Baochang Zhang. Q-dm: An efficient low-bit quantized diffusion model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[21] Shanchuan Lin, Xin Xia, Yuxi Ren, Ceyuan Yang, Xuefeng Xiao, and Lu Jiang. Diffusion adversarial post-training for one-step video generation, 2025.

[22] Zhengyao Lv, Chenyang Si, Junhao Song, Zhenyu Yang, Yu Qiao, Ziwei Liu, and Kwan-Yee K. Wong. Fastercache: Training-free video diffusion model acceleration with high quality. In *The Thirteenth International Conference on Learning Representations*, 2025.

[23] 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.

[24] Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models, 2023.

[25] Junhyuk So, Jungwon Lee, Daehyun Ahn, Hyungjun Kim, and Eunhyeok Park. Temporal dynamic quantization for diffusion models, 2023.

[26] Yang Sui, Yanyu Li, Anil Kag, Yerlan Idelbayev, Junli Cao, Ju Hu, Dhritiman Sagar, Bo Yuan, Sergey Tulyakov, and Jian Ren. Bitsfusion: 1.99 bits weight quantization of diffusion model. *Advances in Neural Information Processing Systems*, 37:76775–76818, 2024.

[27] Shilong Tian, Hong Chen, Chengtao Lv, Yu Liu, Jinyang Guo, Xianglong Liu, Shengxi Li, Hao Yang, and Tao Xie. Qvd: Post-training quantization for video diffusion models, 2024.

[28] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

[29] Changyuan Wang, Ziwei Wang, Xiuwei Xu, Yansong Tang, Jie Zhou, and Jiwen Lu. Towards accurate post-training quantization for diffusion models, 2024.

[30] Wenhao Wang and Yi Yang. Vidprom: A million-scale real prompt-gallery dataset for text-to-video diffusion models. 2024.

[31] WanTeam, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models, 2025.

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

[33] Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, Jianfei Chen, Ion Stoica, Kurt Keutzer, and Song Han. Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity, 2025.

[34] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer, 2025.

[35] Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *CVPR*, 2025.

[36] Jintao Zhang, Kaiwen Zheng, Kai Jiang, Haoxu Wang, Ion Stoica, Joseph E Gonzalez, Jianfei Chen, and Jun Zhu. Turbodiffusion: Accelerating video diffusion models by 100-200 times. *arXiv preprint arXiv:2512.16093*, 2025.

[37] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[38] Tianchen Zhao, Tongcheng Fang, Haofeng Huang, Rui Wan, Widyadewi Soedarmadji, Enshu Liu, Shiyao Li, Zinan Lin, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. Vidit-q: Efficient and accurate quantization of diffusion transformers for image and video generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[39] Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, Yu Qiao, and Ziwei Liu. Vbench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness, 2025.

[40] Kaiwen Zheng, Yuji Wang, Qianli Ma, Huayu Chen, Jintao Zhang, Yogesh Balaji, Jianfei Chen, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Large scale diffusion distillation via score-regularized continuous-time consistency. In *International Conference on Learning Representations*, 2026.

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

[42] Xin Zhou, Dingkang Liang, Kaijin Chen, Tianrui Feng, Xiwu Chen, Hongkai Lin, Yikang Ding, Feiyang Tan, Hengshuang Zhao, and Xiang Bai. Less is enough: Training-free video diffusion acceleration via runtime-adaptive caching. *arXiv preprint arXiv:2507.02860*, 2025.

[43] Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation, 2026.

[44] Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. In *The Thirteenth International Conference on Learning Representations*, 2025.

A Gradient Variance Analysis of Early-Stage Distillation

This section analyzes the variance-reduction aspect of early-stage distillation. The trajectory-anchoring effect follows from the KD objective itself, which explicitly penalizes deviations between the quantized student and the full-precision teacher.

We use the notation in Sec. 4.1. For a fixed noisy latent, condition, and timestep, let

$$z = (x_t, c, t), \qquad s_Q(z) = f^q(x_t, c, t), \qquad J_Q(z) = \nabla_{\theta_Q} f^q(x_t, c, t), \tag{12}$$

where J_Q denotes the surrogate Jacobian used by the Straight-Through Estimator (STE). Under squared-error training, the optimal predictor for the native diffusion or flow target is the conditional mean

$$m(z) = \mathbb{E}[y_t \mid z]\,. \tag{13}$$

We decompose the sample-wise target as

$$y_t = m(z) + \xi_t, \qquad \mathbb{E}[\xi_t \mid z] = 0, \qquad \mathrm{Cov}(\xi_t \mid z) = \Sigma_t(z). \tag{14}$$

Then the conditional expected target loss satisfies

$$\mathbb{E}\Big[\|s_Q(z) - y_t\|_2^2 \mid z\Big] = \|s_Q(z) - m(z)\|_2^2 + \mathrm{Tr}\,\Sigma_t(z). \tag{15}$$

Thus, the sample-wise target loss has the correct population optimum, but its stochastic gradient contains posterior target noise. Specifically, the sample gradient of the target loss in Eq. 3 is

$$g_{\mathrm{Target}}(z, y_t) = \nabla_{\theta_Q} \mathcal{L}_{\mathrm{Target}}(t) = 2J_Q(z)^\top \left(s_Q(z) - y_t\right). \tag{16}$$

and therefore

$$\mathrm{Cov}(g_{\mathrm{Target}} \mid z) = 4J_Q(z)^\top \Sigma_t(z) J_Q(z). \tag{17}$$

In early denoising steps, x_t is dominated by noise and contains limited information about the clean video latent x_1. For flow, velocity, and data-related targets of the generic form $y_t = a_t x_1 + b_t \epsilon$, many plausible clean endpoints remain compatible with the same noisy state. Consequently, the posterior covariance $\Sigma_t(z)$ can be large, and Eq. (17) shows that this uncertainty is directly injected into the STE gradient. Such high-variance updates are especially harmful in low-bit QAT, where small gradient fluctuations can frequently move latent weights across quantization bin boundaries.

