Sparse-vDiT: Unleashing the Power of Sparse Attention to Accelerate Video Diffusion Transformers

Pengtao Chen[1,2], **Xianfang Zeng**[2*], **Maosen Zhao**[1], **Mingzhu Shen**[3], **Wei Cheng**[2], **Gang Yu**[2], **Tao Chen**[1,4†]

[1]College of Future Information Technology, Fudan University, Shanghai, China
[2]StepFun, Shanghai, China
[3]Department of Electrical and Electronic Engineering, Imperial College London, London, U.K.
[4]Shanghai Innovation Institute, Shanghai, China

Abstract

While Diffusion Transformers (DiTs) have achieved breakthroughs in video generation, this long sequence generation task remains constrained by the quadratic complexity of attention mechanisms, resulting in significant inference latency. Through detailed analysis of attention maps in Video Diffusion Transformer (vDiT), we identify three recurring sparsity patterns: diagonal, multi-diagonal, and vertical-stripe structures. And even 3-6% attention heads can be skipped. Crucially, these patterns exhibit strong layer-depth and head-position correlations but show limited dependence on the input content. Leveraging these findings, we propose Sparse-vDiT, a sparsity acceleration framework for vDiT comprising: 1) Pattern-optimized sparse kernels that replace dense attention with computationally efficient implementations for each identified sparsity pattern. 2) An offline sparse diffusion search algorithm that selects the optimal sparse computation strategy per layer and head via hardware-aware cost modeling. After determining the optimal configuration, we fuse heads within the same layer that share the same attention strategy, enhancing inference efficiency. Integrated into state-of-the-art vDiT models (CogVideoX1.5, HunyuanVideo, and Wan2.1), Sparse-vDiT achieves 2.09×, 2.38×, and 1.67× theoretical FLOP reduction, and actual inference speedups of 1.76×, 1.85×, and 1.58×, respectively, while maintaining high visual fidelity, with PSNR values reaching 24.13, 27.09, and 22.59. Our work demonstrates that latent structural sparsity in vDiTs can be systematically exploited for long video synthesis.

1 Introduction

In recent years, diffusion models have achieved significant advances in image generation (Rombach et al. 2022), prompting growing interest in extending them to video synthesis. Early approaches, such as SVD (Blattmann et al. 2023) and Dynamicrafter (Xing et al. 2024), employed a 2D+1D framework that provided computational efficiency but lacked real-time interaction between spatial and temporal features, resulting in limited spatiotemporal consistency. Recent progress in 3D full-attention Video Diffusion Transformers (vDiT) (Peebles and Xie 2023) has effectively addressed these limitations.

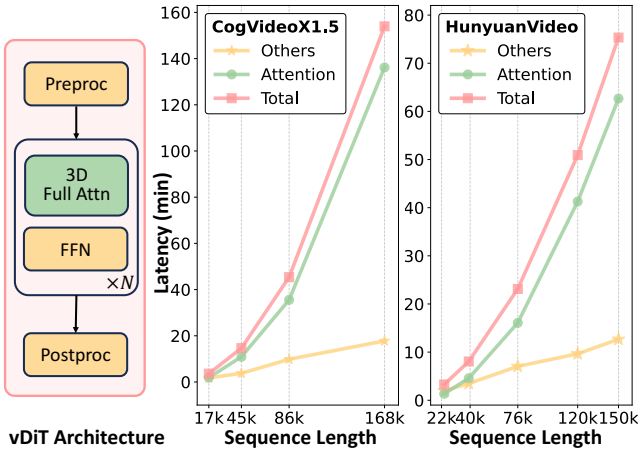

Figure 1: The architecture of vDiT and inference latency analysis of its two variants, CogVideoX1.5 and HunyuanVideo, across different components. The latency of the attention module dominates under long sequence settings.

Built on this foundation, models such as OpenSora (Lin et al. 2024), CogVideoX (Yang et al. 2024), HunyuanVideo (Kong et al. 2024), and Wan2.1 (Wang et al. 2025) demonstrate strong spatiotemporal coherence and high video quality. These methods have been widely applied in fields including animation generation (He et al. 2023; Hu 2024), video editing (Zhang et al. 2025a; Wang et al. 2024), and world modeling (Meng et al. 2024; He et al. 2025).

Although 3D full-attention vDiT models demonstrate strong video generation performance and are widely adopted, they suffer from high computational costs and large inference latency. For instance, generating a 5-second 720p video at 24 fps using the HunyuanVideo model on an NVIDIA A800 GPU takes approximately 50 minutes. This inefficiency stems from joint spatiotemporal tokenization yielding up to 120k tokens, which imposes high costs due to attention's quadratic complexity (Vaswani et al. 2017). Figure 1 reveals attention accounts for 77% of latency in CogVideoX1.5 (86k tokens) and 81% in HunyuanVideo (120k tokens), a proportion that rises with sequence length. Consequently, 3D full attention is the main bottleneck for vDiT-based video generation.

*Project leader. Work was done when interned at StepFun.
†Corresponding author.

