Mixture of Distributions Matters: Dynamic Sparse Attention for Efficient Video Diffusion Transformers

Anonymous Authors[1]

Abstract

While Diffusion Transformers (DiTs) have achieved notable progress in video generation, this long-sequence generation task remains constrained by the quadratic complexity inherent to self-attention mechanisms, creating significant barriers to practical deployment. Although sparse attention methods attempt to address this challenge, existing approaches either rely on oversimplified static patterns or require computationally expensive sampling operations to achieve dynamic sparsity, resulting in inaccurate pattern predictions and degraded generation quality. To overcome these limitations, we propose a **M**ixture-**O**f-**D**istribution **DiT** (**MOD-DiT**), a novel sampling-free dynamic attention framework that accurately models evolving attention patterns through a two-stage process. First, MOD-DiT leverages prior information from early denoising steps and adopts a distributed mixing approach to model an efficient linear approximation model, which is then used to predict mask patterns for a specific denoising interval. Second, an online block masking strategy dynamically applies these predicted masks while maintaining historical sparsity information, eliminating the need for repetitive sampling operations. Extensive evaluations demonstrate consistent acceleration and quality improvements across multiple benchmarks and model architectures, validating MOD-DiT's effectiveness for efficient, high-quality video generation while overcoming the computational limitations of traditional sparse attention approaches.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

1. Introduction

The recent success of diffusion models in image generation has sparked significant interest in extending their application to video synthesis (Ho et al., 2020; Rombach et al., 2022). Early video generation approaches, such as SVD (Blattmann et al., 2023) and Dynamicrafter (Guo et al., 2024), utilized efficient 2D+1D frameworks. However, these methods struggled with maintaining spatio-temporal consistency due to limited interaction between spatial and temporal features. To address this, the introduction of 3D full-attention Video Diffusion Transformers (vDiTs) (Peebles & Xie, 2023) significantly improved video generation by enhancing temporal coherence. Building upon this, subsequent models like OpenSora (Team, 2024), CogVideoX (Yang et al., 2024), HunyuanVideo (Kong et al., 2024), and Wan2.1 (Wang et al., 2025) have demonstrated impressive performance across diverse applications, such as animation generation (Chen et al., 2024), video editing (Qi et al., 2023), and long-term video modeling (Henschel et al., 2025).

Despite these advances, the quadratic complexity of self-attention in vDiTs poses a critical bottleneck for practical deployment. While various sparse attention techniques have been proposed to address this computational challenge (Chen et al., 2025; Xi et al., 2025; Li et al., 2025), they struggle to balance efficiency with the modeling accuracy required during inference. Current sparse attention methods for vDiTs rely on fixed sparsity patterns—including vertical distributions (Chen et al., 2025), binary classification of spatial and temporal heads (Xi et al., 2025), and global patterns based on energy decay (Li et al., 2025). However, these rigid patterns fail to capture the true nature of attention distributions in vDiTs, which exhibit dynamic and probabilistic blending of distinct patterns throughout the denoising process. This mismatch between fixed sparse masks and dynamic attention requirements results in suboptimal performance and degraded video generation quality.To bridge this gap, two fundamental questions must be addressed for the efficient deployment of dynamic sparse attention in DiT inference:

Q1. How can we accurately model the dynamic mixture of distribution patterns in the attention maps of vDiTs?

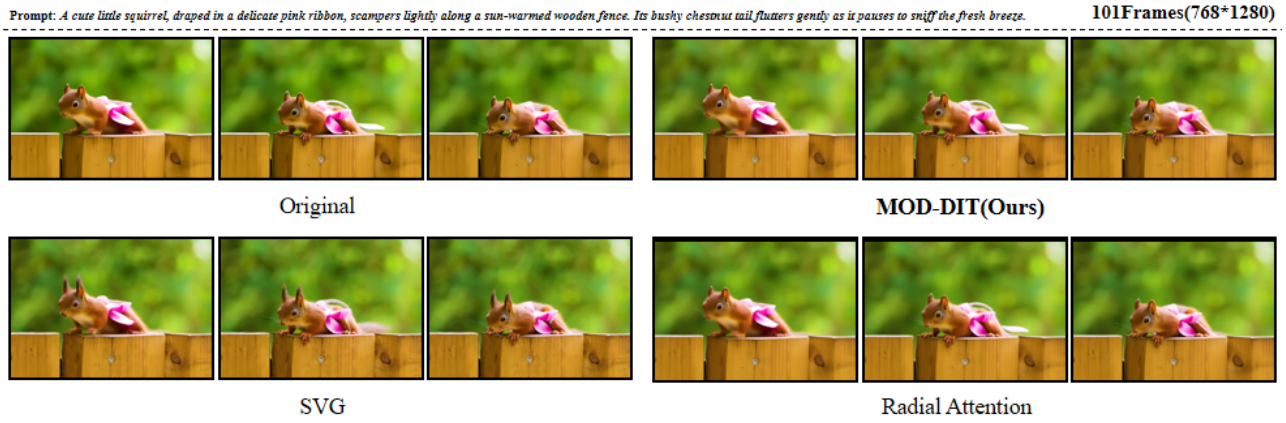


Figure 1. Comparison of the visualization effects of different sparse attention methods on HunyuanVideo(Kong et al., 2024). Our method MOD-DiT consistently achieves 2.2× speedup, and keep almost the same as original videos.

Current sparse attention methods rely on static, over-simplified representations of attention patterns. Sparse-vDiT (Chen et al., 2025) constrains patterns to vertical and block-diagonal structures, while SVG (Xi et al., 2025) categorizes attention heads into binary spatial and temporal types, and Radial Attention (Li et al., 2025) employs fixed global patterns based on energy decay. These approaches fail to capture a crucial characteristic: attention maps in vDiTs exhibit a dynamic mixture of three distinct patterns—block-diagonal for intra-frame temporal coherence, parallel-to-main-diagonal for inter-frame spatial correlation, and vertical for global dependencies across tokens. Furthermore, these patterns continuously evolve throughout the denoising process, making their accurate modeling essential for high-quality video generation.

Q2. How can we design a sampling-free dynamic sparse attention algorithm that adapts to denoising steps and enables high-performance implementation?

Existing dynamic sparse attention techniques (Zhang et al., 2025b; Tan et al., 2025; Xia et al., 2025; Jiang et al., 2024) suffer from three critical limitations. First, sampling-based methods incur substantial computational overhead that negates their intended efficiency gains. Second, even purportedly dynamic approaches fail to adapt to evolving attention distributions—a rigidity that undermines the accuracy-efficiency trade-off. Third, these methods overlook the impact of dynamic denoising time steps on sparse attention, which is essential for capturing the evolving patterns in video diffusion transformers.

Main Contributions. To address the aforementioned open questions, this paper introduces **M**ixture-**O**f-**D**istribution **DiT** (**MOD-DiT**), a novel dynamic sparse attention framework for vDiTs. Our contributions are as follows:

- **Mixture Pattern identification.** We reveal that attention maps in vDiTs exhibit a dynamic mixture of three distinct distribution patterns—block-diagonal, parallel-to-main-diagonal, and vertical—that evolve throughout the denoising process. This identification of patterns challenges existing static models and lays the foundation for more accurate modeling of attention patterns. *(Addresses Q1)*
- **Sampling-free dynamic sparse attention.** We propose an algorithm that eliminates sampling overhead by leveraging prior information from early denoising steps and employing distributed mixing to predict stage-specific mask patterns. Our approach dynamically adjusts both mask patterns and sparsity ratios throughout the denoising process to optimize the accuracy-efficiency trade-off. The algorithm is further optimized for high-performance computing, enabling seamless integration with mainstream APIs for scalable deployment. *(Addresses Q2)*
- **Superior performance.** Extensive experiments demonstrate that MOD-DiT achieves 1.89× on CogVideoX-v1.5(Yang et al., 2024) and 2.29× on hunyuan(Kong et al., 2024) model inference speedup while establishing state-of-the-art results on VBench (Huang et al., 2023). These results confirm the effectiveness of our solutions to both fundamental challenges. *(Validates solutions to Q1 and Q2)*

A comprehensive review of related efforts, including video diffusion transformers, sparse attention mechanisms, and efficient diffusion optimizations, along with their connections to our work, is provided in Appendix A.

2. Preliminaries

Full Attention. Full attention is the core component of Diffusion Transformers. Let B_b denote the batch size, N

the total number of tokens, D the original feature dimension, H the number of attention heads, and $d = D/H$ the dimension per head. Given input features $I \in \mathbb{R}^{B_b \times N \times D}$, linear projections produce query, key, and value tensors $Q, K, V \in \mathbb{R}^{B_b \times H \times N \times d}$. For each head h, the attention output $O_h \in \mathbb{R}^{B_b \times N \times d}$ is computed as:

$$O_h = \text{softmax}\left(\frac{Q_h K_h^T}{\sqrt{d}}\right) V_h$$

The outputs from all H heads are concatenated and projected back to $\mathbb{R}^{B_b \times N \times D}$ to produce the final attention output. While effective, the all-to-all token interaction incurs $\mathcal{O}(N^2)$ computational complexity, severely limiting scalability for video generation tasks.

Sparse Attention. Full attention incurs $\mathcal{O}(N^2)$ computational and memory complexity, where N denotes the number of tokens per attention head. This quadratic scaling becomes prohibitive for vDiTs in long-sequence video generation. Sparse attention addresses this bottleneck by restricting token interactions through a binary mask $M \in \mathbb{R}^{N \times N}$, where $M_{p,q} \in \{-\infty, 0\}$. Setting $M_{p,q} = -\infty$ excludes the interaction between query token p and key token q, while $M_{p,q} = 0$ preserves it. Formally, sparse attention is:

$$\text{SA}(Q, K, V) = \text{softmax}(A + M)V, \tag{1}$$

where Q, K, and V denote the query, key, and value tensors, and $A = QK^T/\sqrt{d}$ represents the scaled attention scores.

Dynamic Sparse Attention. Among dynamic sparse attention variants (Xia et al., 2025; Jiang et al., 2024; Xi et al., 2025; Zhang et al., 2025b), dynamic sparse attention methods typically adjust the mask M across two dimensions: inputs and attention heads to enhance adaptability. While this focus on input and head-level variability improves computational efficiency, it often overlooks the critical dimension of denoising steps. This omission leads to suboptimal sparse patterns that fail to capture the temporal dynamics of diffusion processes, ultimately compromising accuracy in high-quality video generation for vDiTs. See Figure 3 and 11 for more details. Notably, efficient sparse attention implementations like SageAttention (Zhang et al., 2025a) enable hardware-level skipping of invalid token interactions, providing a practical foundation for high-performance sparse inference.

Remark 2.1. By adopting a **sampling-free** approach during denoising steps (see Section 4.2 for details), MOD-DiT accurately **characterizes and predicts the dynamic patterns** of the Attention Sparsity map. This enables the generation of dynamic masks across three dimensions—**inputs, attention heads, and denoising steps**—resulting in both excellent video generation performance and high efficiency.

3. Mixture of Patterns in Attention

3.1. Dynamic Evolution of Attention Sparsity Maps

Attention Sparsity Map. We process each attention head independently: for the attention map A_h of the h-th head, we partition A_h into non-overlapping blocks of size $B \times B$, yielding $n = N/B$ blocks per dimension, and convert it to a sparsity map $S \in \mathbb{R}^{n \times n}$. The sparsity value for block (i, j) is defined as:

$$S_{h,(i,j)} = \frac{1}{B^2} \sum_{x=0}^{B-1} \sum_{y=0}^{B-1} \mathbb{I}\left(A_{h,(iB+x,jB+y)} < \eta\right), \tag{2}$$

where η denotes the sparsity threshold and $\mathbb{I}(\cdot)$ is the indicator function. Less $S_{h,(i,j)}$ indicates more informative blocks. For the convenience of symbol reading, we uniformly use S instead of S_h in the subsequent descriptions and formulas.

