FPSAttention: Training-Aware FP8 and Sparsity Co-Design for Fast Video Diffusion

Akide Liu[*1,2,3] **Zeyu Zhang**[*2] **Zhexin Li**[†2,4] **Xuehai Bai**[†3] **Yuanjie Xing**[†2] **Yizeng Han**[2]
Jiasheng Tang[‡2,4] **Jichao Wu**[2,4] **Mingyang Yang**[2,4] **Weihua Chen**[2] **Jiahao He**[3]
Yuanyu He[3] **Fan Wang**[2] **Gholamreza Haffari**[1] **Bohan Zhuang**[‡2,3]

[1]Monash University [2]DAMO Academy, Alibaba Group [3]ZIP Lab, Zhejiang University [4]Hupan Lab
[*]Co-first authors. [†]Co-second authors. [‡]Project leads.
Corresponding authors: fan.w@alibaba-inc.com, bohan.zhuang@gmail.com

Abstract

Diffusion generative models have become the standard for producing high-quality, coherent video content, yet their slow inference speeds and high computational demands hinder practical deployment. Although both quantization and sparsity can independently accelerate inference while maintaining generation quality, naively combining these techniques in existing training-free approaches leads to significant performance degradation, as they fail to achieve proper joint optimization. We introduce FPSAttention, a novel training-aware co-design of FP8 quantization and Sparsity for video generation, with a focus on the 3D bi-directional attention mechanism. Our approach features three key innovations: 1) A unified 3D tile-wise granularity that simultaneously supports both quantization and sparsity. 2) A denoising step-aware strategy that adapts to the noise schedule, addressing the strong correlation between quantization/sparsity errors and denoising steps. 3) A native, hardware-friendly kernel that leverages FlashAttention and is implemented with optimized Hopper architecture features, enabling highly efficient execution. Trained on Wan2.1's 1.3B and 14B models and evaluated on the VBench benchmark, FPSAttention achieves a 7.09× kernel speedup for attention operations and a 4.96× end-to-end speedup for video generation compared to the BF16 baseline at 720p resolution—without sacrificing generation quality. Project page: https://fps.ziplab.co.

1 Introduction

Diffusion models have revolutionized AI through breakthrough image synthesis [39, 6, 34] and are advancing into complex video generation [40, 45]. Diffusion Transformers (DiTs) [28] now enable efficient, high-quality synthesis [1, 23], powering billion-parameter models like Wan2.1 [37] that produce coherent, long-duration, high-fidelity videos.

Despite progress, crippling computational demands persist [32]: (i) iterative reverse-diffusion requiring hundreds of steps, and (ii) quadratic-complexity spatio-temporal attention ($\mathcal{O}(N^2)$, N denotes the number of tokens) [22, 17, 37]. Particularly, the computational burden of attention becomes prohibitive for high-resolution, long-duration videos, often consuming >70% of inference time [51, 43,

Method	Kernel Speedup	E2E Speedup
BF16	1.00×	1.00×
FP8	1.84×	1.26×
STA	5.15×	3.60×
FPSAtten	**7.09**×	**4.96**×

Table 1: Efficiency comparison of the BF16 baseline, FP8 quantization, STA sparse attention, and our FPSAttention method on Wan2.1-14B at 720p resolution on an NVIDIA H20 GPU. We report both kernel-level and end-to-end speedups relative to the BF16 baseline.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

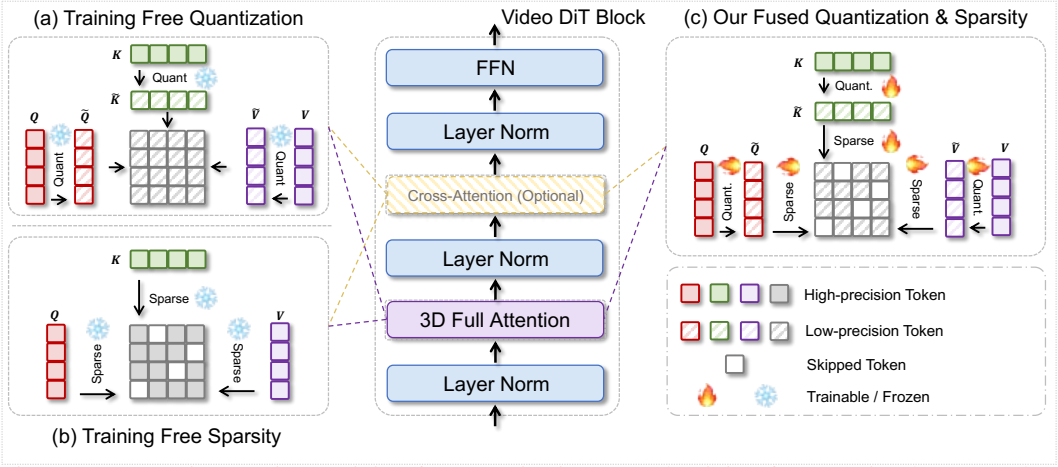


Figure 1: Comparing previous training-free quantization (a) and training-free sparsity (b) approaches reveals substantial accuracy degradation and a lack of compatibility when used independently. In contrast, our `FPSAttention` framework (c) integrates low-precision and sparse patterns in a single training process, yielding near-zero accuracy loss and seamless deployment.

35]. For instance, Wan2.1-14B requires approximately 2.5 hours on an NVIDIA H20 GPU to generate a 5s video.

To address the efficiency challenge, numerous acceleration methodologies have been proposed [7, 24, 33], among which quantization and sparsity have emerged as predominant techniques [38, 25]. Quantization reduces numerical precision (e.g., FP32→INT8/FP8), thereby decreasing the memory footprint and enabling faster computations (Figure 1 (a)). The recent post-training quantization (PTQ) method SageAttention [49] quantizes attention modules into INT8 with calibration strategies, providing moderate acceleration with downgraded generation quality. Compared to INT8, the emerging format FP8 quantization offers a wider dynamic range[26], facilitating both training and inference [18, 31]. Nevertheless, training-free FP8 quantization, despite its theoretical advantages, introduces significant quantization errors that degrade model performance [49]. Apart from quantization, sparsity techniques address the quadratic computational complexity of 3D full attention by selectively skipping computations, as illustrated in Figure 1 (b). Representatively, Sparse-VideoGen [43] implements per-head spatial-temporal masks aligned with GPU blocks, SpargeAttn [50] employs a two-stage filtering mechanism, and Sliding Tile Attention (STA) [51] leverages local 3D sliding windows with kernel-level optimizations.

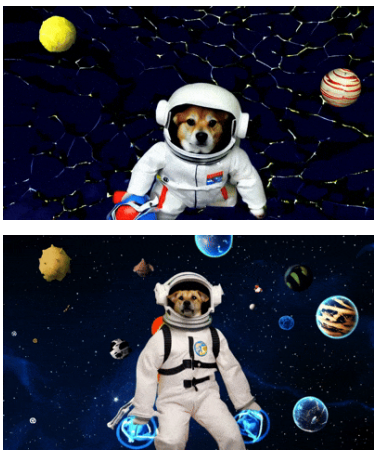


Figure 2: Comparison of video generation results. Top: Training Free FP8 + STA. Bottom: Our Training-Aware `FPSAttention`. Click the image to play the video via Acrobat Reader.

To enjoy the benefits of both worlds, a straightforward strategy is jointly applying FP8 quantization and sparsity. However, a naive combination presents significant challenges, as quantization errors can be magnified when combined with sparsity mechanisms [44], as shown in Figure 2. Intuitively, sparsity techniques prioritize token retention with high-magnitude attention scores, while quantization disproportionately introduces errors in these high-magnitude values. This intrinsic tension necessitates holistic approaches that consider both techniques simultaneously, potentially framing sparsification as a specialized form of 0-bit quantization to achieve optimal balance between efficiency and generation quality [27].

Furthermore, existing approaches largely overlook training-aware joint optimization of quantization and sparsity, creating a substantial training-inference gap. Our empirical analysis (Section 3.3) reveals that diffusion models can tolerate and even correct for hardware-friendly tile-wise errors. The error resilience is particularly pronounced when the model is aware of approximations via quantization-aware training (QAT), ensuring consistent performance at inference time.

In this paper, we introduce `FPSAttention`, as shown in Figure 1(c), a novel training-aware co-design framework that synergistically integrates FP8 quantization and structured sparsity for 3D attention in video DiTs.

`FPSAttention` proposes three key innovations:

- *Unifying Tile-wise Operations*: Implementing a 3D tile-wise granularity for both FP8 quantization and block sparsity (section 3.2), which directly aligns with efficient hardware execution patterns (e.g., GPU Tensor Cores) and forms the basis for structured acceleration.
- *Denoising Step-Aware Scheduling*: Introducing an adaptive strategy (section 3.3) that dynamically adjusts quantization and sparsity granularity according to the varying error sensitivity and corrective capacity of the model across different denoising timesteps.
- *Hardware-Optimized Kernel Design*: Developing a native, high-performance kernel (section 3.4) leveraging features like FlashAttention and NVIDIA Hopper architecture optimizations to translate theoretical FLOPs reduction into tangible wall-clock speedups.

As demonstrated in Table 1, by training on Wan2.1's 1.3B and 14B models and evaluating on the vBench benchmark, FPSAttention achieves a **7.09**$\times$ kernel speedup for attention operations and a **4.96**$\times$ end-to-end speedup for video generation compared to the BF16 baseline, all without sacrificing generation quality, significantly outperforming approaches that apply quantization (1.84$\times$ kernel speedup) or sparsity (5.15$\times$ kernel speedup) independently. Our work not only provides a practical solution for accelerating video diffusion but also offers a new perspective on the robustness of diffusion models to aggressive, structured compression.