Distillation replaces the sample-wise target with the full-precision teacher prediction. A well-trained teacher can be written as

$$f(x_t, c, t) = m(z) + \delta(z), \tag{18}$$

where $\delta(z)$ denotes the teacher residual with respect to the conditional mean. The sample gradient of the distillation loss in Eq. 4 is

$$g_{\mathrm{KD}}(z) = \nabla_{\theta_Q} \mathcal{L}_{\mathrm{KD}}(t) = 2J_Q(z)^\top \left(s_Q(z) - f(x_t, c, t)\right). \tag{19}$$

Conditioned on the same z, the teacher output is deterministic, so the label-induced conditional covariance is removed:

$$\mathrm{Cov}(g_{\mathrm{KD}} \mid z) = 0. \tag{20}$$

Distillation therefore trades the variance term induced by ξ_t for the teacher residual $\delta(z)$. In early high-noise steps, the posterior target variance is large, while the teacher provides a stable estimate of the conditional denoising direction. KD is therefore preferable as a variance-reduction mechanism: it stabilizes QAT updates and anchors the global denoising trajectory.

This argument also explains why distillation should not dominate all timesteps. As denoising approaches the low-noise refinement regime, the variance-reduction benefit of KD becomes less dominant, while the teacher residual $\delta(z)$ and the limited capacity of the quantized hypothesis space become more important. In this regime, optimizing the native target loss better aligns the quantized model with detail reconstruction instead of forcing it to imitate the full-precision teacher residual. This motivates our denoising-stage schedule, which uses KD in early steps and gradually shifts supervision toward the target loss in later steps.

B Two-Stage Analysis of Late-Step CFG Quantization Error

We analyze late-step quantization degradation as a two-stage process: a mismatch first forms between the conditional and unconditional branch errors, and CFG then amplifies this mismatch. To isolate quantization effects from denoising-trajectory accumulation, we compare FP and quantized predictions under the same latent state, prompt, and timestep. Define

$$e_c(t) = f_c^q(t) - f_c(t), \qquad e_\emptyset(t) = f_\emptyset^q(t) - f_\emptyset(t), \tag{21}$$

where $f_c, f_\emptyset$ and $f_c^q, f_\emptyset^q$ are the FP and quantized conditional and unconditional predictions, respectively.

Stage I: Formation of branch mismatch. We first explain why the two branch errors need not cancel under dynamic activation quantization. Consider a dynamic quantizer

$$Q(h) = s(h)\,\mathrm{round}\left(\frac{h}{s(h)}\right), \qquad e(h) = Q(h) - h, \tag{22}$$

where the scale $s(h)$ is computed from the current activation tensor, token, or channel group. Although the two CFG branches share the same weights and noisy latent, they receive different text conditions. As these signals propagate through cross-attention, normalization, and feed-forward layers, they induce different activation distributions:

$$h_c^\ell \neq h_\emptyset^\ell, \qquad s_c^\ell = s(h_c^\ell), \qquad s_\emptyset^\ell = s(h_\emptyset^\ell). \tag{23}$$

In general, $s_c^\ell \neq s_\emptyset^\ell$, so the branches are rounded on different quantization grids. Their layer-wise errors are therefore not perfectly aligned, and the accumulated output-level errors $e_c(t)$ and $e_\emptyset(t)$ need not cancel. We characterize their alignment by

$$\rho_t = \frac{\mathbb{E}[\langle e_c(t), e_\emptyset(t)\rangle]}{\sqrt{\mathbb{E}\|e_c(t)\|_2^2 \mathbb{E}\|e_\emptyset(t)\|_2^2}}. \tag{24}$$

This formulation does not assume independence: the branches may share correlated error because they use the same weights and latent input. The relevant quantity is the residual mismatch after any correlated components cancel.

Let

$$\sigma_c^2(t) = \mathbb{E}\|e_c(t)\|_2^2, \qquad \sigma_\emptyset^2(t) = \mathbb{E}\|e_\emptyset(t)\|_2^2, \tag{25}$$

and define the branch mismatch as

$$\Delta^q(t) = e_c(t) - e_\emptyset(t). \tag{26}$$

Its expected energy is

$$\mathbb{E}\|\Delta^q(t)\|_2^2 = \sigma_c^2(t) + \sigma_\emptyset^2(t) - 2\rho_t \sigma_c(t)\sigma_\emptyset(t). \tag{27}$$

Equation (27) separates two causes of a larger late-step mismatch. First, the individual branch errors can increase as increasingly structured late-stage activations become more sensitive to low-bit quantization. Second, weaker cross-branch error correlation reduces cancellation between the two errors. Together, these effects enlarge $\Delta^q(t)$ before CFG is applied.

Stage II: Amplification of branch mismatch by CFG. The FP and quantized guided predictions under the same state are

$$\hat{f}_\omega(t) = f_\emptyset(t) + \omega\big(f_c(t) - f_\emptyset(t)\big), \qquad \hat{f}_\omega^q(t) = f_\emptyset^q(t) + \omega\big(f_c^q(t) - f_\emptyset^q(t)\big). \tag{28}$$

Their difference gives the final guided quantization error:

$$\delta_\omega^q(t) = \hat{f}_\omega^q(t) - \hat{f}_\omega(t) = \underbrace{e_\emptyset(t)}_{\text{single-branch error}} + \omega\ \underbrace{\big(e_c(t) - e_\emptyset(t)\big)}_{\text{branch mismatch } \Delta^q(t)}. \tag{29}$$

Equation (29) makes the second stage explicit: after the branch mismatch $\Delta^q(t)$ has formed in Stage I, CFG directly scales its contribution to the final guided error by ω.

Table 8: **Identical CFG-drop controls on Wan2.1-1.3B.** We compare each model with and without the same late-step CFG-drop schedule. QVGen and Base use symmetric W4A4 quantization; the full-precision model uses BF16.

Method	#Bits (W/A)	VBench Avg↑			LV↑		
		w/o drop	w/ drop	Δ	w/o drop	w/ drop	Δ
Full Prec.	16/16	66.28	66.44	+0.16	808	850	+42
QVGen	4/4	63.50	64.73	+1.23	665	856	+191
Base	4/4	65.09	66.31	+1.22	674	842	+168

This amplification is most harmful in late denoising steps. In this regime, the useful FP guidance direction

$$f_c(t) - f_\emptyset(t) \tag{30}$$

becomes relatively weak because the latent already contains rich visual structure, while the branch mismatch formed in Stage I can continue to grow. The amplified term $\omega\Delta^q(t)$ can therefore dominate the useful guidance signal and appear as high-frequency artifacts. Denoising-Stage Gated Guidance removes this multiplier in the final steps by using only the conditional prediction, suppressing Stage II amplification after the useful FP guidance direction has diminished.