Pattern Mixtures. As illustrated in Figure 2, the attention sparsity map $S^{(t)} \in \mathbb{R}^{n \times n}$ at iteration t exhibits a fundamental characteristic during vDiT denoising —specifically, we performed full attention with the CogVideoX-v1.5 model (Yang et al., 2024) and visualized the attention sparsity maps extracted at each denoising step: it asymptotically converges to three distinct structured patterns—block-diagonal for intra-frame temporal coherence, parallel-to-main-diagonal for inter-frame spatial correlation, and vertical for global cross-token dependencies. During mid-to-late denoising steps, however, the sparsity map manifests as a dynamic mixture of these patterns, as exemplified in pattern (d).

3.2. Generalized Linear Approximation Model

To accurately model the dynamic pattern mixture in $S^{(t)}$, we propose a linear approximation that unifies the three core structured patterns:

$$S^{(t)} = \sum_{k=1}^{2n-1} c_k^{(t)} C_k + \sum_{k=1}^{n} d_k^{(t)} D_k + \sum_{k \in \mathcal{A}} e_k^{(t)} E_k + R^{(t)}, \tag{3}$$

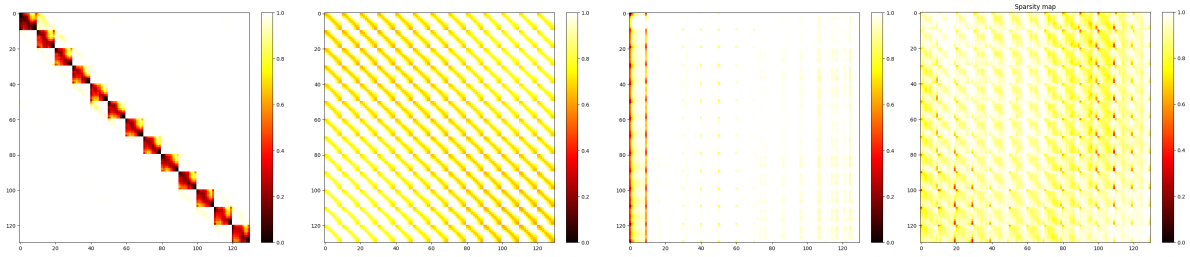
where $C_k, D_k, E_k \in \mathbb{R}^{n \times n}$ are binary basis matrices representing parallel-diagonal, vertical, and block-diagonal patterns, respectively; $c_k^{(t)}, d_k^{(t)}, e_k^{(t)} \in \mathbb{R}$ are intensity scalars quantifying each pattern's contribution at step t; $\mathcal{A}$ denotes the set of block-diagonal indices; and $R^{(t)} \in \mathbb{R}^{n \times n}$ captures the approximation residual.

Parallel-diagonal basis C_k (offset $\delta_k = k - (n-1)$, $\delta_k \in [-(n-1), n-1]$):

$$C_k(i,j) = \begin{cases} 1 & \text{if } j - i = \delta_k, \\ 0 & \text{otherwise} \end{cases} \qquad (i, j \in \{1, 2, ..., n\})$$

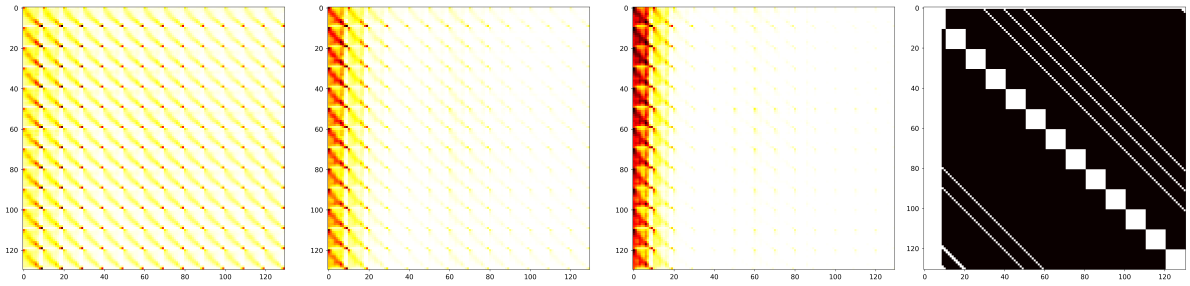
Vertical basis D_k**:**

$$D_k(i,j) = \begin{cases} 1 & \text{if } j = k, \\ 0 & \text{otherwise} \end{cases} \qquad (i, j \in \{1, 2, ..., n\})$$



(a) Block-diagonal pattern **(b) Parallel-to-main pattern** **(c) Vertical pattern** **(d) Mixture pattern**

Figure 2. Visualization of the four attention patterns in CogVideoX-v1.5(Yang et al., 2024).



(a) Step 12 **(b) Step21** **(c) Step31** **(d) Step 41**

Figure 3. Evolution of the Attention Sparsity Map in CogVideoX-v1.5(Yang et al., 2024) (Layer 0, Head 9) over Denoising Steps, which demonstrates significant differences in the attention sparsity patterns across denoising steps.

Block-diagonal basis E_k (let $B_k = [a_k, b_k] \times [a_k, b_k]$ denote the k-th block region):

$$E_k(i,j) = \begin{cases} 1 & \text{if } (i,j) \in B_k, \\ 0 & \text{otherwise} \end{cases} \qquad (i,j \in \{1,2,...,n\},\ k \in \mathcal{A})$$

To estimate the parameter vector $\mathbf{X}^{(t)} \in \mathbb{R}^{(3n-1+|\mathcal{A}|)\times 1}$:

$$\mathbf{X}^{(t)} = \left[c_1^{(t)}, \ldots, c_{2n-1}^{(t)}, d_1^{(t)}, \ldots, d_n^{(t)}, e_1^{(t)}, \ldots, e_{|\mathcal{A}|}^{(t)}\right]^T,$$

we construct the design matrix $M \in \mathbb{R}^{n^2 \times (3n-1+|\mathcal{A}|)}$:

$$M = \left[\{\text{vec}(C_k)\}_{k=1}^{2n-1}, \{\text{vec}(D_k)\}_{k=1}^{n}, \{\text{vec}(E_k)\}_{k=1}^{|\mathcal{A}|}\right],$$

where $\text{vec}(\cdot)$ denotes matrix vectorization. The optimization objective minimizes the residual norm:

$$\mathbf{X}^{(t)} = \arg\min_{\mathbf{X}} \left\| \text{vec}(S^{(t)}) - M\mathbf{X} \right\|_2^2. \qquad (4)$$

To verify the effectiveness and scalability of our generalized linear approximation model (Eq. (3)) in capturing the core structure of attention sparsity maps across different sequence lengths, we evaluate the approximation error using a normalized residual metric. For a given sequence length N, the **normalized approximation error** is defined as

$$NAE(N) = \frac{\|R^{(t)}\|_2}{\|S^{(t)}\|_2}$$

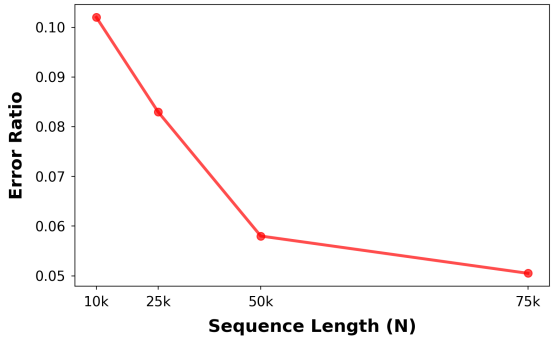


Figure 4. Normalized approximation error of linear approximation model (eq. (3)) across sequence lengths on hunyuan video.

As observed in Figure 4, the error remains consistently low across all tested sequence lengths. This demonstrates that our linear approximation model can accurately capture the core structural patterns of attention sparsity maps regardless of sequence length, validating its effectiveness and scalability. To ensure real-time inference, we implement a hardware-optimized kernel for this least-squares problem. See Appendix E for more details.

Remark 3.1. The least-squares acceleration kernel in the Appendix E accelerates solving by leveraging the sparsity and special structure of basis matrices to **decompose**

high-complexity matrix operations into low-dimensional structured computations, combined with GPU multi-level parallelism to optimize memory access and efficiency.

3.3. Pattern Intensity Evolution

Using the linear approximation model (Eq. (4)), we analyze the evolution of attention sparsity maps $S^{(t)}$ during full attention inference by solving for the intensity scalars of vertical and parallel-diagonal patterns across denoising steps. Given that the block-diagonal distribution pattern remains largely invariant during denoising, we will employ a static thresholding criterion (section 4.3) to characterize it, as opposed to a dynamic prediction approach. As shown in Figure 5, early denoising steps exhibit complex, non-linear intensity variations due to unstructured spatiotemporal dependencies. However, these intensities converge to approximately piecewise linear trends in mid-to-late steps.

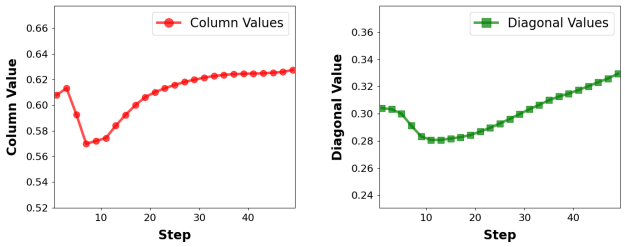


Figure 5. Evolution of vertical and parallel-diagonal pattern intensities across denoising steps, showing convergence to piecewise linearity.

This piecewise linearity enables predictive masking: by modeling $c_k^{(t)}$ and $d_k^{(t)}$ as linear functions of t, we design a pattern predictor (Section 4.2) that dynamically adjusts masks across denoising steps without costly sampling.

4. Method

We propose the **MOD-DiT** framework, with its workflow illustrated in Figure 6 and the detailed algorithm provided in Appendix 1, whose design is strictly guided by the empirical insights in Section 3. To realize the dynamic sparse attention mechanism of MOD-DiT, we construct the framework through three core steps: sparsity map reconstruction, linear pattern prediction, and dynamic mask generation. These steps work in synergy to model the dynamic evolution of attention patterns in video diffusion transformers, enabling sampling-free and high-accuracy sparse attention for efficient video generation.

Key Notations. We define: t as the denoising step, $t_p^{(i)} = m + i \cdot \Delta t$ ($i \geq 0$) as the linear prediction step with interval Δt, and $S^{(t)} \in \mathbb{R}^{n \times n}$ as the attention sparsity map.

Warm-Up with Full Attention. As shown in Figure 5, in the early denoising stage, the sparsity map $S^{(t)}$ exhibits **unstructured spatiotemporal dependencies**, making direct application of sparse attention via Eq.(1) prone to irreversible information loss. To address this, we perform warm-up via m steps of full attention. Specifically, we set $m = 12$ in our experimental settings to capture the initial unstructured dependencies with full attention. See Appendix B for more details.

4.1. Sparsity Map Reconstruction

To extract the three structural patterns (Figure 2) via the linear approximation model (3), a complete attention map is required. However, in practice, sparse attention (Eq.(1)) only yields an incomplete attention map $A_{\text{masked}}^{(t_p^{(i)})}$, making direct pattern extraction impossible. We thus need to reconstruct the complete attention map from the masked version.

The reconstruction follows an iterative temporal fusion strategy, where the complete attention map at step $t_p^{(i+1)}$ (denoted $\hat{A}^{(t_p^{(i+1)})}$) is formed by combining the masked attention map at $t_p^{(i+1)}$ and the reconstructed complete attention map at step $t_p^{(i)}$.

$$\hat{A}^{(t_p^{(i+1)})}(p,q) = \begin{cases} A_{\text{masked}}^{(t_p^{(i+1)})}(p,q), & M^{(t_p^{(i+1)})}(p,q) = 0 \\ \hat{A}^{(t_p^{(i)})}(p,q), & M^{(t_p^{(i+1)})}(p,q) = -\infty \end{cases} \tag{5}$$

where $\hat{A}^{(t_p^{(0)})} = A^{(m)}$, $A^{(m)}$ is the complete attention map from the final warm-up step ($t = m$). Notably, performing this reconstruction every Δt steps integrates sparse attention information into the denoising process, which helps correct cumulative information bias induced by subsequent pattern predictions. We further convert the reconstructed complete attention map $\hat{A}^{(t)}$ into the sparsity map $\hat{S}^{(t)}$ via Eq.(2) for pattern intensity estimation, ensuring consistency with the linear approximation model. Ablation confirm that this reconstruction operation introduces minimal errors. See Appendix B for more details.