2 Related Work

Quantization for video generation models. The computational expense of video generation models, particularly Diffusion Transformers (DiTs) [17], driven by iterative sampling [11] and quadratic attention complexity [35], necessitates model quantization techniques [32]. Post-Training Quantization (PTQ) has been explored for its efficiency [10, 20, 13, 54, 15]; however, applying PTQ to video DiTs presents unique challenges beyond standard image models [42, 3, 53]. Temporal variability of activation statistics across denoising steps [14] has prompted PTQ methodologies to implement time-step-wise calibration [46], adaptive quantization, and dynamic smoothing techniques [30]. Recent work has evaluated these techniques on standardized benchmarks (e.g., VBench [16]), assessing temporal consistency alongside perceptual quality. While PTQ approaches show promising results, Quantization-Aware Training (QAT) for video diffusion models remains largely unexplored. Our work addresses this gap by introducing an FP8 QAT framework that jointly optimizes quantization and sparsity, enabling efficient video generation while maintaining visual fidelity.

Sparse attention for video generation models. Recent advancements in sparse video generation and efficient attention mechanisms have improved memory utilization and computational efficiency. Sparse VideoGen [43] leverages sparsely sampled motion priors to produce realistic videos while reducing temporal redundancy. Efficient attention mechanisms have proven crucial for handling long-range dependencies in video data. Sliding Tile Attention[51] introduces a tiled sparse attention mechanism for modeling spatial-temporal correlations, while SpargeAttn [50] proposes progressive sparsification by selectively pruning attention tokens based on importance scores. DiTFastAttn [2] accelerates attention computation by dynamically filtering irrelevant patches, achieving significant speedups without compromising quality. These approaches illustrate the trend of combining structured sparsity with content-aware selection for scalable video generation. However, these methods are typically limited to inference-time acceleration, lack integration with model training procedures, and are not fully compatible with quantization techniques, creating challenges for developing holistically efficient video generation frameworks.

3 Method

Our technique integrates algorithmic innovation with hardware-conscious kernel optimization to enhance the efficiency of video DiTs. This section begins by establishing fundamental concepts essential to our methodology. Subsequently, we introduce the architecture of our proposed `FPSAttention`,

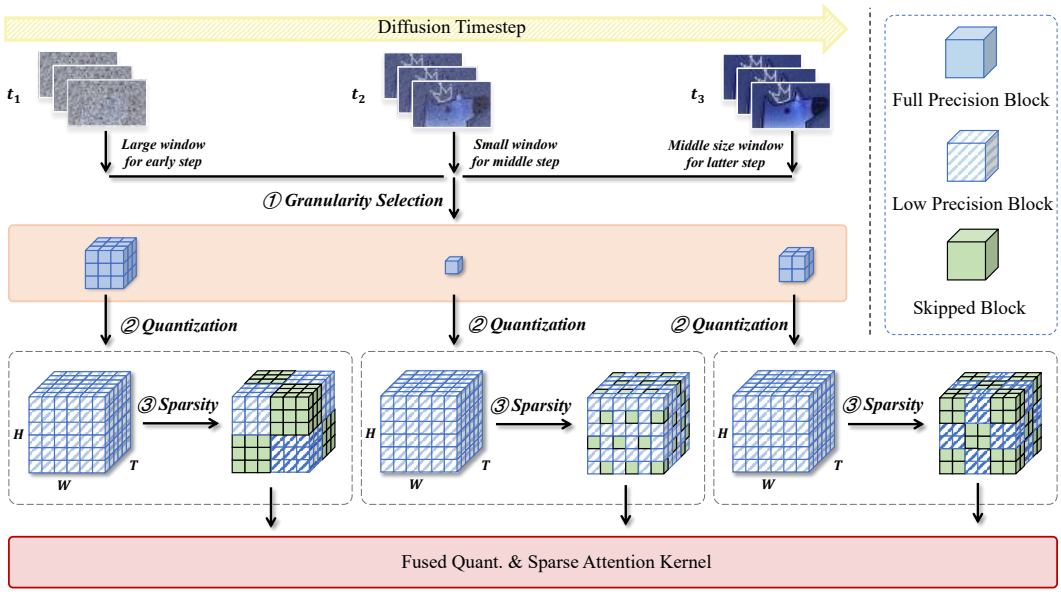


Figure 3: Overview of `FPSAttention`. (1) Our approach synergistically optimizes joint quantization and sparsity patterns within the attention mechanism for efficient video generation. (2) We introduce a novel denoising step-aware strategy that dynamically adapts the granularity throughout the diffusion process, balancing computational efficiency and perceptual fidelity. Empirical observations are shown in Figure 5. (3) A fused hardware-friendly kernel is applied for attention operations.

detailing its two primary algorithmic contributions: a unified tile-wise quantization and sparse attention mechanism, and a denoising step-aware strategy for dynamic adaptation of quantization and sparsity hyperparameters. Finally, we outline our hardware-optimized kernel implementation that plays a crucial role in translating theoretical computational savings into practical efficiency gains. Figure 3 provides a high-level conceptual overview of our `FPSAttention` framework.

3.1 Background

This subsection establishes the two foundational techniques that underpin our methodology: 8-bit floating-point (FP8) quantization and Sliding Tile Attention (STA).

FP8 quantization. Video Diffusion Transformers (DiTs) process $L = T \times H \times W$ spatiotemporal tokens, where T, H, and W represent temporal frames, height, and width dimensions. To reduce memory bandwidth requirements for activation tensors, FP8 quantization approximates each value $X_{i,j}$ using an 8-bit floating-point representation. Unlike INT8 quantization, which maps continuous values to a scaled integer grid, FP8 conversion preserves the floating-point nature by utilizing dedicated sign, exponent, and mantissa bits (in formats such as E4M3 or E5M2).

The FP8 conversion employs a scaling factor s_g for each tile of values g to map the original values into the representable dynamic range of FP8:

$$\hat{X}_{\text{FP8}}(X_{i,j}; s_g) = \text{dequantize}(\text{FP8\_convert}(X_{i,j} \cdot s_g))/s_g. \tag{1}$$

To enhance approximation accuracy, tile-wise FP8 quantization employs per-tile scaling factors $\{s_g\}$ that minimize quantization error within each specific tile. This approach preserves attention head-specific and frame-specific activation dynamics while typically reducing data size by half (e.g., from 16-bit to 8-bit). The result is a theoretical $2\times$ reduction in memory bandwidth requirements, with further effective improvements achievable through specialized FP8 hardware acceleration.

Sliding Tile Attention (STA). Standard attention operations on $N = L$ tokens with feature dimension d incur a computational complexity of $\mathcal{O}(N^2 d)$, creating a significant bottleneck for high-resolution video generation. STA addresses this challenge by partitioning the 3D token space into M non-overlapping tiles $\{\mathcal{T}_u\}$ of dimensions (T_t, T_h, T_w).

The key innovation of STA is its locality-based attention mechanism: each query tile u attends exclusively to key tiles v within a local neighborhood $\mathcal{W}(u)$, defined by the distance constraint:

$$\mathcal{W}(u) = \left\{ v : \|c_u - c_v\|_\infty \leq \left(\frac{W_t}{2T_t}, \frac{W_h}{2T_h}, \frac{W_w}{2T_w} \right) \right\}, \tag{2}$$

where (W_t, W_h, W_w) denote the window dimensions measured in tile units, and c_u, c_v are the centers of tiles u and v. This constraint effectively replaces full attention with a tile-wise masked attention:

$$P_{q,k} = \begin{cases} \text{Softmax}(Q_q K_k^\top / \sqrt{d}), & \text{if token } k \text{ is in a tile } \mathcal{T}_v \text{ where } v \in \mathcal{W}(u), \\ -\infty, & \text{otherwise.} \end{cases} \tag{3}$$

where Q and K are queries and keys. STA generates $M \times |\mathcal{W}(u)|$ dense attention blocks that are compatible with optimized implementations such as FlashAttention[5]. This design provides substantial speedup by replacing the irregular sparse patterns of token-wise sliding window attention with structured tile-based computations that align well with GPU memory hierarchies. As demonstrated in [51], this approach can accelerate attention by 2.8–17× over FlashAttention-2[4] and 1.6–10× over FlashAttention-3[29] for video generation tasks.

3.2 Joint Tile-wise FP8 Sparse Attention

Building upon FP8 quantization and tiled attention techniques, we introduce `FPSAttention`, a *Joint Tile-wise FP8 Quantization and Sparse Attention* mechanism that synergistically optimizes computational efficiency and perceptual accuracy in video DiTs.

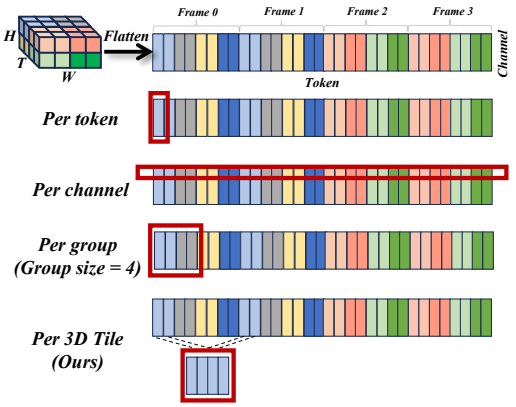


Figure 4: Quantization granularities: per-token, per-channel, per-group, and our per 3D-tile, which aligns with hardware compute patterns.

Our tile-wise granularity approach (Figure 4, last row) is motivated by three primary considerations. First, it offers an optimal accuracy-efficiency trade-off compared to conventional methods (per-token, per-channel, per-group; Figure 4, first three rows) that often fail to align with underlying hardware architectures. While per-group quantization provides a reasonable balance, it frequently overlooks GPU compute tile patterns, thereby reducing hardware utilization efficiency. Second, our approach maintains full compatibility with the STA sparsity design, allowing seamless integration of quantization and sparsity optimizations at matching granularity. Third, our tile-wise design exhibits superior hardware compatibility, aligning precisely with compute tiles in optimized kernels such as FlashAttention, which enables direct translation of theoretical computational savings into practical speedups.

The `FPSAttention` mechanism processes neural activations through a systematic workflow: (1) organizing query (Q) and key (K) activations into contiguous 3D tiles aligned with GPU cache layouts for enhanced data locality; (2) quantizing each tile to FP8 precision with a locally optimized scale factor; (3) enforcing tile-granularity sparse attention patterns, leveraging spatial locality and low-bit arithmetic; and (4) dequantizing the aggregated attention output to higher precision (BF16/FP16).