Fortunately, the 3D full attention mechanism exhibits significant redundancy despite its considerable computational cost. First, we observe that some attention heads in vDiT are redundant, as skipping their computations has minimal effect on the final output. Second, redundancy is also present in the computation of the attention map, namely in the $\boldsymbol{QK}^T$ product. We find that vDiT attention maps commonly follow four distinct patterns: full attention, diagonal sparsity, multi-diagonal sparsity, and vertical-strip sparsity. The latter three patterns suggest that computing the full attention map is often unnecessary. Further experiments reveal that these sparse patterns remain stable across different inputs and are primarily determined by the position of attention within the vDiT. This fixed redundancy provides a strong basis for optimization.

Building on these findings, we propose Sparse-vDiT, a sparse method that accelerates vDiT for video generation. To reduce redundancy among attention heads, we introduce a head skipping strategy. We observe that vDiT's attention maps commonly follow three sparse patterns: diagonal, multi-diagonal, and vertical-stripes. To enable actual speedup, we design predefined kernels tailored to each pattern. Since these sparsity patterns are relatively fixed and input-invariant, we develop an offline sparse diffusion search algorithm that identifies the optimal attention pattern for each head using only a small number of samples. After the search, the computation pattern of each head is fixed. We then group heads with the same sparsity pattern within each layer and fuse them to further accelerate inference by leveraging their fixed structure. We conducted experiments on three widely used vDiT-based models: CogVideoX1.5, HunyuanVideo, and Wan2.1. On CogVideoX1.5, it reduced FLOPs by 2.09× and accelerated inference by 1.76× (0.14 LPIPS, even improved ImageQual). For HunyuanVideo and Wan2.1, it achieved 2.38×/1.67× FLOPs reduction and 1.85×/1.58× speedup, maintaining high quality (SSIM 0.87/0.80, PSNR 27.03/22.59). These results indicate that Sparse-vDiT effectively balances computational efficiency and generation quality.

The contributions of our paper are as follows:

- We find that attention heads in vDiT are partly redundant. Meanwhile, many heads often exhibit recurring sparse attention patterns, including diagonal sparsity, multi-diagonal sparsity, and vertical-stripe sparsity. These redundant patterns are closely associated with the depth and position of attention heads within the vDiT architecture.
- Building on these insights, we propose Sparse-vDiT, which accelerates vDiT by skipping redundant heads and applying pattern-aligned sparse attention kernels. It introduces an offline sparse diffusion search that selects the optimal sparse mode for each head using a small number of samples, followed by intra-layer fusion of heads with identical attention patterns to enhance inference efficiency.
- Sparse-vDiT achieves 2.09×, 2.38× and 1.67× theoretical FLOPs reduction on CogVideoX1.5, HunyuanVideo and Wan2.1, respectively. It also delivers 1.76×, 1.85×, and 1.58× end-to-end video generation speedups while maintaining comparable generation quality, with PSNR scores of 24.13, 27.09, and 22.59. Sparse-vDiT consistently outperforms existing state-of-the-art (SOTA) methods.

2 Related Work

Efficient Diffusion Model. Diffusion models are inherently slow because of their iterative denoising process, leading to growing interest in accelerating inference. Existing approaches include pruning methods (Fang, Ma, and Wang 2023; Castells et al. 2024) that reduce model parameters, quantization techniques (Wu et al. 2024; Zhao et al. 2025) that decrease parameter bit-width and computational overhead, and caching strategies (Ma, Fang, and Wang 2024; Chen et al. 2024; Shen et al. 2024) that trade memory for computation speed. However, most of these methods are primarily designed for image generation, with relatively few acceleration methods specifically tailored for video diffusion models. For video diffusion, techniques like PAB (Zhao et al. 2024), TeaCache (Liu et al. 2024), and FasterCache (Lv et al. 2024) reuse features by exploiting the similarity between adjacent denoising steps. Other methods reduce the number of timesteps using distillation (Lin et al. 2025) or compress latent spaces using high-ratio VAEs (Tian et al. 2024). In contrast, our approach accelerates inference by exploiting the sparsity in vDiT's attention.

Efficient Attention Mechanism. Attention mechanisms (Vaswani et al. 2017) are central to transformers but suffer from quadratic complexity, limiting long-sequence efficiency. To address this, vision models like Swin Transformer (Liu et al. 2021), NAT (Hassani et al. 2023), and Sparse Transformers (Child et al. 2019), as well as Longformer (Beltagy et al. 2020) in NLP, employ local window attention. Large language models (Touvron et al. 2023) have identified attention sink phenomena (Xiao et al. 2023, 2024), introducing streaming attention that combines sink masking with windowing. Later works, such as MInference (Jiang et al. 2024) and FlexPrefill (Lai et al. 2025), explore diverse static and dynamic sparse patterns. In diffusion models, DiT-FastAttn (Yuan et al. 2024; Zhang et al. 2025b) noted strong local neighbor attention in DiTs, enabling acceleration via windowed attention and cached contexts. CLEAR (Liu, Tan, and Wang 2024), DiG (Zhu et al. 2024), and SANA (Xie et al. 2024) further exploit the sparsity of the attention mechanism to achieve linearized computation. For video diffusion, Efficient-vDiT (Ding et al. 2025) observed that each frame in the attention primarily attends to a fixed set of other frames. This observation introduces tile-based attention to linearized computation. SVG (Xi et al. 2025) identified spatiotemporal sparsity in video attention and optimized attention computation through data reordering and an online scheme. There also emerged other sparse methods (Zhang et al. 2025d,c; Wu et al. 2025; Yang et al. 2025; Sun et al. 2025). In contrast to prior work, we analyze vDiT attention heads and reveal that many focus on a few salient tokens or contribute minimally to noise estimation. Moreover, these redundant patterns are closely associated with the depth and position of attention heads within the network. Based on these findings, we propose an offline sparse acceleration framework that integrates head skipping with three attention sparsity patterns. Considering the fixed nature of offline optimization, fusion optimization is performed on a fixed attention pattern at each attention layer. This is potentially orthogonal to current timestep-distillation methods (Lv et al. 2025).