C General and Quantization-Specific Effects of CFG-Drop

Late-step CFG suppression can also benefit full-precision diffusion sampling, raising the question of whether CFG-drop is merely a general sampling heuristic. To separate this generic effect from its role in quantized models, we apply an identical CFG-drop schedule to the full-precision model, QVGen, and our matched KD-only quantized baseline (Base; Appendix I.1), while keeping all other inference settings fixed. Table 8 reports both the VBench average and Laplacian Variance (LV), where LV measures visual sharpness.

CFG-drop yields a small improvement for the full-precision model, so we do not attribute its entire effect exclusively to quantization. However, both quantized models receive substantially larger gains in VBench average and, especially, visual sharpness. These controlled results show that the improvement cannot be explained solely by a generic sampling benefit. Instead, CFG-drop is particularly effective under low-bit quantization, consistent with the mechanism in Appendix B: once the late-stage branch mismatch has formed, removing CFG prevents this quantization-specific error from being further amplified.

D Detailed Training Settings

The detailed training settings including batch size, steps, training frames, and learning rate are provided in Tab. 9. We use the Adam optimizer with zero weight decay; the learning rate for each model is listed in Tab. 9. We train our model on H20 GPUs with 140GB memory. By default we use local batch=1 for training, so the global batch size is equal to the number of GPUs we use.

For training data, we use synthetic samples. For Wan2.1-1.3B, we synthesize data using the publicly released VidProM-derived prompt set [30], following prior work [9]. For Wan2.1-14B, we directly use the synthetic latent data released by TurboDiffusion [36]. For the other models, to reduce data synthesis cost, we decode the TurboDiffusion latents into videos with the Wan2.1 VAE and then re-encode the videos with the corresponding model-specific VAE to obtain latent training data.

E Detailed Evaluation Settings

Following QVGen [11], we set the random seed according to the process rank at the beginning of evaluation, ensuring that each generated video is sampled from different random noise. We use 128 GPUs for inference on Wan2.1-14B and 32 GPUs for all other models.

Table 9: Detailed training settings for each model. We set bz=1 for each gpu to evaluate training time.

Models	Batch size	Total steps	Warmup steps	Lr	Frames	Resolution	Time (H20 days)
CogVideoX-2B	32	2000	200	3e-5	49	480p	6.07
CogVideoX1.5-5B	32	2000	200	5e-5	81	720p	∼65
Wan2.1 1.3B	24	2000	200	3e-5	81	480p	8.19
Wan2.1 14B	64	1000	100	3e-5	81	480p	∼64
Wan2.2 5B	32	4000	400	3e-5	81	480p	6.9

F Timestep-Conditioned Linear Layers and LUT Precomputation

Previous quantization research [26, 18, 10] on image diffusion models has demonstrated that precomputing time features during inference can enhance accuracy without compromising inference efficiency. This is feasible because, during the inference phase, timesteps are discrete integers (e.g., 1,000 distinct values ranging from 0 to 999), forming a finite set. Therefore, we can pre-compute the time features for all timesteps and store them on the CPU. During inference, only the specific feature corresponding to the current timestep is dynamically loaded onto the GPU. Consequently, all timestep-related linear layers can be entirely removed without incurring any additional GPU memory overhead. Below, we detail the procedure for precomputing the Look-Up Table (LUT) for CogVideoX and Wan, alongside a size comparison between the generated LUT and the original linear layers.

We analyze the timestep-conditioned computation in CogVideoX-2B and Wan2.1-T2V-1.3B. Let the inference scheduler use a fixed set of denoising timesteps

$$\mathcal{T}_K = \{t_1, t_2, \ldots, t_K\}, \tag{31}$$

where we use $K = 50$ in the main comparison. Unless otherwise specified, all storage costs are computed with fp16/bf16 precision, i.e., 2 bytes per scalar.

F.1 CogVideoX-2B

For CogVideoX-2B, the hidden dimension and timestep embedding dimension are

$$D_c = 30 \times 64 = 1920, \qquad E_c = 512, \qquad L_c = 30. \tag{32}$$

Given a scheduler timestep t, CogVideoX first computes a sinusoidal timestep embedding and then maps it through a timestep MLP:

$$p_t = \mathrm{SinEmb}_{D_c}(t) \in \mathbb{R}^{D_c}, \tag{33}$$

$$h_t = \mathrm{TEmb}(p_t) \in \mathbb{R}^{E_c}. \tag{34}$$

Here $\mathrm{SinEmb}_{D_c}(\cdot)$ is deterministic and has no trainable linear weight. The trainable timestep MLP can be written as

$$h_t = W_t^{(2)} \sigma\left(W_t^{(1)} p_t + b_t^{(1)}\right) + b_t^{(2)}, \tag{35}$$

where $\sigma(\cdot)$ denotes the activation function.

Each CogVideoX transformer block contains two timestep-conditioned AdaLN-Zero modules, one before attention and one before the feed-forward network. For block $\ell \in \{1, \ldots, L_c\}$ and AdaLN index $r \in \{1, 2\}$, the timestep-conditioned modulation vector is

$$a_{\ell,r}(t) = W_{\ell,r}\sigma(h_t) + b_{\ell,r} \in \mathbb{R}^{6D_c}. \tag{36}$$

This vector is split into six per-channel vectors:

$$a_{\ell,r}(t) = \left[\beta^v_{\ell,r}(t), \gamma^v_{\ell,r}(t), g^v_{\ell,r}(t), \beta^e_{\ell,r}(t), \gamma^e_{\ell,r}(t), g^e_{\ell,r}(t)\right], \tag{37}$$

where

$$\beta^v_{\ell,r}, \gamma^v_{\ell,r}, g^v_{\ell,r} \in \mathbb{R}^{D_c}, \qquad \beta^e_{\ell,r}, \gamma^e_{\ell,r}, g^e_{\ell,r} \in \mathbb{R}^{D_c}. \tag{38}$$

The final output AdaLN produces only shift and scale:

$$o(t) = W_o\sigma(h_t) + b_o \in \mathbb{R}^{2D_c}, \tag{39}$$

$$o(t) = [\beta_o(t), \gamma_o(t)], \qquad \beta_o(t), \gamma_o(t) \in \mathbb{R}^{D_c}. \tag{40}$$

Therefore, for a fixed inference timestep set $\mathcal{T}_K$, the CogVideoX timestep LUT can be defined as

$$\mathcal{L}_{\text{Cog}} = \{a_{\ell,r}(t_s), o(t_s) \mid t_s \in \mathcal{T}_K, \ell = 1, \ldots, L_c, r \in \{1, 2\}\}. \tag{41}$$