4.2. Linear Prediction of Vertical & Parallel Pattern

As observed in Figure 5, the intensity scalars of parallel-diagonal ($c_k^{(t)}$) and vertical-line ($d_k^{(t)}$) patterns in attention sparsity maps exhibit stable piecewise linearity during the mid-late denoising stage (i.e., $t > m$), a property validated by our systematic ablation study with 300 data points (Appendix B.1.7). Leveraging this regularity, we design a lightweight linear prediction method to generate sparsity maps for subsequent denoising steps.

Prediction Principle and Formulation. For any three consecutive prediction steps $t_p^{(i)}$, $t_p^{(i+1)}$, and $t_p^{(i+2)}$, we use the reconstructed sparsity maps $\hat{S}^{(t_p^{(i)})}$ and $\hat{S}^{(t_p^{(i+1)})}$ to linearly

predict the intensity scalars $\hat{c}_k^{(t)}$ and $\hat{d}_k^{(t)}$ for all denoising steps $t \in (t_p^{(i+1)}, t_p^{(i+2)}]$.

$$\hat{c}_k^{(t)} = \hat{c}_k^{(t_p^{(i+1)})} + \frac{\hat{c}_k^{(t_p^{(i+1)})} - \hat{c}_k^{(t_p^{(i)})}}{t_p^{(i+1)} - t_p^{(i)}} \cdot \left(t - t_p^{(i+1)}\right)$$

$$\hat{d}_k^{(t)} = \hat{d}_k^{(t_p^{(i+1)})} + \frac{\hat{d}_k^{(t_p^{(i+1)})} - \hat{d}_k^{(t_p^{(i)})}}{t_p^{(i+1)} - t_p^{(i)}} \cdot \left(t - t_p^{(i+1)}\right)$$

where $\hat{c}_k^{(t_p^{(i)})}$, $\hat{d}_k^{(t_p^{(i)})}$ are extracted from $\hat{S}^{(t_p^{(i)})}$; $\hat{c}_k^{(t_p^{(i+1)})}$, $\hat{d}_k^{(t_p^{(i+1)})}$ are derived from $\hat{S}^{(t_p^{(i+1)})}$ (via Eq. (3)).

This piecewise linear prediction is critical for quality preservation. Ablation shows static one-time prediction degrades performance significantly, confirming the necessity of dynamic tracking (see Appendix B for details).

4.3. Dynamic Mask Generation

We design a dynamic mask generation strategy that integrates pattern-based Top-K selection and conditional block-diagonal preservation. For each attention head h and denoising step t, the mask $M_h^{(t)}(p, q)$ is defined as

$$M_h^{(t)}(p,q) = \begin{cases} 0, & (p,q) \in \mathcal{K}_t \cup \mathcal{K}_E \\ -\infty, & \text{otherwise} \end{cases}$$

where $\mathcal{K}_t$ is the Top-K retained set of informative patterns, constructed by merging parallel-diagonal (C_k) and vertical-line (D_k) patterns, sorting their intensity scalars $\{\hat{c}_k^{(t)}\} \cup \{\hat{d}_k^{(t)}\}$ in ascending order, and selecting the top K patterns. $\mathcal{K}_E = \text{supp}(E)$ denotes the support set of the block-diagonal pattern, which is preserved if and only if $e_{\min} = \min\left\{\hat{e}^{(m-1)}, \hat{e}^{(m)}\right\} > \tau_e$ (τ_e is a threshold). Notably, the block-diagonal pattern characterizes intra-frame token interactions with fixed spatial size and position across denoising steps (each block maps to a single frame's token region). Unlike the dynamically evolving parallel-to-main-diagonal and vertical patterns that require linear prediction for intensity tracking, it only needs thresholding to verify existence.

The final layer-wise mask $M^{(t)}$ is constructed by concatenating masks across all H attention heads

$$M^{(t)} = \text{cat}\left(M_1^{(t)}, M_2^{(t)}, \ldots, M_H^{(t)}\right).$$

4.4. Hardware Acceleration

To fully leverage the efficiency advantages of MOD-DiT, we designed three targeted optimization strategies.

Optimized Least-Squares Kernel. We designed an optimized least-squares kernel based on CUDA, reducing the solving time of E.q (4) by 100× compared to native solvers 'torch.lstsq' in PyTorch. See Appendix D for more details.

Hybrid Attention Execution. We use FlashAttention-2(Dao, 2023) in the warm-up stage to minimize the computational overhead of full attention; in the sparse stage, we switch to SageAttention (Zhang et al., 2025a), which directly skips invalid token interactions at the hardware level, further reducing redundant computations and improving execution speed.

Block-wise Attention Computation. We adopt a block-wise strategy for attention calculation acceleration(Jiang et al., 2024): we partition query (Q) and key (K) tensors into non-overlapping blocks with a fixed block size of 128 via e.q (2). This block-wise execution enables localized memory access and leverages GPU warp-level parallelism to process intra-block token interactions in batches, further lowering the latency of attention computation.

Remark 4.1. The additional computational operations introduced by the **MOD-DiT** framework are negligible compared to the cost of full attention. Experimental measurements on VBench(Huang et al., 2023) confirm that the total overhead of these operations accounts for only 1-2% of the full attention computation time. See Appendix E for more details.

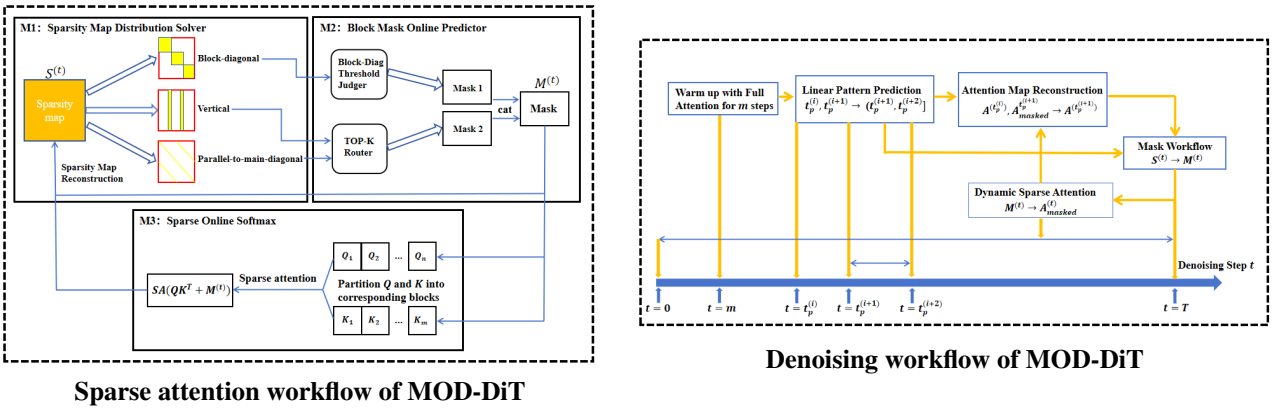
5. Experiments

5.1. Experimental Settings

Models. We evaluate MOD-DiT both on popular small and large video generation models including CogVideoX-v1.5(Yang et al., 2024), HunyuanVideo(Kong et al., 2024) and Wan2.1(Wang et al., 2025), which contain 2b, 13b and 14b parameters respectively. After 3D VAE, CogVideoX-v1.5 consumes 23 frames with 1536 tokens per frame in 3D full attention, while HunyuanVideo works on 33 frames with 2944 tokens per frame, and Wan2.1 works on 23 frames with 4096 tokens per frame.

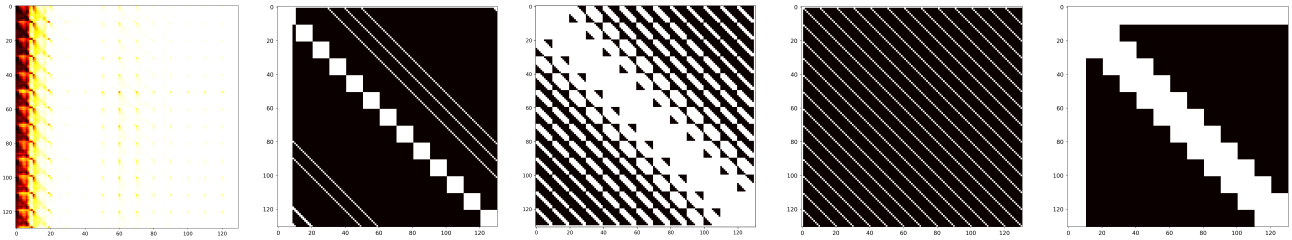
Baselines. We compare MOD-DiT with latest sparse attention methods:

- **SVG(Xi et al., 2025)**: Accelerates video models with sparse attention by dynamically classifying attention heads as spatial or temporal and applying corresponding masks.
- **Radial(Li et al., 2025)**: Accelerates video models by Constructing a static mask with Spatiotemporal Energy Decay in video generation.
- **MInference(Jiang et al., 2024)**: A classical sparse acceleration technique migrated from large language models. which choose different sparse patterns for different heads.



Sparse attention workflow of MOD-DiT | **Denoising workflow of MOD-DiT**

Figure 6. Workflow of MOD-DiT



Sparsity map | **MOD-DiT mask** | **Radial mask** | **Temporal mask** | **Spatial mask**

Figure 7. Visual comparison of various sparse attention methods (Radial attention(Li et al., 2025), SVG(Xi et al., 2025), and MOD-DiT) generate corresponding masks for specific attention heads during the inference of CogVideo(Yang et al., 2024).

- **SpargeAttn(Zhang et al., 2025b)**: Accelerates attention calculation by pruning blocks and Online softmax threshold filtering.
- **LiteAttention(Shmilovich et al., 2025)**:An efficient sparse attention mechanism that leverages temporal coherence between denoising steps by early identifying and propagating computationally skipable blocks.

Datasets. We use the VBench(Huang et al., 2023) dataset to rate MOD-DiT and other baselines, which demonstrate the quality of generated videos across multiple dimensions.

Metrics. We employ a comprehensive set of metrics to evaluate the generated videos. The overall video quality is assessed using the VBench score, with a specific focus on its imaging quality and subject consistency dimensions. Additionally, we measure visual fidelity, similarity to ground truth, and perceptual differences between the outputs of the original models and the benchmarked methods using Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) for quantifying numerical similarity, as well as Learned Perceptual Image Patch Similarity (LPIPS) for assessing perceptual differences. For sparsity evaluation, we define **sparsity** as the ratio of blocks to be computed to the total number of blocks, where blocks are partitioned into non-overlapping $B \times B$ sizes.

5.2. Experimental Results Analysis

Impact of Sequence Length. We evaluated the impact of sequence length on inference time using the Hunyuan model (Kong et al., 2024), adjusting Top-K hyperparameter of MOD-DiT accordingly. As shown in Figure 8, the acceleration achieved by MOD-DiT becomes more substantial than Full Attention as the sequence length increases.