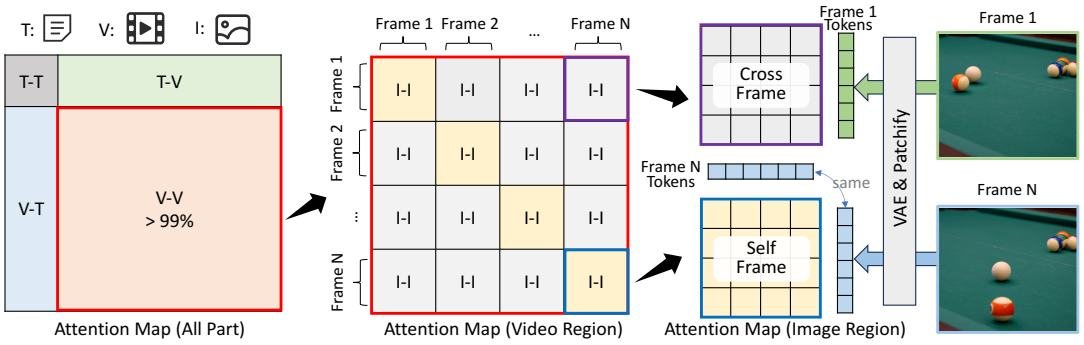

Figure 2: Visualization of the vDiT attention map showing four interaction regions. The dominant V-V region has diagonal blocks for self-frame and off-diagonal blocks for cross-frame interactions.

3 Preliminary

Full Attention. The multi-head attention mechanism (Vaswani et al. 2017) constitutes a fundamental building block in vDiT. Let the input hidden features be denoted as $\boldsymbol{I} \in \mathbb{R}^{B\times N\times D}$, where B is the batch size, N the number of tokens, and D the original feature dimension. Through learnable linear projections, $\boldsymbol{I}$ is transformed into three tensors: query ($\boldsymbol{Q}$), key ($\boldsymbol{K}$) and value ($\boldsymbol{V}$). Each of these tensors has dimensions $\mathbb{R}^{B\times H\times N\times d}$, where H denotes the number of attention heads, and $d = D/H$ represents the reduced feature dimension per head. The attention outputs refined features $\boldsymbol{O} \in \mathbb{R}^{B\times N\times D}$ preserving the original dimension of $\boldsymbol{I}$. The attention transformation process is defined as follows: for each head $h \in \{1, ..., H\}$,

$$Attn(\boldsymbol{Q}_h, \boldsymbol{K}_h, \boldsymbol{V}_h) = softmax(\frac{\boldsymbol{Q}_h \boldsymbol{K}_h^T}{\sqrt{d}})\boldsymbol{V}_h \in \mathbb{R}^{B\times N\times d}, \tag{1}$$

where $\boldsymbol{Q}_h, \boldsymbol{K}_h, \boldsymbol{V}_h$ are slice operations on the head dimension. Then, merging along the head dimension yields the final output $\boldsymbol{O}$ of the attention. For the full attention mechanism, the entire process described above is executed.

Sparse Attention. In Eq 1, $softmax(\boldsymbol{Q}_h \boldsymbol{K}_h^T/\sqrt{d})$ is known as the attention map, where each value represents how much one token attends to another at the corresponding position. Since its computational complexity is $\mathcal{O}(N^2)$, generating the attention map takes up most of the computation in the attention mechanism. However, in practice, a token usually attends to only a small number of other tokens, rather than maintaining global attention. This results in most values in the attention map being close to zero, showing strong sparsity. In most cases, it is sufficient to compute only the dense regions of the attention map to obtain a sufficiently accurate result. If the sparsity pattern of the attention map is structured, computations involving sparse regions can be omitted at the hardware level using Triton (Tillet, Kung, and Cox 2019) or CUDA, enabling practical acceleration.

4 Methodology

4.1 Attention Mechanism in vDiT

We present the vDiT attention mechanism, beginning with the attention map layout tailored for video generation. We then demonstrate that the mechanism exhibits substantial redundancy. Finally, we show that this redundancy is intrinsic to the architecture and insensitive to input variations.

CogVideoX1.5	PSNR ↑	SSIM ↑	LPIPS ↓
skipping 1%	36.62	0.96	0.01
skipping 3%	33.31	0.95	0.02
skipping 6%	30.02	0.92	0.04
skipping 10%	26.87	0.85	0.09
HunyuanVideo	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
skipping 1%	31.84	0.95	0.02
skipping 3%	28.94	0.91	0.06
skipping 6%	24.21	0.81	0.12
skipping 10%	17.98	0.72	0.22

Table 1: Quantitative impact of skipping different ratios of attention heads on the final generation.