The number of cached scalar values is

$$|\mathcal{L}_{\text{Cog}}| = K\left(2L_c \cdot 6D_c + 2D_c\right). \tag{42}$$

Substituting $K = 50$, $L_c = 30$, and $D_c = 1920$ gives

$$|\mathcal{L}_{\text{Cog}}| = 50\left(2 \cdot 30 \cdot 6 \cdot 1920 + 2 \cdot 1920\right) = 34{,}752{,}000. \tag{43}$$

Thus, the fp16/bf16 LUT size is

$$34{,}752{,}000 \times 2 = 69{,}504{,}000 \text{ bytes} \approx 69.50 \text{ MB} \approx 66.28 \text{ MiB}. \tag{44}$$

The total number of timestep-related linear parameters in CogVideoX-2B is

$$\begin{aligned} N_{\text{Cog}} = &\underbrace{(D_cE_c + E_c) + (E_c^2 + E_c)}_{\text{time embedding MLP}} \\ &+ \underbrace{2L_c\left((6D_c)E_c + 6D_c\right)}_{\text{block AdaLN linears}} \\ &+ \underbrace{(2D_c)E_c + 2D_c}_{\text{final AdaLN linear}}. \end{aligned} \tag{45}$$

Substituting $D_c = 1920$, $E_c = 512$, and $L_c = 30$:

$$N_{\text{Cog}} = 357{,}801{,}728. \tag{46}$$

Therefore, the fp16/bf16 parameter storage is

$$357{,}801{,}728 \times 2 = 715{,}603{,}456 \text{ bytes} \approx 715.60 \text{ MB} \approx 682.45 \text{ MiB}. \tag{47}$$

F.2 Wan2.1-T2V-1.3B

For Wan2.1-T2V-1.3B, the hidden dimension, sinusoidal frequency dimension, and number of transformer layers are

$$D_w = 1536, \qquad F_w = 256, \qquad L_w = 30. \tag{48}$$

Given timestep t, Wan first computes a sinusoidal embedding:

$$r_t = \text{SinEmb}_{F_w}(t) \in \mathbb{R}^{F_w}. \tag{49}$$

Then the global timestep embedding is computed as

$$e_t = W_e^{(2)}\sigma\left(W_e^{(1)}r_t + b_e^{(1)}\right) + b_e^{(2)} \in \mathbb{R}^{D_w}. \tag{50}$$

Wan then computes a global six-way timestep projection:

$$e_t^0 = \text{reshape}\left(W_p\sigma(e_t) + b_p\right) \in \mathbb{R}^{6\times D_w}. \tag{51}$$

Unlike CogVideoX-2B, Wan does not use a separate large timestep-conditioned linear layer inside each transformer block. Instead, each block ℓ has a learned modulation tensor

$$M_\ell \in \mathbb{R}^{6\times D_w}. \tag{52}$$

The block-specific timestep modulation is obtained by a lightweight element-wise addition:

$$q_\ell(t) = M_\ell + e_t^0 \in \mathbb{R}^{6\times D_w}. \tag{53}$$

This tensor is split into six vectors:

$$q_\ell(t) = \left[\beta_\ell^a(t), \gamma_\ell^a(t), g_\ell^a(t), \beta_\ell^f(t), \gamma_\ell^f(t), g_\ell^f(t)\right], \tag{54}$$

where the superscripts a and f denote the self-attention and feed-forward branches, respectively.

The final output head also uses the global timestep embedding e_t. Let the head modulation be

$$M_{\text{head}} \in \mathbb{R}^{2\times D_w}. \tag{55}$$

Then

$$q_{\text{head}}(t) = M_{\text{head}} + e_t \in \mathbb{R}^{2\times D_w}, \tag{56}$$

which is split as

$$q_{\text{head}}(t) = \left[\beta_{\text{head}}(t), \gamma_{\text{head}}(t)\right]. \tag{57}$$

Therefore, for Wan2.1-T2V-1.3B, it is sufficient to cache only the global timestep quantities:

$$\mathcal{L}_{\text{Wan}} = \left\{ e_{t_s}, e_{t_s}^0 \;\middle|\; t_s \in \mathcal{T}_K \right\}. \tag{58}$$

The number of cached scalar values is

$$|\mathcal{L}_{\text{Wan}}| = K\left(D_w + 6D_w\right) = 7KD_w. \tag{59}$$

Substituting $K = 50$ and $D_w = 1536$ gives

$$|\mathcal{L}_{\text{Wan}}| = 50 \times 7 \times 1536 = 537{,}600. \tag{60}$$

Thus, the fp16/bf16 LUT size is

$$537{,}600 \times 2 = 1{,}075{,}200 \text{ bytes} \approx 1.075 \text{ MB} \approx 1.025 \text{ MiB}. \tag{61}$$

The total number of timestep-related linear parameters in Wan2.1-T2V-1.3B is

$$N_{\text{Wan}} = \underbrace{(F_w D_w + D_w) + (D_w^2 + D_w)}_{\text{time embedding MLP}} + \underbrace{(6D_w)D_w + 6D_w}_{\text{global time projection}}. \tag{62}$$

Substituting $F_w = 256$ and $D_w = 1536$:

$$N_{\text{Wan}} = 16{,}920{,}576. \tag{63}$$

Therefore, the fp16/bf16 parameter storage is

$$16{,}920{,}576 \times 2 = 33{,}841{,}152 \text{ bytes} \approx 33.84 \text{ MB} \approx 32.27 \text{ MiB}. \tag{64}$$

F.3 Why LUT Precomputation is Feasible

For both models, the cached quantities are deterministic functions of the discrete scheduler timestep and fixed model weights:

$$\mathcal{L}(t) = f_\theta(t), \qquad t \in \mathcal{T}_K, \tag{65}$$

where θ denotes the frozen parameters of the timestep-conditioned linear layers.

Importantly, these quantities do not depend on the latent tokens, text tokens, attention activations, or feed-forward activations:

$$\frac{\partial \mathcal{L}(t)}{\partial X} = 0, \qquad \frac{\partial \mathcal{L}(t)}{\partial C} = 0. \tag{66}$$

Thus, given a fixed inference scheduler, the timestep-conditioned linear outputs can be precomputed once before denoising:

$$\left\{ f_\theta(t_1), f_\theta(t_2), \ldots, f_\theta(t_K) \right\}, \tag{67}$$

and reused throughout the denoising process by indexing the LUT with the current timestep.