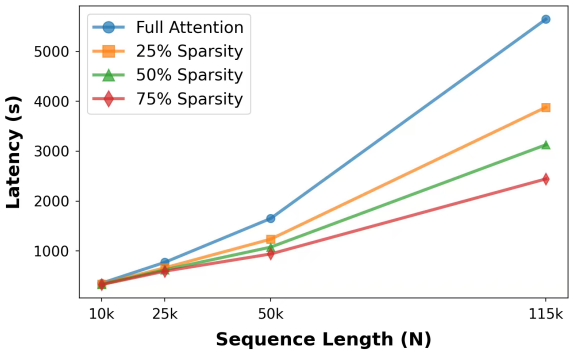


Figure 8. Comparison of inference time between Full Attention and MOD-DiT under varying sequence lengths

Quality Evaluation. We evaluate the performance of various methods on two mainstream video generation models—CogVideoX-v1.5 and Hunyuan Video—across

Table 1. Quantitative comparison on the CogVideoX-v1.5(Yang et al., 2024), HunyuanVideo(Kong et al., 2024), and Wan2.1(Wang et al., 2025) benchmarks. We use VBench prompts for inference. For CogVideoX-v1.5 (2B parameters), all methods are evaluated on 89-frame videos at 640×512 resolution with experiments on NVIDIA A800 80G GPUs. For HunyuanVideo (13B parameters), evaluations are conducted on 129-frame videos at 768×896 resolution using NVIDIA H800 80G GPUs. For Wan2.1 (14B parameters), tests are performed on 81-frame videos at 512×832 resolution with NVIDIA A100 80G GPUs. Notably, MInference, Radial, SVG, SpargeAttn, and our method all utilize the same API to ensure fair comparison. Our method outperforms others in quality (PSNR, SSIM, etc.) and efficiency, achieving maximum speedup with the highest scores while maintaining the **highest sparsity**.

Model	Method	Sparsity	Quality					Efficiency	
			PSNR↑	SSIM↑	LPIPS↓	SubConsist↑	ImageQual↑	Latency(s)↓	Speedup↑
CogVideoX-v1.5	Full	0.00%	-	-	-	0.9230	**0.6255**	987s	1×
	MInference	64.9%	15.01	0.601	0.334	0.8679	0.5580	696s	1.42×
	Radial	70.7%	22.89	0.866	0.172	0.9214	0.6167	611s	1.62×
	SVG	75.0%	21.15	0.818	0.183	0.9158	0.5948	596s	1.65×
	Sparge	67.3%	20.34	0.773	0.255	0.9043	0.5966	661s	1.49×
	Ours	**80.1%**	**25.77**	**0.868**	**0.133**	**0.9266**	0.6239	**542s**	**1.82×**
Hunyuan Video	Full	0.00%	-	-	-	**0.9701**	0.6396	1980s	1×
	MInference	68.2%	18.21	0.638	0.490	0.8867	0.6021	1365s	1.45×
	Radial	76.3%	26.72	**0.885**	0.125	0.9533	0.6415	993s	1.99×
	SVG	79.7%	26.44	0.861	0.170	0.9158	0.6072	903s	2.19×
	Sparge	70.7%	25.43	0.842	0.195	0.9092	0.6125	1125s	1.76×
	Ours	**87.8%**	**27.23**	0.879	**0.119**	0.9690	**0.6533**	**865s**	**2.29×**
Wan2.1	Full	0.00%	-	-	-	**0.9511**	**0.6582**	1002s	1×
	MInference	62.5%	15.81	0.675	0.343	0.9002	0.5987	701s	1.43×
	Radial	73.6%	21.57	0.818	0.167	0.9420	0.6513	562s	1.78×
	SVG	75.1%	20.93	0.795	0.222	0.9377	0.6351	573s	1.75×
	Sparge	72.7%	18.67	0.735	0.198	0.9454	0.6480	612s	1.63×
	LiteAttention	43.8%	22.67	0.815	**0.148**	0.9484	0.6521	547s	1.83×
	Ours	**78.0%**	**22.75**	**0.821**	0.152	0.9497	0.6532	**511s**	**1.96×**

three core dimensions: Sparsity, Quality, and Efficiency. Quality metrics include Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), Learned Perceptual Image Patch Similarity (LPIPS)(Zhang et al., 2018), Subject Consistency (SubConsist), and Imaging Quality (ImageQual). Efficiency is quantified by Latency and Speedup. For warm-up, we use 12 full-attention steps, retaining sparsity maps from the final two steps for subsequent prediction. Thereafter, sparsity maps are computed every 10 steps ($\Delta t = 10$). It is worth noting that MOD-DiT and all baseline methods (MInference, Radial, SVG, and Sparge) are **block-level sparse attention** approaches, ensuring strict fairness in the comparative evaluation. Across all three models, MOD-DiT achieves the highest sparsity among all baseline methods, while comprehensively outperforming MInference, Radial, SVG, Sparge and LiteAttention in both quality and efficiency. When comparing under practical inference scenarios consistent with benchmark settings, MOD-DiT delivers the lowest latency and highest speedup, while leading in all quality metrics, fully confirming its unparalleled superiority in the quality-efficiency tradeoff.

Ablation Study. We also conduct ablation and validation experiments, including those on top-k selection, sparsity construction threshold η, reconstruction interval Δt, reconstruction Error, warm up steps m, sparsity map dynamic deviation, and linearity of pattern intensity scalars, with details presented in the Appendix B.

Accuracy of Pattern Capturing. As illustrated in Figure 7, the mask generated by MOD-DiT more accurately captures the structural features of the attention sparsity map compared to other methods, which aligns with the dynamic mixture of attention patterns in video diffusion transformers.

6. Conclusion and limitation

Conclusions. MOD-DiT is a training-free dynamic sparse attention framework designed to mitigate the quadratic complexity of self-attention in video diffusion transformers. It identifies three core attention patterns—block-diagonal, parallel-to-main-diagonal, and vertical—that dynamically mix and evolve, modeled via a sampling-free linear prediction algorithm. Seamlessly adapting to existing vDiTs without retraining, it delivers significant inference speedups while preserving high video quality across multiple models and benchmarks. For future work, we plan to combine MOD-DiT with complementary efficiency techniques such as model quantization and feature caching to further push the boundaries of efficient high-quality video synthesis.

Limitations. MOD-DiT incorporates a short full-attention warm-up phase to capture initial unstructured spatiotemporal dependencies, but this may introduce additional computational overhead in specific short-sequence scenarios.

References

Beltagy, I., Peters, M. E., and Cohan, A. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S. W., Fidler, S., and Kreis, K. Stable video diffusion: Scaling latent video diffusion models to large datasets. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Chen, H., Zhang, Y., Cun, X., Xia, M., Wang, X., Weng, C., and Shan, Y. Videocrafter2: Overcoming data limitations for high-quality video diffusion models, 2024. URL https://arxiv.org/abs/2401.09047.

Chen, P., Zeng, X., Zhao, M., Ye, P., Shen, M., Cheng, W., Yu, G., and Chen, T. Sparse-vdit: Unleashing the power of sparse attention to accelerate video diffusion transformers. *arXiv preprint arXiv:2506.03065*, 2025.

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

Guo, C. et al. Sparse transformers for efficient video understanding. *IEEE International Conference on Computer Vision (ICCV)*, 2021.

Guo, Y., Zheng, Y., Tan, M., Chen, Q., Chen, J., Zhao, P., and Huang, J. Nat: Neural architecture transformer for accurate and compact architectures, 2020. URL https://arxiv.org/abs/1910.14488.

Guo, Y., Yang, C., Rao, A., Wang, Y., Qiao, Y., Lin, B., and Dai, X. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2401.13715*, 2024.

Henschel, R., Khachatryan, L., Poghosyan, H., Hayrapetyan, D., Tadevosyan, V., Wang, Z., Navasardyan, S., and Shi, H. Streamingt2v: Consistent, dynamic, and extendable long video generation from text, 2025. URL https://arxiv.org/abs/2403.14773.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., Wang, Y., Chen, X., Wang, L., Lin, D., Qiao, Y., and Liu, Z. Vbench: Comprehensive benchmark suite for video generative models, 2023.

Jiang, H., Li, Y., Zhang, C., Wu, Q., Luo, X., Ahn, S., Han, Z., Abdi, A. H., Li, D., Lin, C.-Y., Yang, Y., and Qiu, L. Mlinference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention, 2024. URL https://arxiv.org/abs/2407.02490.

Kahatapitiya, K., Liu, H., He, S., Liu, D., Jia, M., Zhang, C., Ryoo, M. S., and Xie, T. Adaptive caching for faster video generation with diffusion transformers, 2024a. URL https://arxiv.org/abs/2411.02397.

Kahatapitiya, K., Liu, H., He, S., Liu, D., Jia, M., Zhang, C., Ryoo, M. S., and Xie, T. Adaptive caching for faster video generation with diffusion transformers, 2024b. URL https://arxiv.org/abs/2411.02397.

Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Lai, X., Lu, J., Luo, Y., Ma, Y., and Zhou, X. Flexprefill: A context-aware sparse attention mechanism for efficient long-sequence inference, 2025. URL https://arxiv.org/abs/2502.20766.

Li, X., Li, M., Cai, T., Xi, H., Yang, S., Lin, Y., Zhang, L., Yang, S., Hu, J., Peng, K., et al. Radial attention: $\mathcal{O}(n \log n)$ sparse attention with energy decay for long video generation. *arXiv preprint arXiv:2506.19852*, 2025.

Liu, F., Zhang, S., Wang, X., Wei, Y., Qiu, H., Zhao, Y., Zhang, Y., Ye, Q., and Wan, F. Timestep embedding tells: It's time to cache for video diffusion model, 2025.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Lv, Z., Si, C., Song, J., Yang, Z., Qiao, Y., Liu, Z., and Wong, K.-Y. K. Fastercache: Training-free video diffusion model acceleration with high quality, 2025. URL https://arxiv.org/abs/2410.19355.

Ma, J., Peng, Q., Zhu, X., Xie, P., Chen, C., and Lu, H. Pluggable pruning with contiguous layer distillation for diffusion transformers, 2025. URL https://arxiv.org/abs/2511.16156.

Melnik, A., Ljubijanac, M., Lu, C., Yan, Q., Ren, W., and Ritter, H. Video diffusion models: A survey, 2024. URL https://arxiv.org/abs/2405.03150.

Park, G. Y., Lee, S. W., and Ye, J. C. Inference-time diffusion model distillation, 2024. URL https://arxiv.org/abs/2412.08871.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Qi, C., Cun, X., Zhang, Y., Lei, X., Wang, X., Shan, Y., and Chen, Q. Fatezero: Fusing attentions for zero-shot text-based video editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15932–15942, 2023.

Qiu, J., Liu, L., Wang, S., Lu, J., Chen, K., and Hao, Y. Accelerating diffusion transformer via gradient-optimized cache, 2025. URL https://arxiv.org/abs/2503.05156.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022.

Shmilovich, D., Wu, T., Dahan, A., and Domb, Y. Liteattention: A temporal sparse attention for diffusion transformers. *arXiv preprint*, nov 2025. URL https://arxiv.org/abs/2511.11062. Submitted on 14 November 2025.

Tan, X., Chen, Y., Jiang, Y., Chen, X., Yan, K., Duan, N., Zhu, Y., Jiang, D., and Xu, H. Dsv: Exploiting dynamic sparsity to accelerate large-scale video dit training, 2025. URL https://arxiv.org/abs/2502.07590.

Team, O. Opensora: Democratizing efficient video generation for all. *arXiv preprint arXiv:2407.11455*, 2024.

Tian, S., Chen, H., Lv, C., Liu, Y., Guo, J., Liu, X., Li, S., Yang, H., and Xie, T. Qvd: Post-training quantization for video diffusion models. In *Proceedings of the 32nd ACM International Conference on Multimedia (MM '24)*, pp. 10572–10581. ACM, 2024. doi: 10.1145/3664647.3681050. URL https://doi.org/10.1145/3664647.3681050.

Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.-W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Xi, H., Yang, S., Zhao, Y., Xu, C., Li, M., Li, X., Lin, Y., Cai, H., Zhang, J., Li, D., Chen, J., Stoica, I., Keutzer, K., and Han, S. Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity. *arXiv preprint arXiv:2502.01776*, 2025.

Xia, Y., Ling, S., Fu, F., Wang, Y., Li, H., Xiao, X., and Cui, B. Training-free and adaptive sparse attention for efficient long video generation, 2025.

Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

Zeng, Q., Hu, C., Song, M., and Song, J. Diffusion model quantization: A review, 2025. URL https://arxiv.org/abs/2505.05215.