Attention Map in vDiT. Current mainstream vDiT models, such as CogVideoX and HunyuanVideo, mainly adopt the MM-DiT paradigm (Esser et al. 2024). In this design, the token sequence is formed by concatenating text tokens and video tokens, and the corresponding attention map is shown on the left side of Figure 2. The attention map is divided into four parts based on token type and position: T–T, T–V, V–T, and V–V, where T denotes text tokens and V denotes video tokens. Text tokens make up only a small portion of the sequence, while video tokens account for over 99%. In the V–V region (the middle part of Figure 2), video tokens are arranged in the temporal order of frames. As a result, the diagonal blocks represent self-frame interactions, while the off-diagonal ones capture cross-frame interactions.

Analyzing Attention Redundancy in vDiT. We find that attention in vDiT contains considerable redundancy. Some attention heads are non-essential, and skipping them results in minimal performance loss. Moreover, the attention maps exhibit patterns of structured sparsity, which can be exploited to enable efficient sparse computation.

Head Skipping. Not all attention heads in vDiT contribute equally to performance. Evaluating head skipping via minimum MSE, we find that skipping 6% of heads in

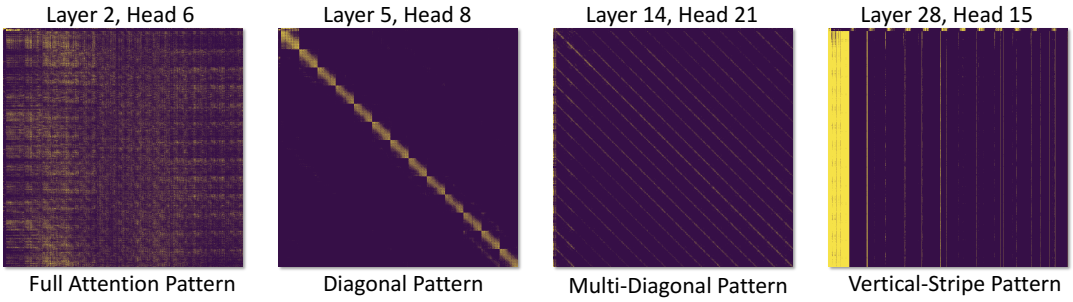

Figure 3: Visualization of the four recurring attention patterns in vDiT.

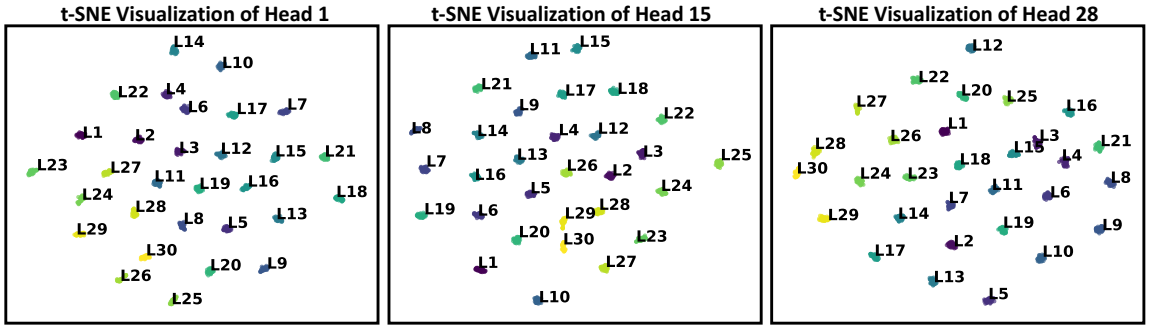

Figure 4: t-SNE visualization of attention patterns along the head dimension on a VBench subset, with different layers indicated by distinct colors. Patterns from different prompts exhibit clustering.

CogVideoX1.5 and 3% in HunyuanVideo preserves quality (Table 1). These results indicate that certain attention heads in vDiT are redundant, suggesting that head skipping may be a practical means to improve efficiency. However, relying solely on skipping is insufficient to achieve high efficiency. As shown in Table 1, both models exhibit noticeable degradation at a 10% skip ratio, indicating that a fine-grained strategy is required for greater speedup.

Given that the sparsity of attention maps can improve the efficiency of transformer models, we conduct an in-depth analysis of the attention map in vDiT. Taking CogVideoX as an example, we visualize its attention maps in Figure 3 and identify four recurring patterns:

Full Attention Pattern. The attention values are evenly distributed, indicating global interactions among tokens. Applying sparse computation to such dense patterns often degrades performance, making efficiency optimization difficult.

Diagonal Pattern. Large values appear along the main diagonal, representing interactions among neighboring tokens within the same frame (as shown in Figure 2). This pattern reflects the model's ability to capture self-frame structure. Since most off-diagonal values are close to zero, the full attention can be well approximated by computing only the diagonal elements of the attention map. This structured sparsity allows for efficient acceleration using window attention.

Multi-Diagonal Pattern. Large values are distributed along multiple evenly spaced diagonals. These diagonals align with the diagonal blocks in the I-I region of Figure 2, indicating strong attention between tokens at nearby spatial positions across different frames. Therefore, this pattern is associated with vDiT's ability to model cross-frame consistency. By rearranging tokens (Xi et al. 2025), this pattern can be transformed into a diagonal structure suitable for optimization with window attention.