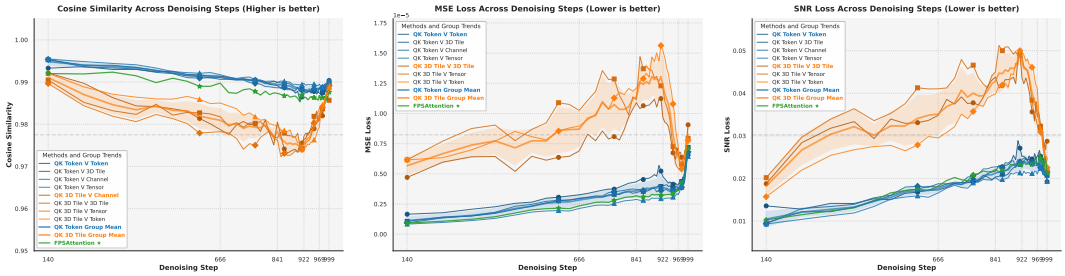
Tile-wise FP8 quantization for Q and K. The matrices $Q, K \in \mathbb{R}^{L \times d}$ are partitioned along the sequence dimension L into non-overlapping tiles $\{\mathcal{T}_u\}$ of dimensions (T_t, T_h, T_w). For each tile $\mathcal{T}_u$, we compute separate scaling factors s_u^Q and s_u^K to map their values optimally to the FP8 representable range via

$$s_u^Q = \max_{(i,j) \in \mathcal{T}_u} |Q_{i,j}| / M_{\text{FP8\_max}}, \quad s_u^K = \max_{(i,j) \in \mathcal{T}_u} |K_{i,j}| / M_{\text{FP8\_max}}, \tag{4}$$

where $M_{\text{FP8\_max}}$ is the maximum representable magnitude in FP8 and bounded by specific format. Then each element is independently quantized:

$$\hat{Q}_{i,j} = \text{FP8}(Q_{i,j}; s_u^Q), \quad \hat{K}_{i,j} = \text{FP8}(K_{i,j}; s_u^K). \tag{5}$$

This per-tile scaling strategy minimizes quantization error for both Q and K independently, preserving attention dynamics more effectively than global scaling approaches. When combined with the STA



(a) Cosine similarity. (b) MSE loss. (c) SNR loss.

Figure 5: Joint quantization and sparsity error patterns across denoising steps. Blue: token-level granularity; orange: our 3D tile-wise granularity with sparse attention. Key insight: early/late steps tolerate coarser quantization and higher sparsity, while intermediate steps require finer granularity and denser attention. Our `FPSAttention` (green) closely approximates highest-granularity methods, validating our adaptive scheduling strategy. All measurements from inference are with identical prompts. Performance gaps primarily stem from FP8 quantization rather than sparsity constraints.

formulation, the attention weights P are computed using the quantized $\hat{Q}$ and $\hat{K}$, following Eq. 3. This formulation enforces a regular, block-sparse pattern (Figure 3) that efficiently maps to modern GPU compute architectures.

Channel-wise FP8 quantization of V and tensor-wise FP8 quantization of P. For the value matrix $V \in \mathbb{R}^{L\times d}$, a channel-wise FP8 quantization approach is employed. For each channel, we compute the scaling factor s_j^V to map the values optimally to the FP8 representable range: $s_j^V = \max_{i\in L} |V_{i,j}|/M_{\text{FP8\_max}}$. Each element in V_j is subsequently quantized as $\widehat{V}_j = \text{FP8}(V_j; s_j^V)$. We observe that keeping the fine granularity for V is critical for the performance. Following SageAttention2 [48], we use a fixed scalar $\frac{1}{448}$ to quantize P, obtaining $\widehat{P}$ in FP8.

Aggregation and dequantization. The attention output is computed as $\widehat{X} = \widehat{P}\widehat{V}$. This low-precision output is then dequantized to higher precision (BF16/FP16) to maintain computational stability in subsequent layers.

By quantizing Q and K tiles independently while using channel-wise quantization for V, the `FPSAttention` mechanism captures fine-grained statistical properties of each attention component while maintaining optimal alignment with GPU memory hierarchies. Furthermore, the tile-constrained sparse attention pattern creates $M \times |\mathcal{W}(u)|$ dense, structured attention blocks that avoid the inefficiencies of unstructured sparsity patterns. This approach enables direct wall-clock speedups during execution by maximizing hardware utilization and minimizing memory access overhead, as demonstrated in our experimental results.

3.3 Denoising Step-aware Quantization and Sparsity Strategy

The mechanism described in Section 3.2 employs uniform quantization and sparsity across denoising steps. However, as illustrated in Fig. 5, video DiTs exhibit varying sensitivity to numerical precision and sparsity levels throughout the diffusion process. Specifically, early and late denoising steps demonstrate greater tolerance to coarser quantization and higher sparsity, whereas intermediate steps demand finer numerical precision and lower sparsity. This also suggests that diffusion models can intrinsically correct the approximation errors of attention, which motivates our training-aware scheme to mitigate the training-inference gap. Based on these observations, we propose an adaptive, denoising step-aware compression schedule, for both training and inference. We adjust the quantization granularity $g(t)$ and sparsity window size $W(t)$ based on the denoising step t.

Piecewise schedule for quantization and sparsity. For D denoising steps, we partition the process into three regimes using thresholds $t_1 = \alpha_1 D$ and $t_2 = \alpha_2 D$ $(0 < \alpha_1 < \alpha_2 < 1)$, each associated with different quantization tile sizes $g(t)$ and sparsity window sizes $W(t)$. Smaller $g(t)$ corresponds to a finer quantization granularity, while larger $W(t)$ indicates a denser attention pattern.

We show quantization and sparsity schedule as following. We define a time-dependent hyperparameter vector $S(t) = [g(t), W(t)]$, which is governed by:

$$S(t) = \begin{cases} [g_{\text{coarse}}, W_{\text{sparse}}], & t \leq t_1 \text{ (Early-Denoising Steps)}, \\ [g_{\text{fine}}, W_{\text{dense}}], & t_1 < t \leq t_2 \text{ (Mid-Denoising Steps)}, \\ [g_{\text{intermediate}}, W_{\text{medium\_density}}], & t > t_2 \text{ (Late-Denoising Steps)}, \end{cases} \tag{6}$$

with $g_{\text{coarse}} > g_{\text{intermediate}} > g_{\text{fine}}$ and $W_{\text{dense}} > W_{\text{medium\_density}} > W_{\text{sparse}}$, ensuring the finest quantization granularity and densest attention patterns during these mid-denoising steps, as illustrated in Figure 5.

These hyperparameters are selected at inference time to match the model's varying tolerance to quantization and sparsity across different denoising stages, and then transferred to the training to avoid the prohibitive computational overhead. During training, this configuration allows the model to adaptively compensate for joint quantization-sparsity errors, leading to satisfiable stability and convergence throughout the training process, as shown in Figure 7.

3.4 Hardware-Optimized Kernel Design

Our algorithmic designs are complemented by a hardware-optimized kernel implementation for maximum practical efficiency. The implementation addresses several key aspects: memory access coalescing through structured operations that enable efficient GPU memory loads/stores with tiling support; maximized parallelism via tile-wise operations that process independent tiles concurrently; exploitation of dedicated acceleration units such as Tensor Cores on NVIDIA Hopper/Ada architectures for mixed-precision and FP8 computations; and operation fusion that combines multiple logical steps (attention, sparsity, dequantization) into single triton kernels, significantly reducing overhead and memory traffic while maintaining high tensor core utilization and computational intensity.

4 Experiments

Implementation details. We implemented our proposed framework on the Wan2.1[36] architecture (1.3B and 14B variants), preserving the original model structure while introducing `FPSAttention`, joint FP8 quantization and structured sparsity. The quantization schemes and sparsity patterns were applied across attention mechanisms using score mod and mask mod functions via FlexAttention [8] . Fused kernels were compiled using Triton to accelerate inference on Hopper GPUs. The models were trained on high-quality video data (480p×16fps×5s). We trained `FPSAttention` on Wan2.1-14B using 64 nodes with 8 H20 GPUs for 7 days. For detailed hardware specifications, training procedures, dataset preparation, evaluation protocols, and baseline comparisons, please refer to Appendix.

Evaluation protocol. We evaluate the our method on the public video dataset, VBench [17]. We following the common practice [55, 19] to sample 5 videos per evaluation prompts defined in the VBench dataset, and assess the video generation quality across 16 VBench dimensions. We also report Peak Signal-to-Noise Ratio (PSNR) [12], Structural Similarity Index (SSIM)[41], and Learned Perceptual Image Patch Similarity (LPIPS) [52] metrics.

4.1 Main Results

Quality evaluation. We compare `FPSAttention` with several state-of-the-art optimization methods, as shown in Table 2. Our baselines include sparsity-based approaches (SparseVideoGen [43] and STA), quantization methods (SageAttention [49]), and hybrid approaches (SpargeAtten [50], a training-free method that jointly applies attention sparsification and activation quantization). As demonstrated in Table 2, `FPSAttention` achieves superior performance across all quality metrics. Particularly notable is the average PSNR of 25.74353 on the Wan2.1-14B model, significantly outperforming all baseline methods. This objective metric confirms `FPSAttention`'s ability to generate videos with exceptional fidelity to reference images. Furthermore, `FPSAttention` maintains excellent performance on perceptual metrics, with high Video Quality (0.7103) and strong spatial-temporal consistency (0.9435) on the VBench evaluation. Interestingly, after joint training, `FPSAttention` exhibits a slight increase in VBench scores, while [47, 51] also demonstrate performance improvements when trained with structured sparsity—potentially driven by the inductive bias of locality. These results validate that our joint sparsity and quantization approach preserves visual quality while substantially improving computational efficiency. Visual examples in Figure 6 further illustrate that

Table 2: Quality and efficiency benchmarking results. † We reproduce the results of the baseline methods from the original papers. Note that VBench results here may differ from official results due to the randomness in generated samples and prompt extensions. The quality and efficiency evaluation is based on 480p videos. Specifically, ‡ indicates the speedups via 720p with longer sequence length.