Zhang, J., Wei, J., Huang, H., Zhang, P., Zhu, J., and Chen, J. Sageattention: Accurate 8-bit attention for plug-and-play inference acceleration. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025a. URL https://openreview.net/forum?id=OL44KtasKc. Published as a conference paper at ICLR 2025; arXiv:2410.02367.

Zhang, J., Xiang, C., Huang, H., Wei, J., Xi, H., Zhu, J., and Chen, J. Spargeattention: Accurate and training-free sparse attention accelerating any model inference. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025b. URL https://github.com/thu-ml/SpargeAttn. ICML 2025.

Zhang, P., Chen, Y., Su, R., Ding, H., Stoica, I., Liu, Z., and Zhang, H. Fast video generation with sliding tile attention, 2025c.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

A. Related work

Video Diffusion Models and Transformers. Diffusion models have achieved state-of-the-art results in image synthesis (Ho et al., 2020), prompting their extension to video generation. Early approaches employed decoupled spatial and temporal attention (Blattmann et al., 2023), while recent advances adopt 3D dense attention (Melnik et al., 2024; Peebles & Xie, 2023) to jointly model spatio-temporal dynamics and capture long-range dependencies. Leading models including OpenSora (Team, 2024), CogVideoX (Yang et al., 2024), HunyuanVideo (Kong et al., 2024), and Wan2.1 (Wang et al., 2025) now support diverse applications spanning animation, editing, and long-video generation (Qi et al., 2023; Chen et al., 2024; Henschel et al., 2025). However, the quadratic complexity of self-attention remains a fundamental bottleneck.

Sparse Attention for Efficient Video Diffusion Transformers. Sparse attention methods address computational costs by restricting token interactions through either static or dynamic designs. Static approaches employ fixed sparsity patterns: Sparse-vDiT (Chen et al., 2025) predefines vertical and block-diagonal patterns, Radial Attention (Li et al., 2025) introduces exponential decay achieving $\mathcal{O}(n \log n)$ complexity, while Sliding Tile Attention (Zhang et al., 2025c) utilize fixed 3D windows and distance-based restrictions. In contrast, dynamic methods adapt patterns based on input characteristics: Sparse VideoGen (Xi et al., 2025) classifies attention heads as spatial or temporal through online sampling, MInference (Jiang et al., 2024) explores hybrid static-dynamic patterns, AdaSpa (Xia et al., 2025) captures hierarchical sparsity via blockified dynamic patterns while achieving training-free acceleration with LSE-cached online search, SpargeAttention (Zhang et al., 2025b) emphasizes cross-modal compatibility with adaptive structures, and LiteAttention (Shmilovich et al., 2025) leverages temporal coherence across denoising steps to early identify and propagate computationally skipable blocks, reducing redundant computations through block-level skipping. Specifically, SpargeAttention primarily relies on query and key features, while AdaSpa extracts signals from attention block values—neither leverages the critical observation that attention maps inherently manifest as superpositions of structured patterns with gradual evolution. This oversight leads to suboptimal feature extraction efficiency and accuracy. Despite these advances, existing methods fail to capture the **true dynamic mixture of attention patterns** in vDiTs, motivating our approach.

Efficient Diffusion and Attention Mechanisms. Complementary efficiency techniques for diffusion models include pruning to reduce parameters (Ma et al., 2025), quantization to lower bit-width overhead (Zeng et al., 2025; Tian et al., 2024), and caching strategies that trade memory for speed (Qiu et al., 2025; Liu et al., 2025; Kahatapitiya et al., 2024a). Video-specific methods— TeaCache (Liu et al., 2025), FasterCache (Lv et al., 2025), and AdaCache (Kahatapitiya et al., 2024b)—exploit temporal redundancy by reusing features across adjacent denoising steps, while distillation approaches reduce inference timesteps (Park et al., 2024) and high-ratio VAEs compress latent representations. In broader attention research, sparse patterns have been extensively explored: vision models including Swin Transformer (Liu et al., 2021), NAT (Guo et al., 2020), and Sparse Transformers (Guo et al., 2021) employ local windowing, while NLP models like Longformer (Beltagy et al., 2020) utilize sliding windows. Recent advances such as MInference (Jiang et al., 2024) and FlexPrefill (Lai et al., 2025) have explored diverse sparsity patterns for efficient attention computation, yet few are tailored to the unique spatiotemporal dynamics and denoising step dependencies of video diffusion transformers.

B. Additional Experiments

B.1. Ablation Study

B.1.1. Ablation study on the Top K hyperparameter

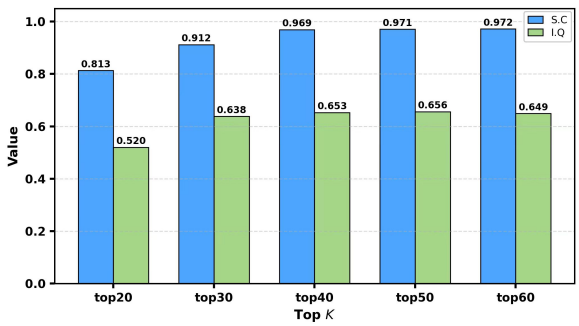


Figure 9. Ablation study on the Top K hyperparameter.

Figure 9 shows the Top K ablation on the Hunyuan(Kong et al., 2024) model. subject Consistency (S.C.) and Image Quality (I.Q.) initially increase with K and then stabilize, indicating its performance sensitivity.

B.1.2. Ablation study on sparsity threshold η

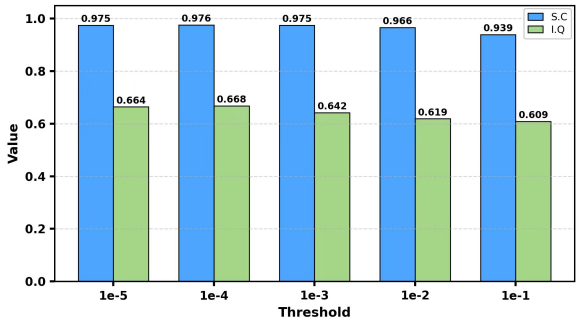


Figure 10. Ablation study on sparsity threshold η in Eq.(2).

We evaluated the performance of MOD-DiT by adjusting the sparsity calculation threshold η. As shown in Figure 10, the results indicate that the threshold has a negligible impact on performance, with only a slight degradation occurring when the threshold is set to an extremely large value. Based on these findings, we selected a threshold of $1e-4$ for MOD-DiT.

B.1.3. Ablation study on dynamic evolution of attention sparsity maps

To further validate the dynamic evolution of attention sparsity maps and justify the necessity of modeling time-varying patterns (addressed in Q1), we conduct an additional experiment to quantify the structural deviation of sparsity maps across denoising steps. Specifically, we fix the warm-up steps to $m = 12$, and define the **difference error ratio** between the sparsity map at step $t \geq 12$ (post warm-up) and the map at the end of warm-up (step 12) as

$$DER(t) = \frac{\|S^{(t)} - S^{(12)}\|_2}{\|S^{(12)}\|_2}$$

where $S^{(t)}$ denotes the attention sparsity map at denoising step t. This metric quantifies how much the sparsity map deviates from its warm-up-converged state—higher values indicate more significant structural changes. As observed in Figure 11, the difference error ratio of most heads exhibits a certain degree of fluctuation across denoising steps. This confirms that the attention sparsity map undergoes substantial structural evolution even after the warm-up phase, rather than remaining static.

B.1.4. Ablation study on Reconstruction Interval

To verify the robustness of MOD-DiT to the reconstruction interval Δt(defined in Section 4.1) and the effectiveness of our segment-wise prediction, we conduct ablation experiments with varying $\Delta t(1, 2, 3, 5, 10)$ and an additional "None" case—where only one pattern prediction is performed post-warm-up without subsequent mask updates. Experiments use evaluation metrics including Subject Consistency (SubConsist) and Imaging Quality (ImageQual).

Table 2. Ablation results of reconstruction interval Δt on CogVideoX-v1.5. Experiments are conducted with 89-frame videos at 640×512 resolution using VBench prompts, on NVIDIA A100 80G GPUs.

Δt	**SubConsis**	**ImageQual**
1	0.925	0.623
2	0.926	0.622
3	0.920	0.621
5	0.926	0.623
10	0.920	0.610
None	0.90	0.57

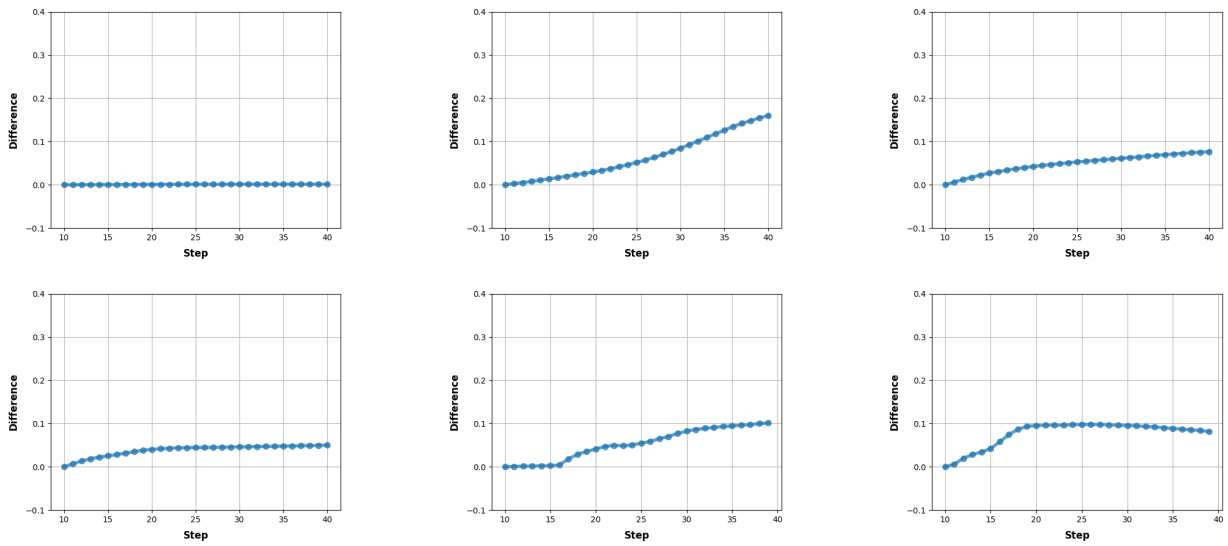


Figure 11. Normalized reconstruction error (NRE) results for 6 randomly sampled attention heads across varying denoising steps. Experiments are conducted on CogVideoX-v1.5 with 49-frame 640×512 videos on NVIDIA A100 80G GPUs.

As observed in table 2, all tested Δt values yield nearly identical performance, which confirms that MOD-DiT is robust to the reconstruction interval. In contrast, the "None" case shows a significant performance drop, validating the necessity of segment-wise prediction. By periodically reconstructing sparsity maps and updating masks, MOD-DiT dynamically tracks evolving attention patterns, ensuring high spatio-temporal coherence and visual quality.

B.1.5. Ablation study on warm-up steps m

To determine the optimal number of warm-up steps for MOD-DiT, we conduct an ablation experiment focusing on the impact of m on video generation quality. The warm-up phase is critical for capturing initial unstructured spatiotemporal dependencies (Section 4), and an appropriate m ensures sufficient information retention without unnecessary computational overhead.

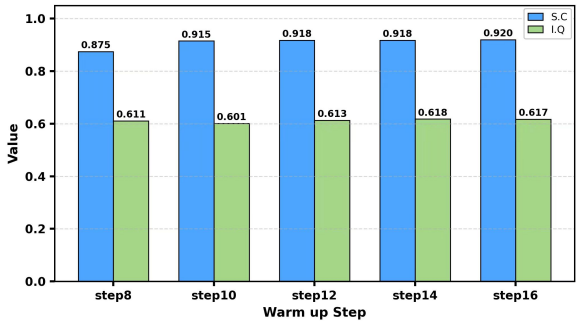


Figure 12. Ablation results of warm-up steps m on CogVideoX-V1.5. Experiments are conducted with 89-frame videos at 640×512 resolution using VBench prompts, on NVIDIA A100 80G GPUs. We use Subject Consistency (SubConsist) and Imaging Quality (ImageQual) as evaluation metrics.