Vertical-Stripe Pattern. The attention maps exhibit vertical stripe patterns in high-value regions, indicating that specific heads focus on key video areas. This supports restricting global attention to these regions, with structured sparsity enabling efficient computation via sparse kernels.

Invariant Property of Attention Patterns. We revealed the presence of diverse attention patterns in vDiT above. We further observe that these patterns are strongly correlated with the depth of the attention layers, while being largely independent of the input text. To verify this, we randomly sampled 50 diverse prompts from VBench as a subset and used them to generate videos. For each layer and each attention head in vDiT, we saved the corresponding attention maps. Since we only needed to determine the pattern types, we stored the maps as memory-efficient image files. We then used a ResNet50 to extract high-dimensional features from the images and applied t-SNE to project them into a 2D space along the head dimension. The results are shown in Figure 4, where different colors represent different layers. We observed that, regardless of the head, the attention patterns from different layers form distinct clusters, while those from different prompts tend to cluster together. This confirms that attention patterns in vDiT are highly correlated with attention position and minimally influenced by the input context.

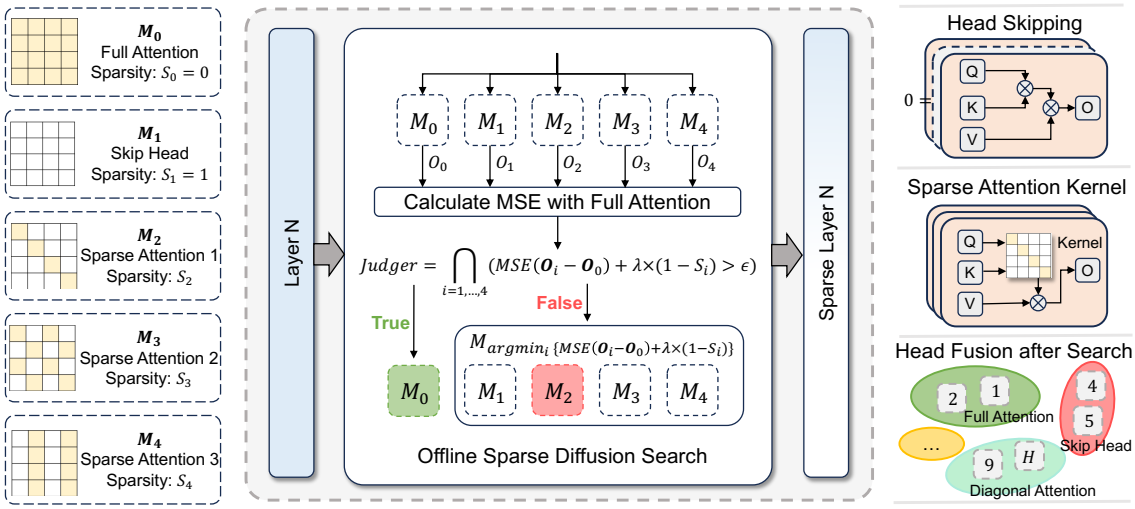

Figure 5: **Overview of the Sparse-vDiT.** We first predefine five types of attention mode $M_{0:4}$. Then, using an offline sparse diffusion search algorithm, we select the best attention mode for each layer and head in vDiT. After the search, for heads set to skip attention, we set their outputs to zero. For the three sparse attention patterns, we create specialized sparse attention kernels to speed up computation. Finally, heads within the same layer that use the same attention mode are fused to improve efficiency.

4.2 Sparse-vDiT: Sparse Acceleration for vDiT

In the previous part, we identified two types of redundancy in the attention mechanism of vDiT: redundancy within the attention heads and redundancy in the attention map computation. We also found that this redundancy is intrinsic to vDiT and only weakly dependent on the input text. Based on these findings, we introduce Sparse-vDiT, a sparse acceleration method designed for vDiT. This method determines the most effective sparse strategy for each head in each layer through offline search, resulting in acceleration. The overall structure of Sparse-vDiT is illustrated in Figure 5.

Sparse Computation Pre-definition. To reduce redundancy in the attention head, we apply the skip strategy M_1, which bypasses the entire process in Eq 1. To maintain consistent output dimensions, the attention output is set to zero. The sparsity of the skip strategy is defined as $S_1 = 1$, while the sparsity of full attention M_0 is $S_0 = 0$. Regarding the three sparse forms $M_i (i = 2, 3, 4)$ shown in Figure 3, we have designed specific sparse kernels to reduce the computation of $softmax(\boldsymbol{Q}\boldsymbol{K}^T/\sqrt{d})$. The sparsity of these kernels is determined by the ratio of actual computation blocks to the total number of blocks, denoted as $S_i (i = 2, 3, 4)$, as shown in Figure 5. In the Sparse-vDiT framework, the sparsity of these kernels is predefined and treated as a fixed constant.