Method	Quality					Efficiency			
	PSNR ↑	SSIM ↑	LPIPS ↓	ImageQual ↑	SubConsist ↑	FLOPS ↓	Latency ↓	Speedup ↑	Speedup‡ ↑
Wan2.1-1.3B†	-	-	-	0.6708	0.9536	77.52 PFLOPS	271s	1.00x	-
SageAttention	20.18990	0.78241	0.18811	0.6699	0.9453	37.61 PFLOPS	141s	1.91x	-
SpargeAtten	17.72979	0.72628	0.26183	0.6541	0.8982	43.15 PFLOPS	205s	1.32x	-
SparseVideoGen	19.51276	0.78891	0.20513	0.6729	0.9292	30.67 PFLOPS	152s	1.78x	-
STA	18.78546	0.76335	0.23187	0.6626	0.8992	31.78 PFLOPS	143s	1.89x	-
Ours Quant	20.99712	0.79820	0.15114	0.6798	0.9458	32.01 PFLOPS	144s	1.88x	-
Ours Quant + Sparse	21.35417	0.80835	0.15398	0.7103	0.9338	32.01 PFLOPS	110s	2.45x	-
Wan2.1-14B†	-	-	-	0.6715	0.9528	637.52 PFLOPS	1301s	1.00x	1.00x
SageAttention	24.33985	0.82283	0.15607	0.6724	0.9530	301.98 PFLOPS	646s	2.01x	1.94x
SpargeAtten	21.38291	0.81452	0.21723	0.6350	0.9173	339.30 PFLOPS	734s	1.77x	2.12x
SparseVideoGen	23.52881	0.80113	0.17032	0.6868	0.9489	259.79 PFLOPS	613s	2.12x	3.13x
STA	22.65635	0.82024	0.19283	0.6577	0.9530	264.34 PFLOPS	548s	2.37x	3.60x
Ours Quant + Sparse	25.74353	0.83171	0.07610	0.7103	0.9435	273.01 PFLOPS	423s	3.07x	4.96x

Baseline, Wan 2.1 1.3B, 1× E2E speedup

Our FPSAttention, 4.96× E2E speedup

Figure 6: Examples of generated videos by `FPSAttention` and the Wan2.1-1.3B baseline. We showcase from five different aspects. `FPSAttention` achieves 4.96× E2E speedup, while maintaining lossless visual quality. Please click the image to play the video clip via Acrobat Reader.

`FPSAttention` **consistently outperforms** baseline methods while maintaining quality comparable to the original model.

Test-time efficiency. We evaluate computational efficiency across both 1.3B and 14B parameter variants of the Wan2.1 model, as reported in Table 2. For the 1.3B parameter model, `FPSAttention` achieves up to 2.45× speedup on 480p videos while maintaining superior quality metrics. More impressively, when tested on the larger 14B parameter model with 720p videos (indicated by ‡ in the table), `FPSAttention` achieves a substantial **4.96× end-to-end speedups** compared to the baseline.

Table 3: Effect of different tile sizes on model performance on Wan2.1 1.3B. We evaluate various tile size combinations for temporal (t), height (h), and width (w) dimensions.

Tile Size			Metrics		
t	h	w	PSNR↑	SSIM↑	LPIPS↓
3	4	4	19.87452	0.77634	0.16521
6	8	8	20.12358	0.78147	0.15982
12	16	16	20.45621	0.78932	0.15743
24	32	32	20.99712	0.79820	0.15114

Table 4: Impact of sparsity window dimensions on model performance and computational efficiency. We evaluate various combinations of temporal (t), height (h), and width (w) window sizes and measure inference kernel speedup.

Window Size			Metrics			
t	h	w	PSNR↑	SSIM↑	LPIPS↓	Speedup↑
3	3	1	19.23465	0.76128	0.17251	3.24x
5	6	10	19.94731	0.77926	0.16327	3.07x
6	6	6	20.12482	0.78341	0.16014	1.69x
6	6	1	20.45621	0.78932	0.15743	5.16x

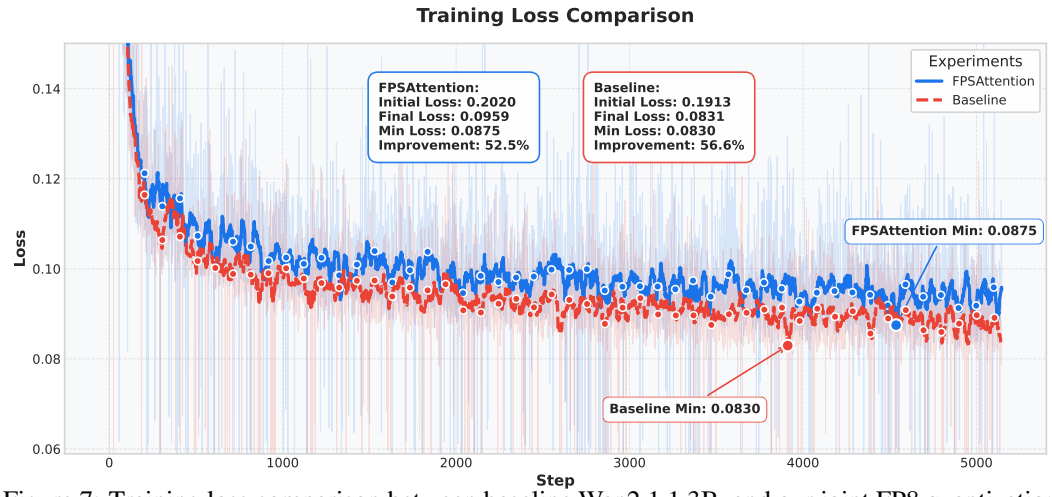


Figure 7: Training loss comparison between baseline Wan2.1 1.3B, and our joint FP8 quantization with structured sparsity. The `FPSAttention` shows slightly higher loss initially but stabilizes with comparable final performance.

4.2 Ablation Study

In this section, we conduct a comprehensive ablation study to analyze the effects of key components in `FPSAttention`. We investigate the different tile size selection for quantization and sparsity, and the relations to the hardware awareness. Additionally, we investigate how sparsity window dimensions affect model performance and computational efficiency. We also study the training stability to verify whether our `FPSAttention` training dynamics highly aligns the baseline trend.

Effect of tile size for quantization and sparsity. We investigate the impact of joint quantization and sparsity granularity by varying tile sizes along temporal (t), height (h), and width (w) dimensions, with results presented in Table 3. Our experiments demonstrate that the largest tile configuration (24,32,32) achieves optimal performance with PSNR 20.99712, SSIM 0.79820, and LPIPS 0.15114. However, we observe minimal performance differences between configurations (6,8,8), (12,16,16), and (24,32,32). Based on these findings, we implement a scheduled approach, using different tile sizes for early, mid, and late denoising steps, respectively. We emphasize the hardware-friendly configuration of (6,8,8) yields the best throughput as it aligns with flash attention block size design and is optimized for the Hopper architecture. Notably, unbalanced tile sizes, such as (3,4,4), lead to significant performance degradation, this configuration is also misalign with flash attention block size leads to inefficiency.

Effect of sparsity window dimensions. The relationship between sparsity window dimensions, inference speed, and model performance is analyzed in Table 4. Our results demonstrate that a configuration (6,6,1) using temporal window of 6, height window of 6, and width window of 1 provides an optimal balance, achieving a substantial 5.16× kernel speed-up while maintaining high visual quality. This suggests that while reducing the temporal and spatial window sizes improves efficiency, there exists a threshold beyond which visual quality deteriorates significantly.

Training stability. Joint FP8 quantization and structured sparsity initially increases training loss by 15% compared to full-precision Wan2.1 baseline (Figure 7). We mitigate these challenges through adaptive learning rate scheduling and gradient accumulation techniques. After 2,000 steps, loss convergence trajectories become nearly identical (<2% difference), confirming that our FP8 sparse attention preserves critical information pathways despite bitwidth and sparsity constraints.

5 Conclusion and Future Work

In this paper, we have introduced `FPSAttention`, a module jointly optimizing FP8 quantization and structured sparsity for video diffusion models. Through unified 3D tile-wise granularity, denoising step-aware adaptation, and hardware-friendly kernel implementation, our approach achieves up to 7.09× kernel speedup and 4.96× end-to-end acceleration without compromising generation quality. The tile-aligned approach ensures quantization and sparsity work synergistically, while step-

aware scheduling adapts compression hyperparameters to varying sensitivity across diffusion phases. Despite these results, our approach has limitations: it performs best on FP8-supporting hardware, requires additional training resources, and introduces certain hyperparameters. We currently validate `FPSAttention` on Wan2.1 due to resource constraints. Future work will focus on broadening the applicability by generalizing beyond specific architectures (e.g. Hunyuan [17] based on MMDiT [9]). We also aim to refine training resource requirements and hyperparameter management, and extend these co-design principles beyond attention mechanisms to other model components.

References

[1] Andreas Blattmann et al. Align your latents: High-resolution video synthesis with latent diffusion models. 2023.

[2] Jintao Chen, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. Ditfastattn: Attention compression for diffusion transformer models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[3] Lei Chen, Yuan Meng, Chen Tang, Xinzhu Ma, Jingyan Jiang, Xin Wang, Zhi Wang, and Wenwu Zhu. Q-dit: Accurate post-training quantization for diffusion transformers. *arXiv preprint arXiv:2406.17343*, 2024.

[4] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

[5] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.

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

[7] Zihan Ding, Chi Jin, Difan Liu, Haitian Zheng, Krishna Kumar Singh, Qiang Zhang, Yan Kang, Zhe Lin, and Yuchen Liu. Dollar: Few-step video generation via distillation and latent reward optimization. *arXiv preprint arXiv:2412.15689*, 2024.