As observed in Figure 12 , excessive warm-up steps introduce redundant computational overhead without meaningful quality improvements. When m=12, MOD-DiT achieves near-optimal S.C. and I.Q. while avoiding unnecessary full-attention costs. Thus, we select 12 warm-up steps as the default setting.

B.1.6. Ablation Study on Reconstruction Error

To verify that the sparsity map reconstruction (Section 4.1) does not introduce significant errors that degrade pattern prediction accuracy, we analyze the difference between the reconstructed sparsity map from MOD-DiT and the ground-truth sparsity map from full attention across layers, attention heads, and denoising steps. We define the **normalized reconstruction error** (NRE) at step t as

$$NRE(t) = \frac{\|\hat{S}^{(t)} - S_{GT}^{(t)}\|_2}{\|S_{GT}^{(t)}\|_2}$$

where $S_{GT}^{(t)}$ and $\hat{S}^{(t)}$ denotes the ground-truth sparsity map and the sparsity map reconstructed by MOD-DiT at denoising step t respectively.

As shown in Figure 13, all sampled attention heads exhibit a stable and low normalized reconstruction error across denoising steps. This directly confirms that MOD-DiT introduces minimal errors when reconstructing attention maps via E.q (5).

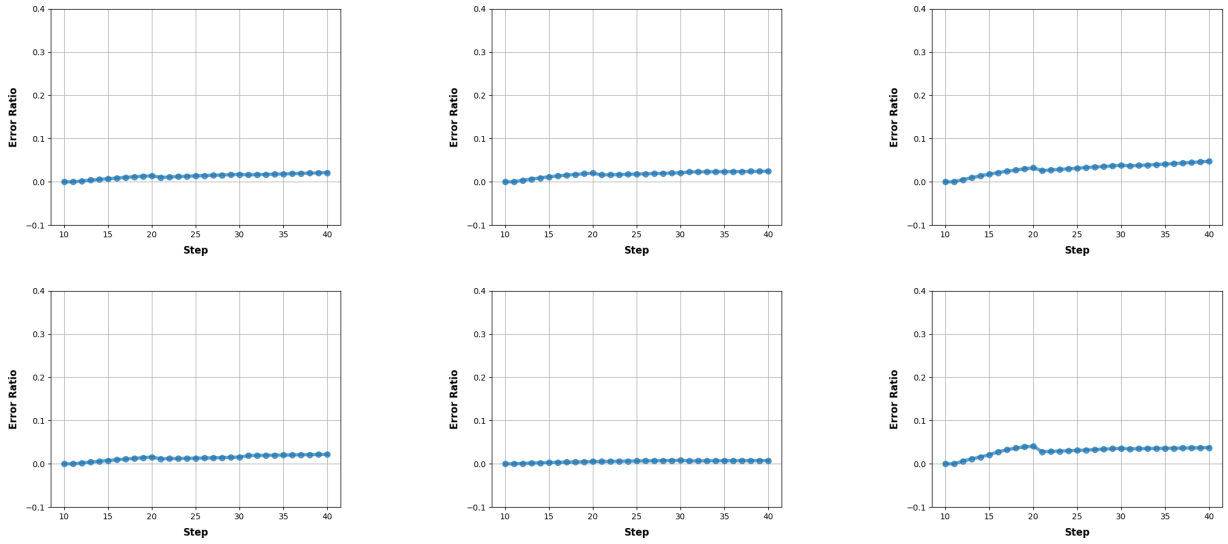


Figure 13. Normalized reconstruction error (NRE) results for 6 randomly sampled attention heads across varying denoising steps. Experiments are conducted under the same VBench prompt using CogVideoX-v1.5 with 49-frame 640×512 videos on NVIDIA A100 80G GPUs.

B.1.7. Ablation Study on Piecewise Linearity of Vertical and Parallel-to-Main-Diagonal Patterns

To validate the core observation in section 4.2 that the intensities of vertical ($d_k^{(t)}$) and parallel-to-main-diagonal ($c_k^{(t)}$) patterns exhibit stable piecewise linearity in the mid-to-late denoising stage ($t > m$, $m = 12$), we conduct a systematic statistical analysis with large-scale data points.

Experimental Setup

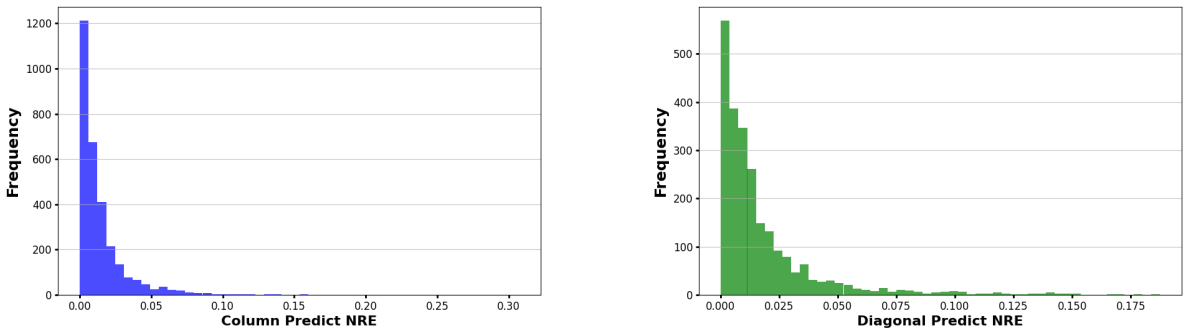
- **Model**: CogVideoX-v1.5
- **Data**: 5 diverse prompts randomly selected from VBench (Huang et al., 2023), covering different motion types and scenes, with 89-frame videos (640×512 resolution) generated for each prompt.
- **Attention Heads/Layers**: 10 layers randomly sampled, 6 attention heads per layer, resulting in $5 \times 10 \times 6 = 300$ independent data points.
- **Data Extraction**: For each prompt, layer, and head, we extract the intensity sequences of vertical/parallel-to-main-diagonal patterns in the mid-to-late denoising stage ($t \in [13, 50]$), and compute the normalized residual error (NRE) of piecewise linear fitting.

Evaluation Metric We define the *Normalized Residual Error (NRE)* to quantify the linear fitting accuracy, which measures the deviation between the true intensity and the fitted value:

$$\text{NRE} = \frac{\sqrt{\frac{1}{N_t}\sum_{t\in\text{segment}}(x_k^{(t)} - \hat{x}_k^{(t)})^2}}{\max(x_k^{(t)}) - \min(x_k^{(t)})}$$

where $x_k^{(t)}$ denotes the true intensity obtained by equation (4)(vertical $d_k^{(t)}$ or parallel-to-main-diagonal $c_k^{(t)}$), $\hat{x}_k^{(t)}$ is the value obtained by linear prediction, and N_t is the number of denoising steps in the segment. A smaller NRE indicates stronger linearity; we define NRE < 0.1 as "strong linearity".

Results and Analysis As observed in Figure 14, the NRE distribution of 300 data points almost all fall within the interval [0.00, 0.10], which confirm that the piecewise linearity of vertical and parallel-to-main-diagonal patterns is not an isolated case but a *universal property* across multiple prompts, layers, and heads in the CogVideo model.



(a)Vertical patterns **(b)Parallel-to-main-diagonal patterns**

Figure 14. Histograms of NRE for vertical and parallel-to-main-diagonal patterns, which are based on 300 data points (5 prompts × 10 layers × 6 heads).

B.2. Video Generation Results of MOD-DiT

To visually validate MOD-DiT's ability to balance generation quality and inference efficiency across diverse inputs and models, Figures 15, 16, and 17 show qualitative comparisons of its visualization results against other sparse attention methods (e.g., SVG(Xi et al., 2025), Radial Attention(Li et al., 2025)) on Hunyuan Video(Kong et al., 2024) and Wan 2.1(Wang et al., 2025). On HunyuanVideo (Figure 1), MOD-DiT achieves a consistent 2.2× speedup with outputs nearly identical to full attention—no visible loss in details, spatiotemporal coherence, or color fidelity. On Wan 2.1 (Figures 16 and 17, processing 81-frame 512×832 videos), it maintains full-attention-level subject consistency and scene realism while delivering a 1.9× speedup. These comparisons confirm MOD-DiT retains full-attention-quality generation with substantial acceleration across different models and inputs, showcasing strong adaptability.

Figure 15. Comparison of the visualization effects of different sparse attention methods on HunyuanVideo(Kong et al., 2024). Our method MOD-DiT consistently achieves 2.2× speedup, and keep almost the same as original videos.

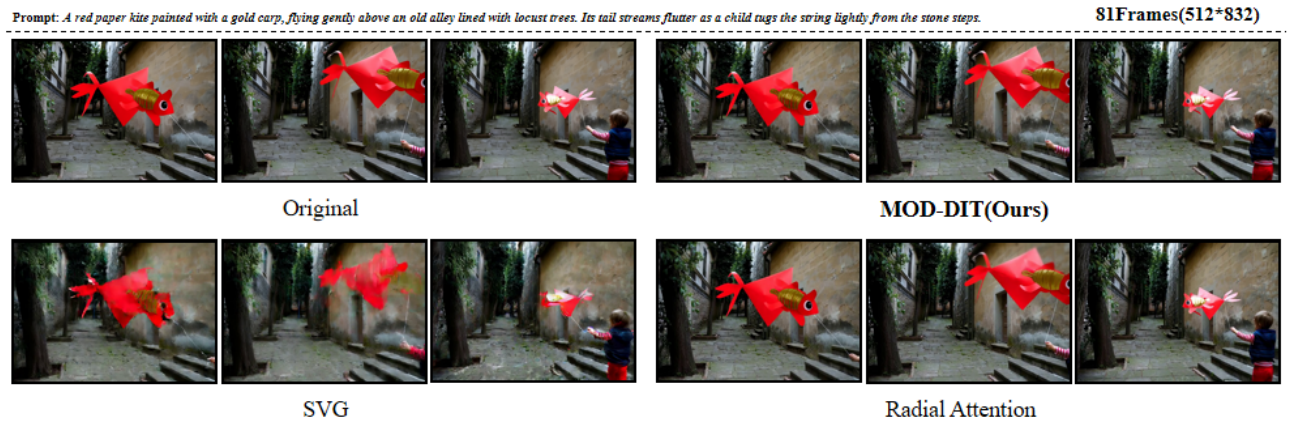


Figure 16. Comparison of the visualization effects of different sparse attention methods on Wan 2.1(Wang et al., 2025). Our method MOD-DiT consistently achieves 1.9× speedup, and keep almost the same as original videos.

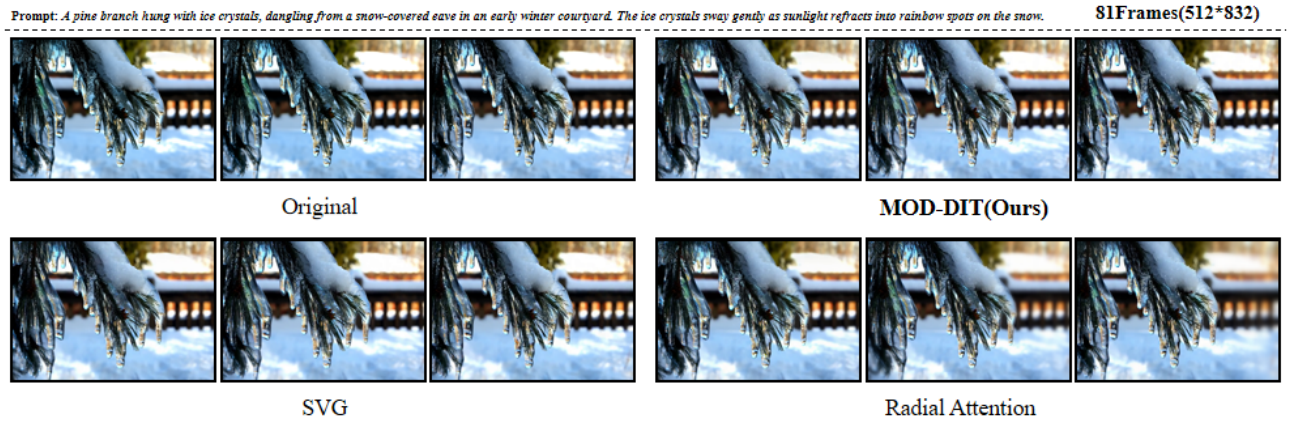


Figure 17. Comparison of the visualization effects of different sparse attention methods on Wan 2.1(Wang et al., 2025). Our method MOD-DiT consistently achieves 1.9× speedup, and keep almost the same as original videos.