Offline Sparse Diffusion Search. In vDiT, different heads at various layers exhibit distinct attention patterns. Given the set $\boldsymbol{M} = \{M_i (i = 0, ..., 4)\}$ of attention computation modes, the challenge lies in selecting the most appropriate mode for each head. In Sparse-vDiT, we propose an offline sparse diffusion search method to address this. As shown in Figure 5, for each layer in every step of vDiT, we pass the inputs through M_0 to M_4, obtaining the corresponding hidden state results $\boldsymbol{O}_0$ to $\boldsymbol{O}_4$. We then compute the MSE distances between $\boldsymbol{O}_1$ to $\boldsymbol{O}_4$ and $\boldsymbol{O}_0$, that is, $MSE(\boldsymbol{O}_i - \boldsymbol{O}_0), i = 1, ..., 4$, which represent the loss introduced by the sparse attention computation. Our final loss is

$$L_i = MSE(\boldsymbol{O}_i - \boldsymbol{O}_0) + \lambda \times (1 - S_i), \tag{2}$$

where the sparsity penalty is added and λ balances quality and computational cost. If all losses in $\boldsymbol{L} = \{L_i (i = 1, ..., 4)\}$ exceed the desired threshold ϵ, the head retains full attention. Otherwise, the sparse mode with the smallest loss replaces full attention. The specific formulation is as follows:

$$Attn(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}, \boldsymbol{M}) = \begin{cases} M_0(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) & \text{, if } \bigcap_{i=1,...,4} (L_i > \epsilon) \\ M_{\text{argmin}_i \{L_i\}}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) & \text{, otherwise} \end{cases} \tag{3}$$

where ϵ controls the overall sparsity ratio during inference, as discussed in the previous part, vDiT's sparse attention pattern is inherent after pretraining and largely independent of input types. Thus, the search in Sparse-vDiT is offline and requires only a small number of input samples. Once the search is completed, the sparse modes for the entire inference process are fixed. This fixity allows heads with the same sparse pattern within a layer to be fused to fast inference.

5 Experiment

5.1 Experimental Settings

Pretrained Model. To evaluate Sparse-vDiT, we conducted text-to-video generation experiments using three leading pretrained vDiT models: CogVideoX1.5 (Yang et al. 2024), HunyuanVideo (Kong et al. 2024), and Wan2.1 (Wang et al. 2025), which generate 81, 129, and 81 frames at 1360×768, 1280×720, and 1280×720, respectively.

Dataset & Evaluation Metrics. We implemented a comprehensive evaluation framework for both video generation quality and efficiency. Quality was assessed using three types

Method	Against Original			XBench Score		PFLOPS (↓)	Latency (↓)	Speedup (↑)
	SSIM (↑)	PSNR (↑)	LPIPS (↓)	ImageQual (↑)	SubConsist (↑)			
CogVideoX1.5 (Yang et al. 2024)	-	-	-	63.28%	92.96%	147.87	901s	1.00×
+ MInference (Jiang et al. 2024)	0.61	14.63	0.37	56.04%	87.12%	84.89	634s	1.42×
+ WinAttn (Spatial)	0.64	19.07	0.32	**64.84%**	90.92%	72.34	531s	1.69×
+ WinAttn (Temporal)	0.69	19.64	0.28	63.69%	92.66%	72.34	537s	1.67×
+ PAB (Zhao et al. 2024)	0.72	20.93	0.23	59.03%	92.38%	105.88	630s	1.43×
+ SVG (Xi et al. 2025)	0.75	21.92	0.22	63.11%	92.49%	74.57	550s	1.64×
+ Sparse-vDiT (Ours)	**0.82**	**24.13**	**0.14**	63.45%	**92.66%**	**70.69**	**511s**	**1.76**×
HunyuanVideo (Kong et al. 2024)	-	-	-	67.28%	96.79%	612.37	3166s	1.00×
+ MInference (Jiang et al. 2024)	0.64	19.23	0.43	60.53%	88.96%	293.87	2042s	1.55×
+ WinAttn (Spatial)	0.56	17.81	0.56	63.55%	90.26%	258.84	1755s	1.80×
+ WinAttn (Temporal)	0.80	23.76	0.22	**67.32%**	96.38%	258.84	1764s	1.79×
+ SVG (Xi et al. 2025)	0.86	26.83	0.14	67.06%	96.54%	259.79	1802s	1.75×
+ Sparse-vDiT (Ours)	**0.87**	**27.09**	**0.12**	67.13%	**96.69%**	**257.09**	**1715s**	**1.85**×
Wan2.1 (Wang et al. 2025)	-	-	-	67.61%	91.95%	660.49	1935s	1.00×
+ MInference (Jiang et al. 2024)	0.62	15.49	0.36	63.29%	89.32%	469.79	1453s	1.33×
+ WinAttn (Spatial)	0.68	19.14	0.25	67.27%	91.34%	401.21	1265s	1.53×
+ WinAttn (Temporal)	0.73	20.29	0.21	**67.40%**	**91.47%**	401.21	1280s	1.51×
+ SVG (Xi et al. 2025)	0.78	21.96	0.18	67.18%	91.27%	403.50	1298s	1.49×
+ Sparse-vDiT (Ours)	**0.80**	**22.59**	**0.16**	67.35%	91.39%	**397.39**	**1228s**	**1.58**×

Table 2: Comparison of video generation quality and efficiency between Sparse-vDiT and the baseline. XBench refers to VBench for CogVideoX1.5 and Wan2.1 evaluation and Penguin Video Bench for HunyuanVideo.

of metrics: reconstruction fidelity (PSNR (Zhao et al. 2024), SSIM (Wang and Bovik 2002), LPIPS (Zhang et al. 2018)), frame-level visual quality (ImageQual from VBench (Huang et al. 2024)), and temporal consistency (SubConsist from VBench (Huang et al. 2024)). Efficiency was evaluated through theoretical FLOPS, actual latency, and speedup relative to the pretrained model. For datasets, we followed the original CogVideoX (Yang et al. 2024) protocol with GPT-enhanced VBench prompts and used Penguin Video Benchmark (Kong et al. 2024) prompts for HunyuanVideo.