[8] Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. Flex attention: A programming model for generating optimized attention kernels. *arXiv preprint arXiv:2412.05496*, 2024.

[9] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

[10] Yefei He, Luping Liu, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Ptqd: Accurate post-training quantization for diffusion models. *arXiv preprint arXiv:2305.10657*, 2023.

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

[12] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010.

[13] Haocheng Huang, Jiaxin Chen, Jinyang Guo, Ruiyi Zhan, and Yunhong Wang. Tcaq-dm: Timestep-channel adaptive quantization for diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, number 16, pages 17404–17412, 2025.

[14] Haocheng Huang, Jiaxin Chen, Jinyang Guo, Ruiyi Zhan, and Yunhong Wang. Tcaq-dm: Timestep-channel adaptive quantization for diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 17404–17412, 2025.

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

[16] 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.

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

[18] Andrey Kuzmin, Mart Van Baalen, Yuwei Ren, Markus Nagel, Jorn Peters, and Tijmen Blankevoort. Fp8 quantization: The power of the exponent. *Advances in Neural Information Processing Systems*, 35:14651–14662, 2022.

[19] Muyang Li, Tianle Cai, Jiaxin Cao, Qinsheng Zhang, Han Cai, Junjie Bai, Yangqing Jia, Kai Li, and Song Han. Distrifusion: Distributed parallel inference for high-resolution diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7183–7193, 2024.

[20] Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. Svdquant: Absorbing outliers by low-rank components for 4-bit diffusion models, 2025.

[21] Zefan Li et al. Cogvideox: Text-to-video diffusion models with an expert pipeline. *arXiv preprint arXiv:2408.06072*, 2024.

[22] Haoyu Lu, Guoxing Yang, Nanyi Fei, Yuqi Huo, Zhiwu Lu, Ping Luo, and Mingyu Ding. Vdt: General-purpose video diffusion transformers via mask modeling. *arXiv preprint arXiv:2305.13311*, 2023.

[23] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *Transactions on Machine Learning Research*, 2025.

[24] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15762–15772, 2024.

[25] Laura Manduchi, Kushagra Pandey, Clara Meister, Robert Bamler, Ryan Cotterell, Sina Däubener, Sophie Fellenz, Asja Fischer, Thomas Gärtner, Matthias Kirchler, et al. On the challenges and opportunities in generative ai. *arXiv preprint arXiv:2403.00025*, 2024.

[26] Paulius Micikevicius, Dusan Stosic, Neil Burgess, Marius Cornea, Pradeep Dubey, Richard Grisenthwaite, Sangwon Ha, Alexander Heinecke, Patrick Judd, John Kamalu, et al. Fp8 formats for deep learning. *arXiv preprint arXiv:2209.05433*, 2022.

[27] Beatrice Alessandra Motetti, Matteo Risso, Alessio Burrello, Enrico Macii, Massimo Poncino, and Daniele Jahier Pagliari. Joint pruning and channel-wise mixed-precision quantization for efficient deep neural networks. *IEEE Transactions on Computers*, 2024.

[28] William Peebles and Saining Xie. Scalable diffusion models with transformers. pages 4195–4205, 2023.

[29] Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision, 2024.

[30] Yihua Shao, Deyang Lin, Fanhu Zeng, Minxi Yan, Muyang Zhang, Siyu Chen, Yuxuan Fan, Ziyang Yan, Haozhe Wang, Jingcai Guo, et al. Tr-dq: Time-rotation diffusion quantization. *arXiv preprint arXiv:2503.06564*, 2025.

[31] Haihao Shen, Naveen Mellempudi, Xin He, Qun Gao, Chang Wang, and Mengni Wang. Efficient post-training quantization with fp8 formats. *Proceedings of Machine Learning and Systems*, 6:483–498, 2024.

[32] Hui Shen, Jingxuan Zhang, Boning Xiong, Rui Hu, Shoufa Chen, Zhongwei Wan, Xin Wang, Yu Zhang, Zixuan Gong, Guangyin Bao, et al. Efficient diffusion models: A survey. *arXiv preprint arXiv:2502.06805*, 2025.

[33] Andy Shih, Suneel Belkhale, Stefano Ermon, Dorsa Sadigh, and Nima Anari. Parallel sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:4263–4276, 2023.

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

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

[36] Team Wan, 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.

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

[38] Changyuan Wang, Ziwei Wang, Xiuwei Xu, Yansong Tang, Jie Zhou, and Jiwen Lu. Towards accurate post-training quantization for diffusion models. *arXiv preprint arXiv:2305.18723*, 2023.

[39] Wei Wang, Yifan Zhu, Yicong Li, and Xu Yan. Controllable Generation in Diffusion Models: A Survey. *Journal of Computer Science and Technology*, 39(3):601–626, 2024.

[40] Xiaolong Wang, Zhijian He, and Xiaojiang Peng. Artificial-intelligence-generated content with diffusion models: A literature review. *Mathematics*, 12(7):977, 2024.

[41] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[42] Junyi Wu, Haoxuan Wang, Yuzhang Shang, Mubarak Shah, and Yan Yan. Ptq4dit: Post-training quantization for diffusion transformers. *arXiv preprint arXiv:2405.16005*, 2024.

[43] Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, et al. Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity. *arXiv preprint arXiv:2502.01776*, 2025.

[44] Weiying Xie, Haowei Li, Jitao Ma, Yunsong Li, Jie Lei, Donglai Liu, and Leyuan Fang. Jointsq: Joint sparsification-quantization for distributed learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5778–5787, 2024.

[45] Zhen Xing, Qijun Feng, Haoran Chen, Qi Dai, Han Hu, Hang Xu, Zuxuan Wu, and Yu-Gang Jiang. A survey on video diffusion models. *ACM Computing Surveys*, 57(2):1–42, 2024.

[46] Sanghyun Yi, Qingfeng Liu, and Mostafa El-Khamy. Hardware-friendly static quantization method for video diffusion transformers. *arXiv preprint arXiv:2502.15077*, 2025.

[47] Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, YX Wei, Lean Wang, Zhiping Xiao, et al. Native sparse attention: Hardware-aligned and natively trainable sparse attention. *arXiv preprint arXiv:2502.11089*, 2025.

[48] Jintao Zhang, Haofeng Huang, Pengle Zhang, Jia Wei, Jun Zhu, and Jianfei Chen. Sageattention2 technical report: Accurate 4 bit attention for plug-and-play inference acceleration. *arXiv preprint arXiv:2411.10958*, 2024.

[49] Jintao Zhang, Haofeng Huang, Pengle Zhang, Jun Zhu, Jianfei Chen, et al. Sageattention: Accurate 8-bit attention for plug-and-play inference acceleration. *arXiv preprint arXiv:2410.02367*, 2024.

[50] Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. Spargeattn: Accurate sparse attention accelerating any model inference. *arXiv preprint arXiv:2502.18137*, 2025.

[51] Peiyuan Zhang, Yongqi Chen, Runlong Su, Hangliang Ding, Ion Stoica, Zhenghong Liu, and Hao Zhang. Fast video generation with sliding tile attention. *arXiv preprint arXiv:2502.04507*, 2025.

[52] 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*, pages 586–595, 2018.

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

[54] Tianchen Zhao, Xuefei Ning, Tongcheng Fang, Enshu Liu, Guyue Huang, Zinan Lin, Shengen Yan, Guohao Dai, and Yu Wang. Mixdq: Memory-efficient few-step text-to-image diffusion models with metric-decoupled mixed precision quantization, 2024.

[55] Xuanlei Zhao, Xiaolong Jin, Kai Wang, and Yang You. Real-time video generation with pyramid attention broadcast. *arXiv preprint arXiv:2408.12588*, 2024.

Appendix

Contents

A FPSAttention Algorithm

Algorithm 1 presents the core computational workflow of our FPSAttention method, which implements joint tile-wise FP8 quantization with structured sparse attention and denoising step-aware adaptation. The algorithm follows the methodology described in the main paper, incorporating tile-wise quantization for queries and keys, channel-wise quantization for values, tensor-wise quantization for attention weights, and dynamic adaptation based on denoising timesteps.

Algorithm 1 FPSAttention: Joint Tile-wise FP8 Quantization and Sparse Attention

Input: Input tensors $Q, K, V \in \mathbb{R}^{L \times d}$, denoising step t, diffusion steps D
Input: Transition points α_1, α_2, quantization granularities $\{g_{\text{coarse}}, g_{\text{fine}}, g_{\text{intermediate}}\}$
Input: Window sizes $\{W_{\text{sparse}}, W_{\text{dense}}, W_{\text{medium\_density}}\}$, tile scheme $\mathcal{T}$
Output: Output tensor $X \in \mathbb{R}^{L \times d}$ (BF16/FP16)