C. Algorithm

Algorithm 1 MOD-DiT: Dynamic Sparse Attention for Efficient Video Diffusion

Require: Input features $I \in \mathbb{R}^{B_b \times N \times D}$, Number of heads H, Total denoising steps T, Warm-up steps m, Prediction interval Δt, Block size B, Sparsity threshold η, Block-diagonal threshold τ_e, Top-K value K
Ensure: Generated video features after efficient diffusion inference

Linear projection of input I to get query Q, key K, value V: $Q, K, V \in \mathbb{R}^{B_b \times H \times N \times d}$ (where $d = D/H$)
Warm-up Phase:
for $t = 1$ To m **do**
  Compute full attention map $A^{(t)} = softmax\left(\frac{QK^T}{\sqrt{d}}\right)$ using FlashAttention-2 (Dao, 2023)
end for
$\hat{A}^{(m-1)} = A^{(m-1)}$, $\hat{A}^{(m)} = A^{(m)}$
Convert $\hat{A}^{(m-1)}$ to sparsity map $\hat{S}^{(m-1)} \in \mathbb{R}^{(N/B)\times(N/B)}$ via Eq.(2): partition $\hat{A}^{(m-1)}$ into non-overlapping $B \times B$ blocks, calculate block sparsity with η
Convert $\hat{A}^{(m)}$ to sparsity map $\hat{S}^{(m)} \in \mathbb{R}^{(N/B)\times(N/B)}$ via Eq.(2), calculate block sparsity with η
Compute $X^{(m-1)} = [c_k^{(m-1)}, d_k^{(m-1)}, e_k^{(m-1)}]$ via Eq.(4) and least-squares kernel (Sec.D)
Compute $X^{(m)} = [c_k^{(m)}, d_k^{(m)}, e_k^{(m)}]$ via Eq.(4) and least-squares kernel (Sec.D)
Predict initial intensities $\{\hat{c}_k^{(t')}, \hat{d}_k^{(t')}\}$ for $t' \in (m, m + \Delta t]$ using linear interpolation (Sec.4.2)
Main Denoising Phase:
for $t = m + 1$ To T **do**
  if $t \mod \Delta t == 0$ **then**
    Reconstruct $\hat{A}^{(t)}$ by fusing masked $A^{(t)}_{\text{masked}}$ (from current sparse attention) and historical $\hat{A}^{(t-\Delta t)}$ (Sec.4.1)
    Convert $\hat{A}^{(t)}$ to sparsity map $\hat{S}^{(t)} \in \mathbb{R}^{(N/B)\times(N/B)}$ via Eq.(2)
    Compute intensities $X^{(t)} = [c_k^{(t)}, d_k^{(t)}, e_k^{(t)}]$ via Eq.(4) and least-squares kernel (Sec.D)
    Predict intensities $\{\hat{c}_k^{(t')}, \hat{d}_k^{(t')}\}$ for $t' \in (t, t + \Delta t]$ using linear interpolation (Sec.4.2)
  end if
  For vertical/parallel-diagonal patterns: Select Top-K informative patterns from $\{\hat{c}_k^{(t)}, \hat{d}_k^{(t)}\}$ (Sec.4.3), map to $B \times B$ block positions
  For block-diagonal patterns: Preserve if $min(\hat{e}^{(m-1)}, \hat{e}^{(m)}) > \tau_e$ (Sec.4.3), where block-diagonal regions are defined as $B \times B$ blocks (consistent with block size B)
  Construct block-wise mask $M^{(t)} \in \mathbb{R}^{(N/B)\times(N/B)}$: mark valid/invalid $B \times B$ blocks by combining Top-K patterns and preserved block-diagonal patterns
  Upsample $M^{(t)}$ to token-level mask $\bar{M}^{(t)} \in \mathbb{R}^{N\times N}$: extend each block's mask state to all $B \times B$ tokens in the block
  Concatenate masks across all attention heads: $\bar{M}^{(t)} = cat(\bar{M}_1^{(t)}, \bar{M}_2^{(t)}, ..., \bar{M}_H^{(t)})$
  Compute block-wise sparse attention via Eq.(1) (SageAttention (Zhang et al., 2025b)):
    1. Split Q, K into $B \times B$ token blocks (consistent with block size B)
    2. Apply $\bar{M}^{(t)}$ to filter invalid blocks, skip redundant token interactions
    3. Online softmax on valid blocks: $O = softmax\left(\frac{QK^T}{\sqrt{d}} + \bar{M}^{(t)}\right)V$
  Update features with sparse attention output O
end for
Return Final video features after T denoising steps

D. Comprehensive Design of the Efficient Least Squares Kernel for Dynamic Attention Pattern Approximation

This section provides a detailed exposition of the optimized least squares kernel designed for efficient approximation of dynamic attention patterns in MOD-DiT. The kernel addresses the fundamental computational challenges through sophisticated mathematical formulation, advanced parallel computation strategies, and robust numerical optimization techniques, enabling real-time inference while maintaining high accuracy.

D.1. Key Notations

To ensure clarity, we first supplement the key notations used in this section (consistent with the main text):

- $S^{(t)} \in \mathbb{R}^{n \times n}$: Attention sparsity map at denoising step t, derived from partitioning the full attention map into $B \times B$ non-overlapping blocks.
- n: Number of blocks per dimension in the sparsity map, where $n = N/B$ (N is total number of tokens, B is block size).
- $M \in \mathbb{R}^{n^2 \times (3n-1+|\mathcal{A}|)}$: Design matrix composed of basis matrices for three core attention patterns.
- $X^{(t)} \in \mathbb{R}^{(3n-1+|\mathcal{A}|)\times 1}$: Coefficient vector to be optimized, including intensities of parallel-to-main-diagonal, vertical, and block-diagonal patterns.
- $\{C_k\}_{k=1}^{2n-1}$: Basis matrices for parallel-to-main-diagonal patterns, with $\delta_k = k - (n-1)$ denoting the diagonal offset.
- $\{D_k\}_{k=1}^{n}$: Basis matrices for vertical patterns.
- $\{E_k\}_{k\in\mathcal{A}}$: Basis matrices for block-diagonal patterns, where $\mathcal{A}$ is the set of block-diagonal indices.
- $|\mathcal{A}|$: Number of block-diagonal basis matrices, determined by the block partition strategy.
- H: Number of attention heads in the transformer architecture.
- b: Number of tokens per frame in video generation.
- m: Number of frames, where $m = n/b$.

D.2. Design Insight and Motivation

The core challenge addressed by this kernel is the efficient approximation of dynamic attention patterns in video diffusion transformers. As revealed in the main text, attention sparsity maps $S^{(t)}$ exhibit a dynamic mixture of three structured patterns (parallel-to-main-diagonal, vertical, block-diagonal) that evolve across denoising steps. Directly solving the least squares problem for pattern approximation would incur prohibitive computational costs: - Naive construction of the design matrix M requires $O(n^2 \times (3n - 1 + |\mathcal{A}|))$ storage, which is infeasible when n reaches thousands of tokens. Explicit matrix operations for least squares solution would result in $O(n^5)$ time complexity, making real-time inference impossible.

To address these issues, MOD-DiT's kernel is designed with three core insights.

Structural Sparsity Exploitation. The three core attention patterns have inherent sparse and regular structures, enabling analytical computation of matrix products (e.g., $M^T M$) without explicit matrix construction.

Computational Complexity Reduction. By decomposing high-complexity matrix operations into low-dimensional structured computations, the kernel reduces time complexity from naive $O(Hn^5)$ to $O(Hn^3)$ and storage complexity from $O(n^3)$ to $O(n^2)$.

Hardware-Aware Optimization. Leveraging GPU multi-level parallelism (inter-head, inter-feature, element-wise) optimizes memory access and computation efficiency, ensuring the kernel overhead accounts for only 1-2% of total attention computation time.

This design enables MOD-DiT to achieve 100× speedup in solving the least squares problem compared to native solvers (e.g., PyTorch's `torch.lstsq`), while maintaining high approximation accuracy for dynamic attention patterns—critical for balancing inference efficiency and video generation quality.

D.3. Theoretical Foundation and Problem Formulation

The core mathematical problem involves approximating the attention sparsity map $S^{(t)} \in \mathbb{R}^{n\times n}$ at denoising step t through a linear combination of structured basis matrices. The objective function is formulated as the minimization of the Frobenius norm:

$$\min_{X^{(t)}} \left\| vec(S^{(t)}) - MX^{(t)} \right\|_F^2$$

where $vec(\cdot)$ denotes matrix vectorization, M is the design matrix integrating the three core pattern families, and $X^{(t)}$ is the coefficient vector quantifying the intensity of each pattern at step t. The Frobenius norm ensures optimal fitting in the matrix space while maintaining numerical stability.

The design matrix M incorporates three distinct pattern types that capture the essential characteristics of attention mechanisms in video generation:

- **Parallel-to-main-diagonal patterns** $\{C_k\}_{k=1}^{2n-1}$ model inter-frame spatial correlation, defined as:

$$C_k(i,j) = \begin{cases} 1, & \text{if } j - i = k - (n-1) \\ 0, & \text{otherwise} \end{cases}$$

  where $k \in [1, \ldots, 2n-1]$ covers all possible parallel-to-main diagonals, with $\delta_k = k - (n-1)$ representing the diagonal offset.

- **Vertical patterns** $\{D_k\}_{k=1}^{n}$ capture global cross-token dependencies through column-wise structures:

$$D_k(i,j) = \begin{cases} 1, & \text{if } j = k \\ 0, & \text{otherwise} \end{cases}$$

- **Block-diagonal patterns** $\{E_k\}_{k\in\mathcal{A}}$ maintain intra-frame temporal coherence by grouping tokens within frames. For the k-th block region $B_k = [a_k, b_k] \times [a_k, b_k]$, the basis matrix is defined as:

$$E_k(i,j) = \begin{cases} 1, & \text{if } (i,j) \in B_k \\ 0, & \text{otherwise} \end{cases}$$

  where b denotes the number of tokens per frame, and $m = n/b$ represents the number of frames.

The complete mathematical representation of the design matrix M and coefficient vector $X^{(t)}$ is:

$$M = \left[\{vec(C_k)\}_{k=1}^{2n-1}, \{vec(D_k)\}_{k=1}^{n}, \{vec(E_k)\}_{k\in\mathcal{A}} \right]$$

$$X^{(t)} = \left[c_1^{(t)}, ..., c_{2n-1}^{(t)}, d_1^{(t)}, ..., d_n^{(t)}, e_1^{(t)}, ..., e_{|\mathcal{A}|}^{(t)} \right]^T$$

where $c_k^{(t)}$, $d_k^{(t)}$, and $e_k^{(t)}$ are intensity scalars quantifying the contribution of each parallel-to-main-diagonal, vertical, and block-diagonal pattern at step t, respectively.

The normal equation derived from the optimization problem provides the computational foundation:

$$M^T M X^{(t)} = M^T vec(S^{(t)})$$

This formulation transforms the pattern approximation problem into solving a linear system, where the structured nature of the basis matrices enables significant computational optimizations.

D.4. Computational Complexity Analysis and Optimization Strategies

Direct implementation of the least squares solution would require constructing the design matrix $M \in \mathbb{R}^{n^2 \times (3n-1+|\mathcal{A}|)}$, incurring $O(n^3)$ storage complexity and $O(n^5)$ time complexity for explicit matrix operations. Such computational demands are prohibitive for video generation tasks where n can reach several thousand tokens.