This removes online execution of the timestep-related linear layers while preserving the original LayerNorm, affine modulation, attention, feed-forward, and residual-gating computation paths.

Table 10: Comparison of timestep-related linear parameters and 50-step LUT size.

Model	Time-related linear scope	Params	Weight size	50-step LUT content	LUT size
CogVideoX-2B	time embedding MLP + block AdaLN linears + final AdaLN linear	357,801,728	715.60 MB / 682.45 MiB	all block AdaLN parameters + final AdaLN parameters	69.50 MB / 66.28 MiB
Wan2.1-T2V-1.3B	time embedding MLP + global time projection	16,920,576	33.84 MB / 32.27 MiB	global $e_t \in \mathbb{R}^{1536}$ and $e_t^0 \in \mathbb{R}^{6\times 1536}$	1.075 MB / 1.025 MiB

Table 11: Supplementary table for VBench results in Tab. 1.

Method	#Bits (W/A)	VBench1↑								
		Imaging Quality	Aesthetic Quality	Motion Smoothness	Dynamic Degree	Background Consistency	Subject Consistency	Scene Consistency	Overall Consistency	Avg
`Wan2.1` 1.3B (`CFG` $= 5.0, 480p$, `fps` $= 16$)										
FP	16/16	64.22	57.85	97.25	71.11	95.72	93.35	26.07	24.70	66.28
QVGen	3/3	59.48	50.33	97.91	42.50	94.78	89.96	11.68	20.32	58.37
Ours	3/3	61.90	54.56	96.41	77.77	94.46	90.88	22.46	24.11	65.32
`Wan2.2` 5B (`CFG` $= 5.0, 720p$, `fps` $= 16$)										
QVGen	3/3	60.65	54.95	97.97	55.0	95.08	92.43	19.56	23.49	62.39
Ours	3/3	61.22	58.53	98.50	60.00	95.74	94.92	24.26	25.27	64.81

F.4 Storage Comparison

The total storage cost comparison is summarized in Tab. 10.

G Supplementary Results for Main Experiments

This section provides supplementary VBench results, summarized in Tab. 11, for the main comparison in Tab. 1. We focus on the more aggressive W3A3 setting for Wan2.1-1.3B and Wan2.2-5B, complementing the main-table results with per-dimension scores under standard prompts.

H Full VBench Results For Ablation Studies

Due to page constraints in the main text, the ablation study section only reports five sensitive metrics from VBench. The comprehensive evaluation results across all metrics are summarized in Tabs. 12 and 13.

I Matched KD-Only Baselines

I.1 Definition of the matched KD-only baseline

QVGen [11] uses KD-based QAT with an auxiliary module for weight quantization. For CogVideoX, the KD-only Base and QVGen use the same training codebase and settings, so they coincide. For Wan, Base denotes a matched KD-only QAT control that retains the QVGen formulation while adopting the flow-matching training settings below. Base includes neither Denoising-Stage Oriented Supervision nor CFG-drop. These fixed training settings are shared with DSAQuant and are not proposed components of our method.

Flow-shifted timestep sampling. Let σ_i be the unshifted noise level at training index i, and let s be the flow-shift factor, matched to the inference scheduler. We use

$$\widetilde{\sigma}_i = \frac{s\sigma_i}{1 + (s-1)\sigma_i} \tag{68}$$

to construct the noisy training latent, while retaining the standard flow target. This aligns the training noise distribution with the inference scheduler.

Table 12: Supplementary table for the ablation study in Tab. 4, including the additional full-precision CFG-drop control. For FP + CFG-drop, "–" in the VBench columns denotes unreported metrics; its LV result is provided in Table 8.

Method	Mixed loss	CFG drop	#Bits (W/A)	VBench1↑								
				Imaging Quality	Aesthetic Quality	Motion Smoothness	Dynamic Degree	Background Consistency	Subject Consistency	Scene Consistency	Overall Consistency	Avg
Wan 1.3B (CFG $= 5.0, 480p,$ fps $= 16$)												
FP	–	–	16/16	64.22	57.85	97.25	71.11	95.72	93.35	26.07	24.70	66.28
FP + CFG-drop	–	✓	16/16	63.57	57.99	–	70.56	–	–	27.50	24.74	66.44
Base	×	×	4/4	62.86	56.32	96.23	71.67	94.68	91.69	22.94	24.35	65.09
Base + CFG-drop	×	✓	4/4	61.67	56.75	96.57	77.50	96.57	92.57	25.58	24.20	66.31
Base + Mixed loss	✓	×	4/4	64.06	57.47	96.07	76.67	94.45	91.52	25.16	24.55	66.24
Ours	✓	✓	4/4	63.52	58.01	96.67	80.00	95.52	92.96	26.54	24.49	67.20

Table 13: Supplementary table for ablation study in Tab. 5.

Method	#Bits (W/A)	VBench1↑								
		Imaging Quality	Aesthetic Quality	Motion Smoothness	Dynamic Degree	Background Consistency	Subject Consistency	Scene Consistency	Overall Consistency	Avg
Wan 1.3B (CFG $= 5.0, 480p,$ fps $= 16$)										
FP	16/16	64.22	57.85	97.25	71.11	95.72	93.35	26.07	24.70	66.28
linear	4/4	62.8	57.1	97.80	70.55	95.88	93.73	20.22	23.84	65.24
exp1	4/4	63.42	57.03	97.52	73.61	95.71	93.79	22.60	23.85	65.94
exp5	4/4	63.53	58.01	96.67	80.00	95.52	92.96	26.54	24.49	67.20
exp10	4/4	63.50	57.71	97.70	74.72	95.77	93.49	18.79	23.43	65.64

Bell-shaped mean-normalized loss weighting. Following the public DiffSynth-Studio training recipe [4], for N discrete training timesteps we define

$$u_i = \exp\left[-2\left(\frac{i-(N-1)/2}{N}\right)^2\right], \qquad w_i = \frac{N\,(u_i - u_{\min})}{\sum_{j=0}^{N-1}(u_j - u_{\min})}, \quad u_{\min} = \min_j u_j. \tag{69}$$

The weights have unit mean over the discrete timestep grid. The Wan training objective is

$$\mathcal{L}_{\text{Wan}} = \mathbb{E}_{i,x_1,c,\epsilon}\left[w_i\,\mathcal{L}(\widetilde{t}_i)\right], \tag{70}$$

where $\widetilde{t}_i$ corresponds to the shifted noise level, expressed using the normalized-time convention of Sec. 3.1. Base uses $\mathcal{L} = \mathcal{L}_{\text{KD}}$, whereas DSAQuant uses $\mathcal{L} = \mathcal{L}_{\text{mix}}$. Thus, the shared weighting w_i is distinct from our KD-to-target supervision schedule.