Baseline. We compared several existing acceleration methods for vDiT. These methods include MInference (Jiang et al. 2024), a classical sparse technique migrated from large language models. WinAttn (Beltagy et al. 2020), which applies window sparsity along both temporal and spatial dimensions of video. SVG (Xi et al. 2025), the SOTA method for sparse accelerating vDiTs, and PAB (Zhao et al. 2024), a caching method designed specifically for video diffusion models.

Implementation Details. The baselines MInference, PAB, and SVG are implemented with their official code and settings. As PAB only supports CogVideo, it is excluded from the evaluation of HunyuanVideo and Wan2.1. WinAttn-Spatial and WinAttn-Temporal use SVG's window sizes. Inference for both CogVideoX1.5 and HunyuanVideo was conducted on a single A800 GPU, while Wan2.1 was tested on a single H800 with a batch size of 1.

5.2 Experimental Results Analysis

The qualitative and quantitative results are shown in Figure 6 and Table 2, respectively. Both consistently demonstrate that Sparse-vDiT effectively accelerates video diffusion models.

Reconstruction Fidelity. Sparse-vDiT outperforms all baselines across fidelity metrics on CogVideoX1.5, HunyuanVideo, and Wan2.1. On CogVideoX1.5, Sparse-vDiT achieves an SSIM of 0.82, higher than SVG (0.75) and previous sparse methods (e.g., MInference 0.61, PAB 0.72). Its PSNR of 24.13 surpasses all baselines, with SVG at 21.92, and Sparse-vDiT's LPIPS score (0.14) indicates better perceptual similarity. On HunyuanVideo, Sparse-vDiT records SSIM 0.87 and LPIPS 0.12, outperforming earlier methods like WinAttn (Temporal) (SSIM: 0.76, LPIPS: 0.22). These show strong preservation of spatial and perceptual detail.

Visual Quality. The VBench ImageQual score measures frame-level visual quality. Sparse-vDiT performs on par with or better than most baselines, achieving 63.45% on CogVideoX1.5 and 67.13% on HunyuanVideo. Although WinAttn (Spatial) slightly surpasses Sparse-vDiT in ImageQual on CogVideoX1.5 (64.84%), it comes with lower fidelity scores and higher LPIPS, suggesting a potential overfitting to local texture patterns at the cost of content preservation. On HunyuanVideo, Sparse-vDiT delivers ImageQual scores highly comparable to the best-performing methods, including SVG (67.06%) and WinAttn (Temporal) (67.32%). Overall, Sparse-vDiT maintains competitive frame-level realism, demonstrating balanced and robust performance.

Temporal Consistency. Temporal coherence is crucial in video generation, and Sparse-vDiT excels in the SubConsist metric, assessing subject and motion consistency across frames. On CogVideoX1.5, it achieves 92.66%, matching top methods like WinAttn (Temporal) and PAB. On HunyuanVideo, Sparse-vDiT reaches 96.69%, closely following the original model's 96.79%. Notably, Sparse-vDiT maintains high temporal stability while delivering top-tier fidelity, unlike other methods that prioritize spatial quality at the expense of temporal consistency. Its sparse acceleration strategy that effectively minimizes temporal artifacts by preserving

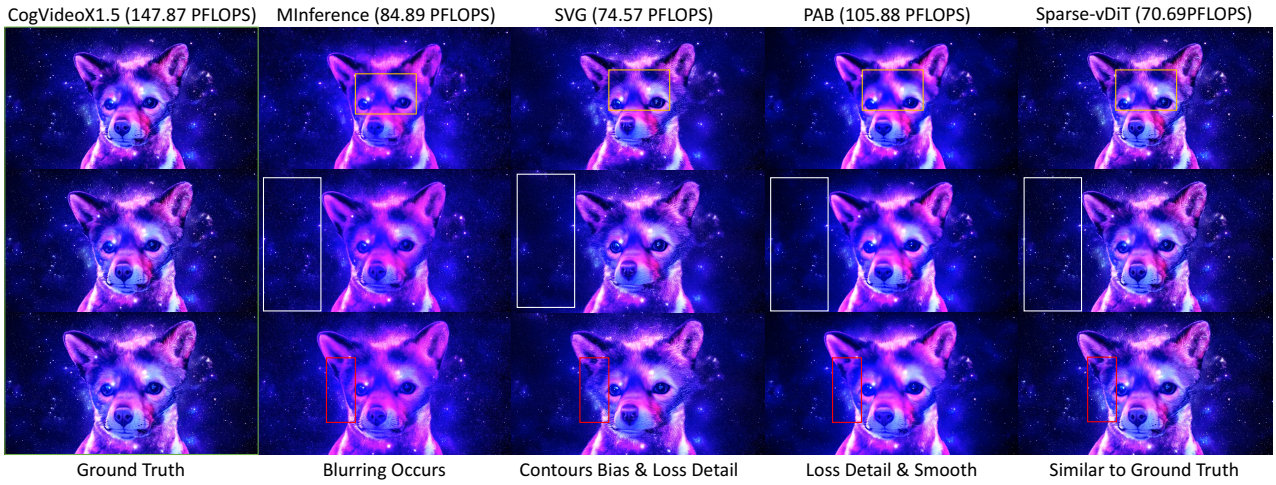

Figure 6: Visual comparison between Sparse-vDiT and the baseline method showing ground truth in green box, differences in blurriness and smoothness in yellow boxes, variations in fine details in white boxes, and contour differences in red boxes.