MOD-DiT's optimization strategy leverages the inherent sparsity and structural properties of the basis matrices to achieve $O(n^2)$ storage complexity and $O(n^3)$ time complexity through analytical block decomposition. The $M^T M$ matrix exhibits a block structure that enables efficient computation:

$$M^T M = \begin{bmatrix} C^T C & C^T D & C^T E \\ D^T C & D^T D & D^T E \\ E^T C & E^T D & E^T E \end{bmatrix}$$

where $C = [C_1, \ldots, C_{2n-1}]$, $D = [D_1, \ldots, D_n]$, and $E = \left[E_1, \ldots, E_{|\mathcal{A}|}\right]$. Each block can be computed analytically using specialized algorithms:

For the parallel-to-main-diagonal block $C^T C$, the elements are computed as:

$$\left[C^T C\right]_{ij} = \langle C_i, C_j \rangle_F = \begin{cases} n - |\delta_i|, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}$$

where $\delta_i = i - (n-1)$. This results in a diagonal matrix with elements representing the lengths of corresponding parallel-to-main diagonals.

The vertical line block $D^T D$ simplifies to:

$$\left[D^T D\right]_{ij} = \begin{cases} n, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}$$

The cross-term block $C^T D$ is computed as:

$$\left[C^T D\right]_{ij} = \begin{cases} 1, & \text{if parallel-to-main-diagonal } i \text{ passes through column } j \\ 0, & \text{otherwise} \end{cases}$$

The block-diagonal term $E^T E$ computes as:

$$E^T E = \text{diag}\left(b^2, b^2, ..., b^2\right) \in \mathbb{R}^{|\mathcal{A}| \times |\mathcal{A}|}$$

where each diagonal element equals b^2 (the number of elements in each block-diagonal region).

The right-hand side vector $M^T vec(S^{(t)})$ is computed through partitioned inner products:

$$M^T vec(S^{(t)}) = \begin{bmatrix} C^T vec(S^{(t)}) \\ D^T vec(S^{(t)}) \\ E^T vec(S^{(t)}) \end{bmatrix}$$

where each component is calculated efficiently:

$$\begin{aligned} \left[C^T vec(S^{(t)})\right]_k &= \sum_{(i,j) \in \mathcal{D}_k} S^{(t)}(i,j), \\ \left[D^T vec(S^{(t)})\right]_j &= \sum_{i=0}^{n-1} S^{(t)}(i,j), \\ \left[E^T vec(S^{(t)})\right]_k &= \sum_{(i,j) \in B_k} S^{(t)}(i,j) \end{aligned}$$

with $\mathcal{D}_k$ denoting the set of indices for the k-th parallel-to-main-diagonal pattern.

D.5. Multi-level Parallelization Architecture and Kernel Implementation

The kernel implements a sophisticated three-level parallelization strategy optimized for modern GPU architectures, ensuring maximal hardware utilization and computational efficiency.

First Level: Inter-head Parallelism leverages the independence between attention heads by distributing computations across multiple GPU streaming multiprocessors. Each attention head $h \in [1, H]$ is processed concurrently, with head indices mapped to the y-dimension of the execution grid. This level exploits the natural parallelism in transformer architectures where attention heads operate independently.

Second Level: Inter-feature Parallelism assigns different geometric features (parallel-to-main-diagonal, vertical, block-diagonal) to separate thread blocks for simultaneous execution. This approach maximizes occupancy by utilizing the GPU's ability to manage multiple concurrent thread blocks, with feature assignments optimized to balance computational load.

Third Level: Element-wise Parallelism employs fine-grained parallel reduction models within thread blocks for efficient summation operations. Using a tree-based reduction pattern with 256-thread blocks, this level ensures optimal memory access patterns and computational density.

The kernel design incorporates specialized functions for different computational patterns.

INNER PRODUCT CALCULATION KERNEL

Implements the Frobenius inner product for matrices $A, B \in \mathbb{R}^{n\times n}$:

$$\langle A, B\rangle_F = \sum_{i=1}^{n}\sum_{j=1}^{n} A(i,j)\cdot B(i,j) = \mathrm{tr}(A^T B)$$

The kernel flattens matrices into vectors $v \in \mathbb{R}^{n^2}$ and employs a hierarchical reduction strategy:

$$s_t^{(0)} = \sum_{k=t}^{n^2} v_k^2, \quad k \equiv t \pmod T$$

where $T = 256$ is the thread block size. The reduction follows a logarithmic pattern:

$$s_t^{(\ell+1)} = s_t^{(\ell)} + s_{t+2^\ell}^{(\ell)}, \quad \ell = 0, 1, \ldots, \log_2(T) - 1$$

with results combined atomically across thread blocks.

PARALLEL-TO-MAIN-DIAGONAL SUMMATION KERNEL

Processes parallel-to-main-diagonal patterns with optimal memory access patterns. For pattern k with offset $\delta_k = k-(n-1)$:

$$\mathcal{D}_k = \begin{cases} \{(i, i+\delta_k) : i \in [0, n-1-\delta_k]\}, & \delta_k \geq 0 \\ \{(i-\delta_k, i) : i \in [0, n-1+\delta_k]\}, & \delta_k < 0 \end{cases}$$

The kernel employs efficient index mapping and shared memory reduction to compute:

$$\langle S_h^{(t)}, C_k\rangle = \sum_{(i,j)\in\mathcal{D}_k} S_h^{(t)}(i,j)$$

VERTICAL LINE SUMMATION KERNEL

Computes column sums through parallel accumulation:

$$\langle D_k, S_h^{(t)}\rangle = \sum_{i=0}^{n-1} S_h^{(t)}(i,k) = \mathbf{1}^T S_h^{(t)} e_k$$

where e_k is the standard basis vector.

BLOCK SUMMATION KERNEL

Handles block-diagonal patterns through partitioned computation:

$$\langle E_k, S_h^{(t)} \rangle = \sum_{(i,j) \in B_k} S_h^{(t)}(i,j)$$

with each block processed concurrently using optimized memory access patterns.

D.6. Numerical Stability and Linear System Solution Techniques

To ensure numerical robustness in solving the linear system $M^T M X^{(t)} = M^T vec(S^{(t)})$, MOD-DiT incorporates several advanced techniques:

REGULARIZATION SCHEME

A Tikhonov regularization approach ensures well-posedness:

$$M^T M \leftarrow M^T M + \lambda I$$

where $\lambda = 10^{-8}$ provides stability without significantly affecting solution quality. The regularization parameter is chosen through empirical analysis to balance numerical stability and approximation accuracy.

ADAPTIVE SOLUTION STRATEGIES

The system employs multiple solution methods adaptively based on matrix properties:

Cholesky Decomposition. Primary method for positive definite systems:

$$M^T M = LL^T, \quad X^{(t)} = L^{-T} \left(L^{-1} (M^T vec(S^{(t)})) \right)$$

This approach offers optimal numerical stability with $O(n^3)$ complexity for the factorization.

LU Decomposition with Partial Pivoting. Fallback method for numerically challenging systems:

$$P(M^T M) = LU, \quad X^{(t)} = U^{-1} \left(L^{-1} P (M^T vec(S^{(t)})) \right)$$

where P is the permutation matrix for numerical stability.

Moore-Penrose Pseudoinverse. Handles rank-deficient cases:

$$X^{(t)} = (M^T M)^{\dagger} M^T vec(S^{(t)})$$

computed via singular value decomposition with threshold-based singular value retention.

D.7. Complexity Analysis and Performance Characterization

The overall computational complexity of MOD-DiT's kernel is analyzed through meticulous examination of each component:

- **Matrix Product Computation**: $M^T M$ calculation requires $O(Hn^2)$ operations leveraging structural sparsity.
- **Right-hand Side Computation**: $M^T vec(S^{(t)})$ evaluation consumes $O(Hn^2)$ operations through optimized summations.
- **Linear System Solution**: The factorization and solve steps require $O(Hn^3)$ operations.

The total complexity of $O(Hn^3)$ represents a significant improvement over the naive $O(Hn^5)$ approach. MOD-DiT's optimized implementation achieves demonstrated speedups of 100× compared to standard solvers like `torch.lstsq`.

E. Computation Analysis

E.1. Additional computation experiments results

We conducted a comparative experiment to evaluate the computational costs of various MOD-DiT operations against a single Full Attention operation. As shown in table3, the additional computational overhead introduced by MOD-DiT is negligible compared to Full Attention. Moreover, when masking 50% of the blocks, MOD-DiT's computational cost is only 74% of that of Full Attention.

Table 3. Comparison of computational costs between MOD-DiT additional operations and one Full Attention operation using the hunyuan model with size $512 \times 832 \times 81$.

Operation Description	FLOPS (TFLOPS)
Full Attention	7.6127
Convert to Sparsity	0.0297
Solve kernel	0.0002
MOD-DiT: topk200 (50% mask)	5.7096

E.2. Theoretical Analysis

We conduct a detailed theoretical analysis of the additional computational overhead introduced by MOD-DiT's core components, focusing on a single attention head to clarify complexity contributions. All overheads are compared against the computational cost of full attention (denoted as $O(N^2D)$, where N is the total number of tokens and D is the feature dimension per token) to verify their negligibility.

E.2.1. Overhead of Attention-to-Sparsity Map Transformation

The transformation from the attention map A to the attention sparsity map S (Eq. (2) in the main text) involves partitioning A into non-overlapping $B \times B$ blocks and calculating the sparsity scalar for each block. For a single attention head, this operation only requires a single traversal of all elements in the attention map. Each element is checked against the sparsity threshold η once, and the sum of valid elements per block is computed. The total computational complexity of this step is $O(N^2)$, as it scales linearly with the number of elements in the attention map.

E.2.2. Overhead of Sparsity Map Reconstruction

Sparsity map reconstruction (Sec.4.1) fuses the masked attention map at the current step with the reconstructed complete attention map from the previous step. For each token pair (p, q), the reconstruction operation simply selects the valid attention value based on the mask state (either retaining the current masked value or reusing the historical reconstructed value). This process requires traversing all elements of the attention map exactly once, resulting in a computational complexity of $O(N^2)$ for a single attention head.

E.2.3. Overhead of Linear Prediction and Top-K Router

Linear Prediction. The linear prediction module (Sec. 4.1) models the intensity scalars of parallel-diagonal and vertical patterns as linear functions of denoising steps. The total number of prediction steps is $\frac{T}{\Delta t}$, where T is the total number of denoising steps and Δt is the interval between consecutive sparsity map updates. Each prediction step leverages precomputed intensity scalars from historical steps and performs a constant-time linear interpolation, leading to an overall complexity of $O\left(\frac{T}{\Delta t}\right)$.

Top-K Router. The Top-K selection (Sec. 4.3) sorts the merged intensity scalars of parallel-diagonal and vertical patterns (totaling $O(n)$ elements, where $n = N/B$ is the number of blocks per dimension) and retains the top K informative patterns. The time complexity of sorting $O(n)$ elements for Top-K selection is $O(n \log K)$, which is efficient given the typical range of K (much smaller than n).

E.3. Aggregate Overhead and Negligibility

Summing the additional computational overheads for a single attention head, the dominant term is $O(N^2)$. The remaining terms $O\left(\frac{T}{\Delta t}\right)$ (linear prediction) and $O(n \log K)$ (Top-K router) are sub-quadratic in N (e.g., $n = N/B$ reduces scale, $\frac{T}{\Delta t}$ is a fixed constant for inference) and thus negligible compared to $O(N^2)$.

Compared to the computational cost of full attention ($O(N^2 D)$), the additional overhead is theoretically negligible. In practical diffusion transformer (DiT) architectures (e.g., DiT-Base/ Large, where $D = 768/1024$; video-specific variants like CogVideoX-v1.5 often use $D \geq 768$), D is typically a large value (hundreds to thousands). This makes $N^2 D$ vastly larger than N^2, leading the additional overhead to account for only 1–2$\%$ of full attention's computation cost—consistent with experimental measurements.