QVGen evaluation and reproduction. We evaluate released QVGen checkpoints for settings where they are available, including Wan2.1-1.3B and CogVideoX-2B W4A4. For the remaining settings, we follow its released code and match the training data, iterations, quantizers, prompts, resolutions, and evaluation protocol to our experiments. Our reproduced implementation also receives the same LUT precomputation, text-embedding exclusion from quantization, and multi-GPU training-data sampling optimizations. For reproduced settings, QVGen was run multiple times and its best result is reported. The symmetric W4A4 and asymmetric W3A3 settings are distinguished explicitly in the main text; published asymmetric W4A4 results are separately marked.

J Supervision Objectives and Preservation of FP Outputs

We compare target-only QAT, KD-only QAT, and DSAQuant in the Wan2.1-1.3B W4A4 ablation setting to assess whether supervision changes preserve the behavior of the original FP model. All three quantized models use the same CFG-drop setting, isolating the training-objective comparison. We report LPIPS [37] and FVD [28] between quantized-model and FP-model outputs; lower values indicate closer agreement with FP, rather than necessarily better generation quality in isolation.

Target-only training causes a substantially larger shift from FP outputs. DSAQuant remains close to KD-only QAT on these consistency measures while improving generation quality through stage-aligned supervision, as shown by the component ablation in Table 4. These results support retaining

Table 14: **Effect of the supervision objective on FP-output preservation.** CFG-drop is applied to all three methods. LPIPS and FVD are measured relative to FP outputs.

Method	LPIPS vs. FP↓	FVD vs. FP↓
Target Only + CFG-drop	0.63	1623
KD Only + CFG-drop	**0.43**	**962**
DSAQuant	0.44	975

teacher supervision in early denoising stages rather than replacing KD with the native target at every step.

K Robustness to Sampling Configurations

K.1 Sampling steps, sampler, and guidance scale

We evaluate Wan2.1-1.3B W4A4 under 20-step inference using FlowMatch and FlowMatch-UniPC, including CFG scales of 3.5, 5, and 7. The CFG-drop window is 15% of the sampling steps in these fast-inference experiments. Table 15 shows improvements from the mixed supervision objective in every setting, with further gains when CFG-drop is included.

Table 15: **Sampling robustness on Wan2.1-1.3B W4A4.** All settings use 20 denoising steps. Entries are VBench averages. FM denotes FlowMatch, and FM-UniPC denotes FlowMatch-UniPC.

Method	FM CFG=5	FM-UniPC CFG=5	FM-UniPC CFG=3.5	FM-UniPC CFG=7
Base	62.86	62.80	63.90	59.80
Base + Mixed loss	63.77	63.79	65.10	60.70
DSAQuant	**65.01**	**64.84**	**65.90**	**64.90**

K.2 CFG-drop windows on a second backbone

We additionally evaluate Wan2.2-5B at 720p with a 50-step FlowMatch sampler and flow shift 5. Scheduler-timestep thresholds of 400, 500, and 600 correspond to dropping CFG in the final 6, 9, and 12 sampling steps, respectively. All three windows outperform the no-drop setting in Table 16.

Table 16: **CFG-drop window sensitivity on Wan2.2-5B at 720p.** The sampler uses 50 FlowMatch steps and flow shift 5. A threshold of zero disables CFG-drop.

Scheduler threshold	Dropped steps	Drop ratio	VBench Avg↑
0	0	0%	64.64
400	6	12%	**65.71**
500	9	18%	65.53
600	12	24%	65.55

These experiments motivate specifying CFG-drop as a late-stage window rather than treating nine steps as a universal constant. The normalized progress t in Sec. 3.1 runs from noise to data, whereas the scheduler timestep decreases during sampling; neither should be confused with the discrete denoising-step index. The mapping from a scheduler threshold to a number of dropped steps depends on the sampler and flow shift, as made explicit above.

L Transfer to NVFP4 and Few-Step Video Generation

L.1 NVFP4 weight and activation quantization

We test both weight and activation quantization in the NVFP4 format on Wan2.1-1.3B. Table 17 reports augmented-prompt VBench averages for the KD-only Base, Base with CFG-drop, and the

complete stage-aligned design. CFG-drop improves the score by 0.8 points, and adding mixed supervision yields a total gain of 1.7 points over Base. This result supports transfer of the two stage-aware components beyond the INT4 setting.

Table 17: **Transfer to NVFP4 on Wan2.1-1.3B.** Both weights and activations use NVFP4. Scores are augmented-prompt VBench averages.

Method	Augmented VBench Avg↑
Base	68.3
Base + CFG-drop	69.1
DSAQuant	**70.0**

L.2 Four-step rCM model

We further apply DSAQuant to the rCM [40] four-step Wan2.1-1.3B model under W4A8 quantization. QAT is performed directly on its native four-step denoising trajectory, preserving the sampling distribution of the step-distilled model rather than reverting to a 50-step scheduler. Table 18 shows a 1.4-point VBench improvement over Base. This demonstrates compatibility with step-distilled video generation in the tested W4A8 setting.

Table 18: **Transfer to a four-step rCM Wan2.1-1.3B model.** Both methods use W4A8 quantization and the native four-step trajectory.

Method	W/A	Sampling steps	VBench Avg↑
Base	4/8	4	64.8
DSAQuant	4/8	4	**66.2**

M Further inference acceleration

We further integrate SageAttention to accelerate inference. The results are summarized in Tab. 19.

Table 19: Inference latency when combining W4A4 quantization with SageAttention. All results are measured under 81 frames at 480p.