▷ 1. Denoising Step-aware Parameter Selection
$t_1 \leftarrow \alpha_1 \cdot D, t_2 \leftarrow \alpha_2 \cdot D$
if $t \leq t_1$ **then**
  $g(t) \leftarrow g_{\text{coarse}}, W(t) \leftarrow W_{\text{sparse}}$ ▷ Early steps
else if $t_1 < t \leq t_2$ **then**
  $g(t) \leftarrow g_{\text{fine}}, W(t) \leftarrow W_{\text{dense}}$ ▷ Mid steps
else
  $g(t) \leftarrow g_{\text{intermediate}}, W(t) \leftarrow W_{\text{medium\_density}}$ ▷ Late steps
end if
▷ 2. Tile-wise FP8 Quantization for Q and K
Partition Q, K into tiles $\{\mathcal{T}_u\}$ with granularity $g(t)$
for each tile $\mathcal{T}_u$ **do**
  $s_u^Q \leftarrow \max_{(i,j)\in\mathcal{T}_u} |Q_{i,j}|/M_{\text{FP8\_max}}$
  $s_u^K \leftarrow \max_{(i,j)\in\mathcal{T}_u} |K_{i,j}|/M_{\text{FP8\_max}}$
  $\hat{Q}_{i,j} \leftarrow \text{FP8}(Q_{i,j}; s_u^Q)$ for $(i,j) \in \mathcal{T}_u$
  $\hat{K}_{i,j} \leftarrow \text{FP8}(K_{i,j}; s_u^K)$ for $(i,j) \in \mathcal{T}_u$
end for
▷ 3. Channel-wise FP8 Quantization for V
for each channel $j \in \{1, \ldots, d\}$ **do**
  $s_j^V \leftarrow \max_{i \in L} |V_{i,j}|/M_{\text{FP8\_max}}$
  $\hat{V}_{i,j} \leftarrow \text{FP8}(V_{i,j}; s_j^V)$ for all i
end for
▷ 4. Structured Sparse Attention Computation via FlexAttention
Define neighborhood $\mathcal{W}(u)$ based on window size $W(t)$:
$\mathcal{W}(u) = \{v : \|c_u - c_v\|_\infty \leq (W_t/(2T_t), W_h/(2T_h), W_w/(2T_w))\}$
▷ Configure FlexAttention mask and score modification functions
Define `mask_mod`(b, h, q, k) = **True** if tile$(q) \in \mathcal{W}$(tile(k)), **False** otherwise
Define `score_mod`$(S, b, h, q, k) = S$ ▷ Identity for quantized inputs
▷ Execute FlexAttention with quantized inputs and custom modifications
$\hat{X} \leftarrow$ `FlexAttention`$(\hat{Q}, \hat{K}, \hat{V}$, `score_mod`, `mask_mod`$)$
▷ 5. Dequantize Output to Target Precision
$X \leftarrow \text{Dequantize}(\hat{X})$ ▷ Dequantize to BF16/FP16
return X

Key Algorithmic Components

The algorithm implements the four core innovations described in the main paper:

- **Denoising Step-aware Adaptation**: Lines 2-8 implement the adaptive scheduling strategy from Equation 6 in the main paper, dynamically adjusting quantization granularity $g(t)$ and sparsity window size $W(t)$ based on the current denoising step t.

- **Tile-wise FP8 Quantization for Q and K**: Lines 10-15 partition queries and keys into 3D tiles with step-dependent granularity and compute per-tile scaling factors s_u^Q and s_u^K to minimize quantization error within each tile.
- **Channel-wise FP8 Quantization for V**: Lines 17-20 apply channel-wise quantization to the value matrix, preserving fine-grained channel information that is critical for generation quality.
- **FlexAttention-based Sparse Attention**: Lines 22-26 implement structured sparse attention using FlexAttention's `mask_mod` and `score_mod` interfaces, enabling hardware-optimized execution with tile-wise sparsity patterns that generate exactly $M \times |\mathcal{W}(u)|$ dense attention blocks.
- **Output Dequantization**: Line 28 dequantizes the FlexAttention output to the target precision (BF16/FP16) to maintain compatibility with the downstream network components.

This implementation ensures full compatibility with the theoretical framework while enabling practical hardware acceleration through structured computation patterns and optimal memory access patterns.

B Additional Implementation Details

Models. We implement and evaluate FPSAttention on the Wan architecture [37], leveraging both 1.3B and 13B parameter variants to demonstrate scalability. The Wan models feature a DiT backbone with cross-attention for text conditioning and temporal attention for inter-frame modeling. Our implementation maintains architectural fidelity while seamlessly integrating FP8 quantization across attention and feed-forward components. The joint quantization and sparsity mechanisms are realized through FlexAttention's score and mask modification interfaces, with the resulting fused kernels compiled via Triton for optimal execution on Hopper architectures.

Hardware. Experiments utilize a distributed computing cluster with high-performance GPU nodes, each containing 192 CPU cores, 960GB system memory, and 8×NVIDIA H20 GPUs (96GB each). InfiniBand interconnects ensure high-bandwidth inter-node communication for distributed training. Training scales from 16 nodes (1.3B model) to 64 nodes (13B model), requiring approximately 7 days per configuration to achieve convergence.

Dataset. Training employs a curated high-quality video dataset processed through a comprehensive filtering pipeline. The preprocessing workflow includes automated subtitle removal, black-border cropping, and monochrome video exclusion, followed by quality-based filtering using established metrics (Q-Align > 3.5, Aesthetic Score > 2.0, optical flow magnitude 0.05–2.0). After deduplication, videos are standardized to 480p resolution, 16fps frame rate, and 5-second duration to optimize the computational efficiency-quality balance across both model scales.

Evaluation. Performance assessment utilizes the VBench benchmark [17], following established protocols [55, 19] with 5-video sampling per prompt. Evaluation encompasses 16 comprehensive VBench dimensions covering aesthetic quality, temporal consistency, motion dynamics, and semantic understanding. Additional quantitative metrics include PSNR [12], SSIM, and LPIPS [52] to provide multi-faceted quality assessment.

Baselines. Our comparative analysis includes representative approaches from three categories: (1) sparsity-based methods (SparseVideoGen [43], STA), (2) quantization-focused techniques (SageAttention [49]), and (3) joint optimization methods (SpargeAttn [50]). This selection enables comprehensive evaluation of FPSAttention against both specialized single-optimization approaches and competing joint methods, providing a thorough assessment of our framework's relative performance and efficiency gains.

C Ablation Study: Challenges of Naive Quantization and Sparsity Combination

To validate our core motivation that naive combination of FP8 quantization and sparsity presents significant challenges, we conduct a comprehensive ablation study comparing three key approaches:

(1) the baseline full-precision model, (2) a training-free naive combination of quantization and sparsity, and (3) our proposed FPSAttention method with joint optimization.

Table 5 presents a detailed comparison across all VBench metrics for the Wan 1.3B model. The training-free approach applies standard FP8 quantization and sparse attention patterns without joint optimization or denoising step-aware adaptation. As hypothesized, this naive combination leads to substantial performance degradation across nearly all evaluation metrics.

Table 5: Ablation study demonstrating the challenges of naive quantization and sparsity combination. We compare baseline full-precision (Baseline), training-free naive combination (Training-Free), and our joint optimization approach (FPSAttention) on Wan 1.3B across all VBench metrics. The severe degradation in the training-free approach validates the need for holistic joint optimization.

Metric	Baseline	Training-Free	FPSAttention
Aesthetic Quality	0.6105	0.2892	**0.6240**
Appearance Style	0.7157	0.7874	**0.7252**
Background Consistency	**0.9503**	0.9280	0.9156
Color	0.9049	0.4836	**0.8932**
Dynamic Degree	0.3014	0.3750	**0.4195**
Human Action	0.7720	0.0200	**0.7780**
Imaging Quality	0.6708	0.6868	**0.7103**
Motion Smoothness	0.9527	0.9513	0.9413
Multiple Objects	0.6091	0.0000	**0.6665**
Object Class	0.7710	0.0109	**0.8185**
Overall Consistency	0.6453	0.1206	**0.6893**
Quality Score	0.8332	0.7473	**0.8428**
Scene	0.3030	0.0129	**0.3870**
Semantic Score	0.6768	0.1733	**0.7088**
Spatial Relationship	0.7317	0.0008	**0.7659**
Subject Consistency	0.9457	0.8887	0.9338
Temporal Flickering	**0.9844**	0.9401	0.9336
Temporal Style	0.6382	0.1239	**0.6558**
Total Score	0.8019	0.6325	**0.8160**
Performance Drop	–	-21.1%	+1.8%

Key Findings: The results clearly demonstrate the challenges inherent in naive quantization and sparsity combination:

- **Severe Quality Degradation**: The training-free approach achieves only 0.6325 total score compared to the baseline's 0.8019, representing a substantial 21.1% performance drop.
- **Critical Failure Modes**: Several metrics show near-zero performance in the training-free approach, including Human Action (0.02), Multiple Objects (0.0), Object Class (0.011), and Spatial Relationship (0.0008), indicating complete failure in complex semantic understanding tasks.
- **Magnified Quantization Errors**: As predicted by our theoretical analysis, sparsity mechanisms amplify quantization errors in high-magnitude attention scores. This is particularly evident in metrics requiring fine-grained semantic understanding, where the interaction between quantization noise and sparse token selection leads to catastrophic information loss.
- **Joint Optimization Success**: In contrast, our FPSAttention approach not only avoids the degradation seen in naive combination but actually improves upon the baseline (0.8160 vs 0.8019, +1.8% improvement), validating the effectiveness of our denoising step-aware joint optimization strategy.

This ablation study emphasizes the necessity of our training-aware co-design scheme.

D VBench Full Evaluation Results

Tables 6 and 7 present comprehensive evaluation results of our method compared to various baselines on VBench for Wan 1.3B and Wan 13B models, respectively. These results demonstrate the effectiveness of our joint FP8 quantization and sparsity approach across multiple video quality metrics.

Table 6 shows performance comparisons across seven methods on the Wan 1.3B model: the baseline (Base), SageAttention (SageAtt) [49], SpargeAttention (SpargeAtt) [50], SparseVideoGen (SparseVG) [43], Sliding Tile Attention (STA) [51], our quantization-only variant (Ours-Q), and our full joint quantization and sparsity method (Ours-Q+S). The evaluation covers 18 comprehensive metrics including aesthetic quality, motion dynamics, temporal consistency, and semantic understanding. Our full method (Ours-Q+S) achieves the highest total score of 0.8160, demonstrating superior performance compared to methods that apply quantization or sparsity independently.

Table 7 presents similar comparisons for the larger 13B model, where our method continues to achieve competitive performance while providing substantial computational savings. The results validate that our approach scales effectively to larger model sizes while maintaining video generation quality across diverse evaluation criteria.

Table 6: Performance comparison of different methods on Wan 1.3B across VBench metrics. We compare the baseline (Base), SageAttention (SageAtt), SpargeAttention (SpargeAtt), SparseVideoGen (SparseVG), Sliding Tile Attention (STA), our quantization-only variant (Ours-Q), and our full joint method (Ours-Q+S). Bold values indicate the best performance for each metric.