Model	Methods	4090D latency(s)
Wan 2.1-1.3B	Ours	193
	Ours + SageAttention	140
Wan 2.1-14B	Ours	930
	Ours + SageAttention	698
Wan 2.2-5B	Ours	58
	Ours + SageAttention	53

N Visual Comparison

We provide representative qualitative comparisons across three Wan family backbones: Wan2.1-1.3B, Wan2.1-14B, and Wan2.2-5B. All samples are generated with the same prompts and inference settings as the quantitative evaluation, allowing a direct comparison between DSAQuant and prior QAT baselines. These examples complement the numerical results by highlighting differences in text-video alignment, texture fidelity, local detail preservation, and quantization-induced artifacts. Specifically, Figs. 5 and 6 show W4A4 and W3A3 results on Wan2.1-1.3B, Figs. 7 and 8 show results on Wan2.2-5B, and Figs. 9 and 10 show results on Wan2.1-14B. Full video comparisons are provided in the supplementary material.

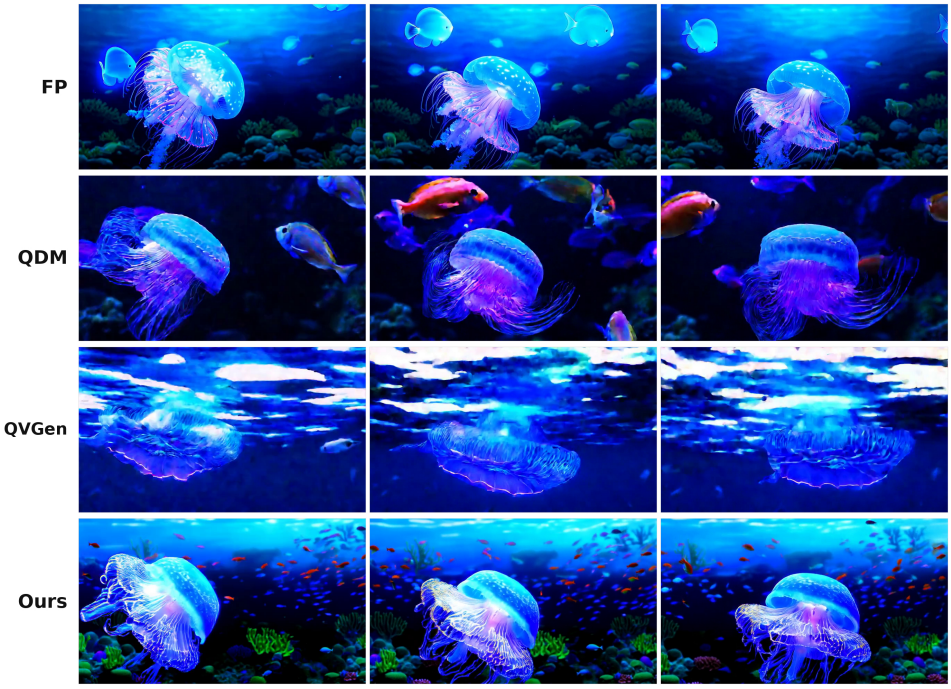

Figure 5: 4bit quantization on Wan2.1 1.3b model with prompt:"CG animation digital art, a majestic jellyfish floating gracefully through the oceanic depths. The jellyfish has intricate patterns on its translucent body, with vibrant hues of blue, green, and purple. Its bioluminescent tentacles emit a soft, mesmerizing glow, creating an ethereal underwater landscape. The tentacles sway gently as the jellyfish glides, illuminating the surrounding water with a captivating light show. The ocean background is filled with schools of colorful fish and drifting coral reefs. The jellyfish is surrounded by a serene, tranquil atmosphere. Soft, ambient ocean sounds play in the background. Low-angle, slow-motion shot focusing on the jellyfish and its glowing tentacles."

O Limitations, Broader impacts

Limitations. Our experiments are currently conducted on bidirectional video diffusion models. Recent advances in autoregressive video generation show strong potential for real-time generation, and evaluating DSAQuant on such models is an important direction for future work.

Broader impacts. This work improves the efficiency of video diffusion models through quantization-aware training. By reducing memory and computational costs while preserving visual fidelity, DSAQuant can make high-quality video generation more accessible and practical on resource-constrained hardware. This may benefit creative production, education, assistive applications, and interactive media. However, efficient video generation also raises potential risks. Lowering the deployment cost of high-fidelity VDMs may make it easier to generate misleading synthetic videos, impersonation content, privacy-violating media, or other harmful outputs. Efficiency gains may also increase overall usage, partially offsetting the environmental benefits of reduced per-sample computation. We therefore encourage the deployment of such models together with safeguards such as watermarking, provenance tracking, content filtering, safety classifiers, and clear disclosure of synthetic content.

P LLM Usage

In this paper, Large Language Models (LLMs) were used to assist with polishing the text and formatting the tables.



Figure 6: 3bit quantization on Wan2.1 1.3b model with prompt:"CG animation digital art, an African elephant taking a peaceful walk through a lush green forest at dawn. The elephant has a grayish-brown coat with soft folds and wrinkles, and gentle eyes. It walks slowly, with its trunk raised slightly, sniffing the air. The forest is filled with tall trees and vibrant greenery, with colorful flowers blooming. Birds fly overhead, chirping melodiously. The sky is a beautiful orange hue as the sun rises. The elephant moves gracefully, with each step careful and deliberate. The background features intricate foliage and subtle shadows. Soft lighting creates a warm and serene atmosphere. Low-angle shot from behind, focusing on the elephant's tranquil expression and movement."