Metric	Base	SageAtt	SpargeAtt	SparseVG	STA	Ours-Q	Ours-Q+S
Aesthetic Quality	0.6105	0.6104	0.5668	0.563	0.5661	0.6091	**0.624**
Appearance Style	0.7157	0.715	0.7744	0.2253	0.7952	0.6922	**0.7252**
Background Consistency	**0.9503**	0.95	0.9123	0.9525	0.9284	0.9472	0.9156
Color	0.9049	0.8866	0.8903	0.9037	0.8974	**0.9146**	0.8932
Dynamic Degree	0.3014	0.307	0.3222	**0.7139**	0.5722	0.3222	0.4195
Human Action	0.772	0.75	0.734	0.73	0.622	0.75	**0.778**
Imaging Quality	0.6708	0.6699	0.6541	0.6729	0.6626	0.6798	**0.7103**
Motion Smoothness	0.9527	0.9527	0.9139	**0.9726**	0.9649	0.9496	0.9413
Multiple Objects	0.6091	0.5837	0.4715	0.471	0.4043	0.6011	**0.6665**
Object Class	0.7710	0.7695	0.6859	0.6935	0.5818	0.7851	**0.8185**
Overall Consistency	0.6453	0.6451	0.6492	**0.6935**	0.6236	0.6478	0.6893
Quality Score	0.8332	0.8337	0.8049	0.2336	0.7994	0.8363	**0.8428**
Scene	0.3030	0.3092	0.2192	0.1732	0.1995	0.3200	**0.3870**
Semantic Score	0.6768	0.6704	0.6412	0.6342	0.6012	0.6780	**0.7088**
Spatial Relationship	0.7317	0.7364	0.7217	0.6469	0.6863	0.7438	**0.7659**
Subject Consistency	0.9457	0.9453	0.8982	0.9292	0.8993	**0.9458**	0.9338
Temporal Flickering	**0.9844**	0.9841	0.9647	0.9883	0.9652	0.9822	0.9336
Temporal Style	0.6382	0.6385	0.6247	0.2265	0.6005	0.6475	**0.6558**
Total Score	0.8019	0.8011	0.7722	0.7827	0.7597	0.8046	**0.8160**

E Clarification on VBench Evaluation Metrics

In this study, we observed that some of the baseline methods we reproduced (including some of our own exploratory experiments prior to FPSAttention) might yield VBench scores slightly lower than those reported in their respective official publications. We attribute this primarily to the following factors: Randomness: The inherent stochasticity in video generation models can lead to slight variations in results and vBench scores across multiple runs, even with identical settings. Prompt Extension: Many prior works [21] may employ specific prompt extension strategies to enrich input prompts. This can influence the content and quality scores of the generated videos. We didn't employ this optimization. Classifier-Free Guidance (CFG) Scale and Other Sampling Strategies: Different CFG scale values and other sampling parameters (e.g., number of sampling steps) significantly impact generation quality. While we endeavored to follow the descriptions in the respective baseline papers, subtle parameter differences might still exist. It is worth noting that similar observations have been made in other research. For instance, in the work on Sliding Tile Attention (STA) [51], their reproduced HunyuanVideo baseline also exhibited lower VBench performance compared to

Vbench's official leaderboard. Despite these potential metric variations, we emphasize that all methods in this study (including our FPSAttention and all compared baselines) were evaluated under an identical VBench evaluation pipeline and parameter settings, ensuring a fair comparison. Our primary research objective is to demonstrate the significant inference speedup achieved by FPSAttention while maintaining comparable (or superior) generation quality relative to baseline methods.

Table 7: Performance comparison of different methods on Wan 13B across VBench metrics. We compare the baseline (Base), SageAttention (SageAtt), SpargeAttention (SpargeAtt), SparseVideoGen (SparseVG), Sliding Tile Attention (STA), and our full joint method (Ours-Q+S). Bold values indicate the best performance for each metric.

Metric	**Base**	**SageAtt**	**SpargeAtt**	**SparseVG**	**STA**	**Ours-Q+S**
Aesthetic Quality	0.6204	0.6209	0.5875	0.6246	0.6033	**0.624**
Appearance Style	0.2164	0.2163	0.7586	0.2306	0.2303	0.2073
Background Consistency	**0.9691**	0.9687	0.9355	0.9589	0.9573	0.9377
Color	0.8879	0.8825	0.8768	0.8883	0.8814	**0.8932**
Dynamic Degree	0.6944	0.7028	0.6028	0.6806	0.7028	**0.8389**
Human Action	0.796	0.8	0.78	0.816	0.778	**0.816**
Imaging Quality	0.6715	0.6724	0.635	0.6868	0.6577	**0.7103**
Motion Smoothness	0.9828	0.9828	0.9413	0.982	0.9714	**0.9804**
Multiple Objects	0.6627	0.6477	0.6066	0.7012	0.6576	0.666
Object Class	0.8299	0.8312	0.7896	**0.8712**	0.81	0.8185
Overall Consistency	0.6912	0.6912	0.6975	0.708	0.6893	0.6893
Quality Score	0.6577	**0.8428**	0.8134	0.7421	0.8246	0.7103
Scene	0.3669	0.3049	0.3495	**0.4129**	0.3387	0.3182
Semantic Score	**0.7572**	0.7091	0.6969	0.7421	0.7077	0.7088
Spatial Relationship	0.7364	0.7405	0.7526	**0.8056**	0.7405	0.7661
Subject Consistency	0.9528	0.953	0.9173	0.9489	0.953	0.9435
Temporal Flickering	**0.9922**	**0.9922**	0.969	0.9891	**0.9922**	0.9754
Temporal Style	0.2408	0.2408	0.6607	0.2438	0.2408	**0.6558**
Total Score	0.8153	0.8158	0.7901	**0.8196**	0.8012	0.816

The results demonstrate that our joint FP8 quantization and sparsity approach achieves competitive or superior performance compared to specialized methods focusing solely on either quantization or sparsity. For the Wan 1.3B model, our method achieves the highest total score (0.8160), outperforming the baseline (0.8019) while providing significant computational benefits. Similarly, for the Wan 13B model, our approach performs on par with the best-performing methods while offering substantial memory and compute savings through the combination of quantization and structured sparsity.

F Training Hyperparameters

Table 8 presents the key hyperparameters used in our experiments for both Wan 1.3B and 13B model training configurations. These hyperparameters were carefully selected to balance training stability, convergence speed, and final model performance while accommodating the constraints imposed by FP8 quantization and structured sparsity.

Table 8: Comprehensive hyperparameter configuration for Wan 1.3B and 13B model training and evaluation. The table covers model architecture specifications, training parameters, diffusion scheduler settings, data configuration, and system-level precision settings used in our experiments.

Category	Parameter	Wan 1.3B	Wan 13B
Model Architecture	Model Type	WanX21FPS	**WanX21FPS-13B**
	Model Dimension	1536	**5120**
	Number of Layers	30	**40**
	Number of Heads	12	**40**
	FFN Dimension	8960	**13824**
	Input/Output Dimension	16	16
	Frequency Dimension	256	256
	Text Dimension	4096	4096
	Patch Size	[1, 2, 2]	[1, 2, 2]
Training	Learning Rate	5e-6	5e-6
	Weight Decay	1e-4	1e-4
	Gradient Clipping	1.0	1.0
	Warmup Steps	200	200
	EMA Decay	0.99	0.99
	Adam Epsilon	1e-15	1e-15
Diffusion Scheduler	Scheduler Type	rflow-wanx	rflow-wanx
	Number of Timesteps	1000	1000
	Sample Steps	50	50
	CFG Scale	5.0	5.0
	Sample Shift	5.0	5.0
	Transform Scale	5.0	5.0
	Sample Method	logit-normal	logit-normal
Data & Sequence	Text Length	512	512
	Max Sequence Length	75600	75600
	Sample FPS	16	16
	Video Resolution	480p	480p
	Video Duration	5s	5s
	Prompt Uncond Probability	0.1	0.1
System & Precision	Data Type	fp8	fp8
	Training Mode	FSDP	FSDP
	Gradient Checkpointing	True	True
	Quantization	True	True
	Sequence Parallel Degree	1	**4**

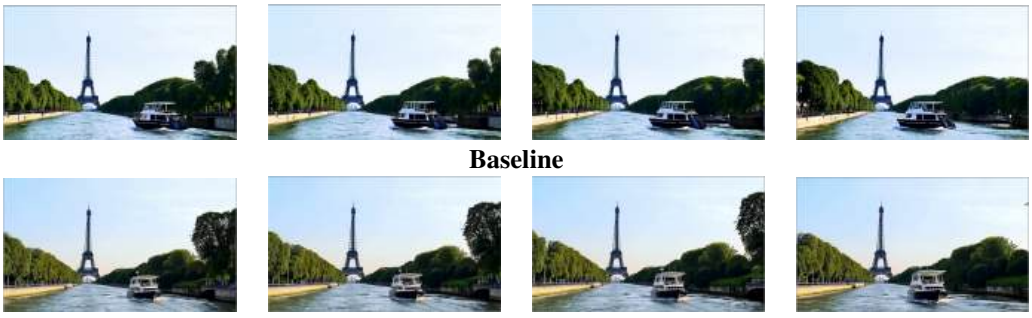
G Limitations

While FPSAttention demonstrates strong performance across our evaluation scenarios, there are some considerations for broader adoption. Our approach works best with modern FP8-capable GPUs such as NVIDIA Hopper architectures, though it can still provide benefits on older hardware with reduced FP8 acceleration. The method benefits from quantization-aware training to achieve optimal results, which involves a moderate increase in training time compared to post-training quantization approaches. The denoising step-aware scheduling includes several hyperparameters (transition points α_1 and α_2, quantization granularities, and window sizes) that can be optimized for different model architectures and datasets. Our current evaluation focuses on the Wan2.1 architecture, and the approach shows strong promise for extension to other video diffusion transformer architectures (e.g., HunyuanVideo, CogVideoX). Additionally, while our tile-wise approach achieves good hardware utilization across tested configurations, there are opportunities for architecture-specific optimization to further improve performance on different GPU memory hierarchies.