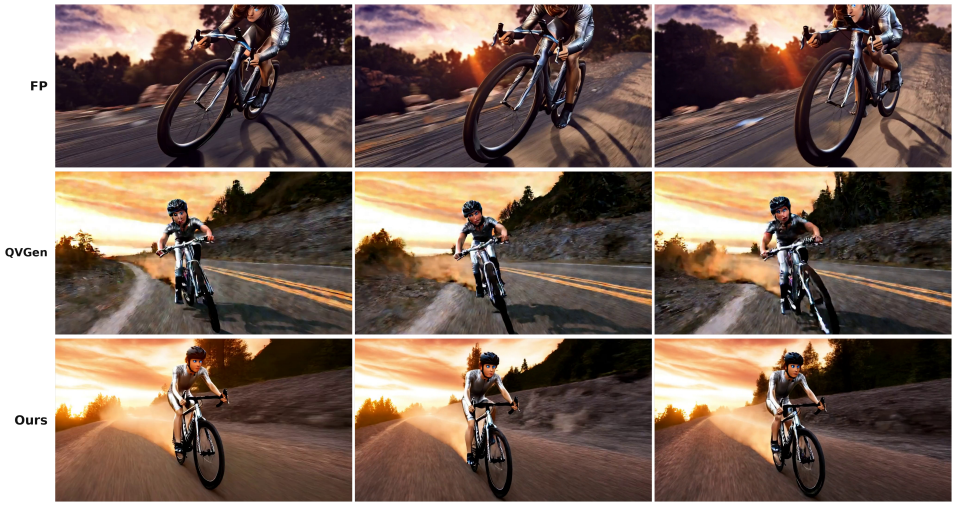

Figure 7: 4bit quantization on Wan2.2 5b model with prompt:"CG animation digital art, a sleek racing bicycle speeding down a winding mountain road. The bike is a deep metallic silver color with intricate patterns etched along its frame. It accelerates rapidly, the rider gripping the handlebars tightly with focused determination. The rider has short, wavy brown hair and piercing blue eyes, wearing a tight-fitting black helmet and a snugly fitting silver jersey. They lean forward slightly, their arms pumping hard to maintain momentum. The road is rugged, covered in loose gravel and rocks, with trees lining the sides. The sun sets behind them, casting dramatic shadows. The background features a sunset sky with wispy clouds and hints of orange and pink hues. The bike's wheels spin furiously as it gains speed, creating a whirlwind of dust and debris. The scene captures the adrenaline rush of a thrilling bicycle race. Close-up, low-angle view."



Figure 8: 3bit quantization on Wan2.2 5b model with prompt:"A playful feline sprinting joyfully across a lush green meadow dotted with wildflowers. The cat has sleek fur, expressive green eyes, and a fluffy tail that wags excitedly as it bounds forward. The meadow stretches out behind it, with vibrant sunflowers and buttercups swaying gently in the breeze. The sky above is a bright azure, filled with fluffy white clouds. The cat's joyful run is captured from a dynamic low-angle perspective, showcasing its agility and boundless energy. The scene is bathed in warm golden light, enhancing the cat's lively demeanor. Grass and petals trail behind the cat as it dashes towards the horizon. The background features a serene rural landscape, with small cottages and winding country roads visible in the distance. The overall composition is energetic and full of life, perfectly capturing the essence of a cat running happily."



Figure 9: 4bit quantization on Wan2.1 14b model with prompt:"CG animation concept art, a young woman with natural beauty applying makeup in the morning. She is wearing a simple white blouse and black pencil skirt. Her hair is styled in loose waves, framing her face. She is sitting at a vanity table with soft lighting illuminating her work. She is using a compact mirror to apply foundation, concealer, and blush. Her fingers move gracefully as she blends and applies products. She pauses occasionally to check her reflection and adjust her technique. The background features a minimalist room with a few scattered items and a vintage vanity set. Soft, gentle brush strokes and subtle motions. Low-angle shot from above, focusing on her hands and facial expressions."



Figure 10: 3bit quantization on Wan2.1 14b model with prompt:"A thrilling motorcycle speeding down a winding mountain road at night. The sleek black motorcycle with bright red accents accelerates rapidly, leaving behind a trail of dust. The rider, a muscular man with short cropped hair, wears a black leather jacket and jeans. His helmet is off, revealing a determined and focused expression. He grips the handlebars tightly, leaning slightly into the turn. The motorcycle's engine roars as it gains momentum, reflecting the intense speed and power. The background is a dimly lit mountain landscape, with flickering streetlights and twinkling stars. The scene captures the adrenaline rush and raw energy of the motorcycle's acceleration. Nighttime low-angle shot from the side."

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: The abstract and introduction describe the two stage-aligned components of DSAQuant and their evaluation on Wan and CogVideoX models under W4A4 and W3A3 quantization. The reported quality improvements and efficiency claims are supported by the main results, ablations, and efficiency comparisons.

   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: Appendix O discusses the restriction of our experiments to bidirectional video diffusion models and identifies evaluation on autoregressive models as future work.

   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: Appendices A and B state the conditioning assumptions and derive the gradient-variance and CFG quantization-error decompositions. Appendix F explains the conditions under which timestep-dependent outputs can be precomputed and provides the associated storage analysis.

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: The main text specifies the training objective, guidance schedule, quantizers, and benchmark comparisons, while Appendices D and E describe the model-specific training data preparation, Adam optimizer with zero weight decay, learning rates, batch sizes, training and warmup steps, and evaluation protocol. Public inference code and W4A4 checkpoints additionally support verification of the released Wan models; the scope of the code and data release is stated separately below.

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 provide public W4A4 inference code and instructions for the Wan models at `https://github.com/robbyant-research/DSAQuant`, with quantized checkpoints at `https://huggingface.co/Robbyant-Research/DSAQuant`.

   Guidelines:

   - 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: The main text and Appendices D and E report the model and quantizer settings, data preparation, Adam optimizer with zero weight decay, model-specific learning rates, batch sizes, training and warmup steps, resolutions, and evaluation protocol. The main-text ablations compare the supervision mixing schedules and CFG-drop thresholds used to select these method hyperparameters.

   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: [No]

Justification: The paper reports aggregate benchmark scores but does not report error bars, confidence intervals, or statistical significance tests. The rank-based random-seed protocol in Appendix E specifies how videos are sampled; it does not quantify uncertainty in the reported results.

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: Appendix D and Table 9 report H20 GPUs with 140GB memory, GPU counts through the global batch size, and per-model training GPU-days; Appendix E reports evaluation GPU counts, and Table 7 reports inference hardware, memory use, and latency. A separate project-wide compute total including preliminary or failed runs is not reported.

   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 follow 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: Appendix O discusses benefits from lower deployment costs and risks including misleading synthetic videos, impersonation, privacy violations, and increased overall usage, together with potential mitigation strategies.

    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: [No]

    Justification: Appendix O discusses misuse risks and recommends watermarking, provenance tracking, content filtering, and safety classifiers. These are deployment recommendations; the paper does not document such safeguards as implemented in the released quantized video-generation models.

    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: The main text and Appendix D cite the model and data sources, including Wan, CogVideoX, VidProM, and TurboDiffusion.

    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:[Yes]

    Justification: We release inference code and W4A4 Wan checkpoints through the public repository and model page linked above. The repository README documents installation, checkpoint loading, inference commands, and quantization details.

    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: The paper does not involve crowdsourcing or 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: The paper does not involve research with human subjects requiring IRB approval or an equivalent review.

    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: DSAQuant does not use an LLM as an important, original, or non-standard component of its core method. LLM assistance with manuscript polishing and table formatting is disclosed in Appendix P.

    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.