H Visualization

The following qualitative comparison demonstrates that our FPSAttention method generates video frames that are visually nearly identical to the baseline Wan model with size of 1.3B. This visual similarity across diverse scenarios—including boats, fish, dogs, desert landscapes, couples, trains, cars, cats, and robot DJs—validates that our joint FP8 quantization and sparsity optimization achieves essentially lossless performance while providing substantial computational acceleration.

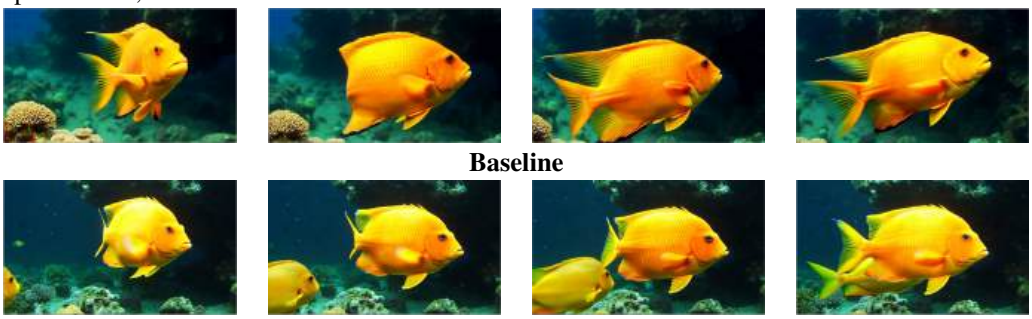
Table 9: Qualitative comparison on the boat group. Prompt: 'A boat sailing leisurely along the Seine River with the Eiffel Tower in background in super slow motion' . Top: Baseline; Bottom: FPSAttention.



Baseline

FPSAttention

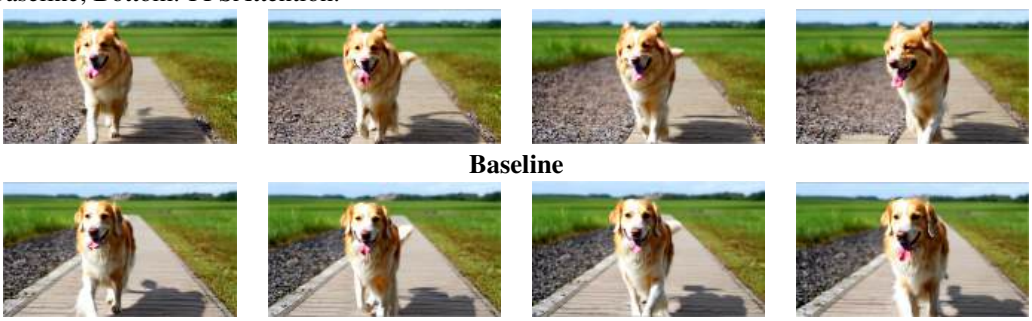
Table 10: Qualitative comparison on the fish group. Prompt : 'Golden fish swimming in the ocean'. Top: Baseline; Bottom: FPSAttention.



Baseline

FPSAttention

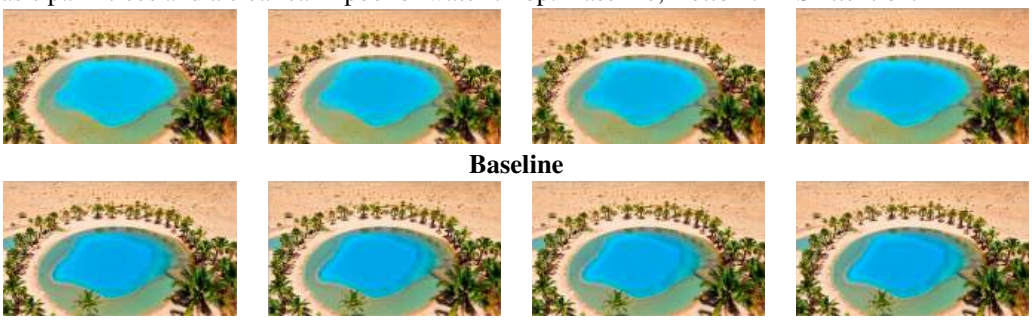
Table 11: Qualitative comparison on the dog group. Prompt: 'A dog enjoying a peaceful walk'. Top: Baseline; Bottom: FPSAttention.



Baseline

FPSAttention

Table 12: Qualitative comparison on the desert group. Prompt: 'Static view on a desert scene with an oasis palm trees and a clear calm pool of water'. Top: Baseline; Bottom: FPSAttention.



Baseline

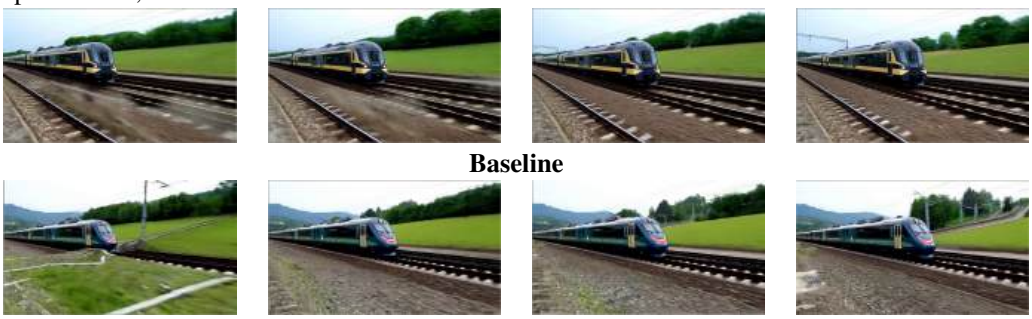
FPSAttention

Table 13: Qualitative comparison on the couple group. Prompt: 'A couple in formal evening wear going home get caught in a heavy downpour with umbrellas'. Top: Baseline; Bottom: FPSAttention.

Baseline

FPSAttention

Table 14: Qualitative comparison on the train group. Prompt: 'A train accelerating to gain speed'. Top: Baseline; Bottom: FPSAttention.



Baseline

FPSAttention

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We have properly made the main claims in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.

Table 15: Qualitative comparison on the rock group. Prompt: 'A tranquil tableau of at the edge of the Arabian Desert, the ancient city of Petra beckoned with its enigmatic rock-carved façades'. Top: Baseline; Bottom: FPSAttention.

Baseline

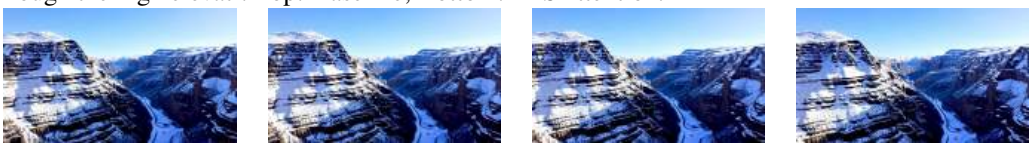
FPSAttention

Table 16: Qualitative comparison on the nursery group. Prompt: 'Nursery'. Top: Baseline; Bottom: FPSAttention.

Baseline

FPSAttention

Table 17: Qualitative comparison on the snow group. Prompt: 'Snow rocky mountains peaks canyon. snow blanketed rocky mountains surround and shadow deep canyons. The canyons twist and bend through the high elevat'. Top: Baseline; Bottom: FPSAttention.



Baseline

FPSAttention

- 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 NA 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?

Table 18: Qualitative comparison on the book group. Prompt: 'A person is reading book'. Top: Baseline; Bottom: FPSAttention.

Baseline

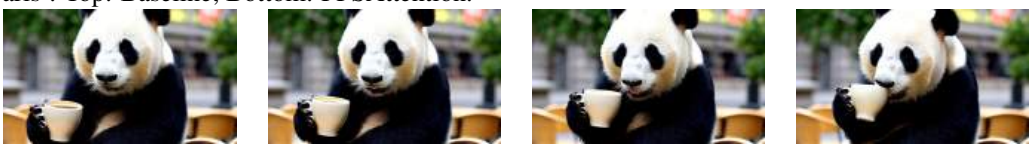
FPSAttention

Table 19: Qualitative comparison on the space group. Prompt: 'An astronaut flying in space'. Top: Baseline; Bottom: FPSAttention.

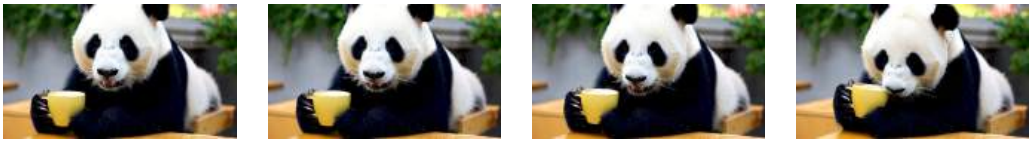
Baseline

FPSAttention

Table 20: Qualitative comparison on the panda group. Prompt: 'A panda drinking coffee in a cafe in Paris'. Top: Baseline; Bottom: FPSAttention.



Baseline



FPSAttention

Answer: [Yes]

Justification: Yes, we include the limitations in last section and Appendix.

Guidelines:

- The answer NA 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: See sections in methodology and appendixes.

   Guidelines:

   - The answer NA 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: We provide detailed experiments setup on Experiment sections, we also have detailed algorithm in appendix for Reproducibility.

   Guidelines:

   - The answer NA 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 code after acceptance.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide implementation details.

   Guidelines:

- The answer NA 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: We verify our results on huge size of DiT models, repeat computation are expensive. But we repeat a few times of exponents to obtain the mean result.

   Guidelines:

   - The answer NA 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: We give the competing requirements in the experiment setup.

   Guidelines:

   - The answer NA 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: Yes we do.

Guidelines:

- The answer NA 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: We have included the broader impacts in the Conclusion.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our methodology not release new data or models.

Guidelines:

- The answer NA 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: Yes, we give proper citations and reference for used resources.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: No new assets.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: This paper does not study human subjects.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: We are not subject to the approval of IRB.

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: We use LLMs to check and correct grammar errors.

    Guidelines:

    - The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.