Hyper-Para.		SSIM ↑	PSNR ↑	LPIPS ↓	ImageQual ↑	SubConsist ↑	Speedup ↑
λ	0	0.8182	24.0864	0.1501	63.37%	92.61%	1.74×
	0.1	0.8180	24.0558	0.1503	63.35%	92.62%	1.73×
	0.5	0.8212	24.1311	0.1477	63.45%	92.66%	1.76×
	1	0.8203	24.0946	0.1479	63.37%	92.58%	1.73×
ϵ	0.5	0.8512	25.4929	0.1219	63.26%	92.60%	1.68×
	1	0.8212	24.1311	0.1477	63.45%	92.66%	1.76×
	3	0.7883	22.7048	0.1785	63.34%	92.66%	1.81×
	5	0.7716	22.0171	0.1947	63.27%	92.45%	1.87×
	10	0.7399	20.8411	0.2231	63.30%	92.49%	1.91×

Table 3: Ablation study on the effects of hyperparameters λ and ϵ in Sparse-vDiT.

computation in more temporally sensitive heads.

Visualization. Figure 6 shows a visual comparison between the video results generated by Sparse-vDiT and those from the top three baseline methods. We observe that MInference produces blurry results, while PAB shows oversmoothing, as indicated by the yellow box in the first row. Both SVG and PAB lose some fine details, as shown in the white box in the second row. For object contours, SVG exhibits a slight misalignment, as indicated by the red box in the third row. In contrast, our method remains closely aligned with the pretrained model in all these aspects.

Computational Efficiency. Sparse-vDiT aims to achieve significant inference acceleration without sacrificing output quality. On CogVideoX1.5, it reduces computational cost by 52.2%, from 147.87 to 70.69 PFLOPS, and on HunyuanVideo, by 57.9%, from 612.37 to 257.09 PFLOPS— the lowest among all methods, demonstrating the efficacy of our sparsity strategy. In terms of latency, Sparse-vDiT reduces inference time from 901 to 511 seconds on CogVideoX1.5 and from 3166 to 1715 seconds on HunyuanVideo, crucial for time-sensitive applications. Additionally, Sparse-vDiT achieves the highest speedup ratios: 1.76× on CogVideoX1.5 and 1.85× on HunyuanVideo, outperforming all baselines.

5.3 Ablation Study

There are two hyperparameters in Sparse-vDiT, λ and ϵ. The parameter λ controls the trade-off between efficiency and quality, while ϵ regulates the overall sparsity of the vDiT. This section analyzes the effects of these methods based on experiments on CogVideoX1.5 with the full VBench.

Quality-Efficiency trade-off. With ϵ fixed at its optimal value of 1, we vary λ across 0, 0.1, 0.5, and 1. Results are reported in Table 3. Comparisons across metrics show that both $\lambda = 0.5$ and $\lambda = 1$ yield strong generation quality. However, λ=1 is less efficient. Thus, $\lambda = 0.5$ offers a better trade-off between generation quality and efficiency, and is used as the default configuration in Table 2.

Performance under different levels of sparsity. Fixing λ at 0.5, we evaluate ϵ values of 0.5, 1, 3, 5, and 10. Table 3 illustrates that increasing ϵ leads to greater sparsity, resulting in higher acceleration. For instance, $\epsilon = 10$ achieves a speedup of 1.91×. However, higher sparsity can impair the quality of generation, as reflected in performance metrics. Notably, at $\epsilon = 5$, Sparse-vDiT achieves a 1.87× speedup while still outperforming the SVG baseline (1.64× speedup). In practice, ϵ can be adjusted to achieve the desired balance between quality and efficiency.

6 Conclusion

We propose Sparse-vDiT, an efficient inference method for vDiT based on structured sparsity. It combines predefined sparsity patterns with an offline diffusion-guided search to assign the most suitable configuration to each attention head. Experiments on CogVideo and HunyuanVideo demonstrate theoretical speedups of 2.09× and 2.38×, and actual speedups of 1.76× and 1.85×, respectively. Despite the acceleration, video quality remains comparable to that of the original models, with PSNR values of 24.13 and 27.09. These results highlight Sparse-vDiT's ability to balance efficiency and generation quality, establishing a new state-of-the-art for sparsity-based vDiT acceleration.

Acknowledgments

This work is supported by Shanghai Science and Technology Commission Explorer Program Project (24TS1401300), Shanghai Natural Science Foundation (No. 23ZR1402900), Shanghai Municipal Science and Technology Major Project (No.2021SHZDZX0103). The computations in this research were performed using CFFF platform of Fudan University.

References

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

Blattmann, A.; Dockhorn, T.; Kulal, S.; Mendelevitch, D.; Kilian, M.; Lorenz, D.; Levi, Y.; English, Z.; Voleti, V.; Letts, A.; et al. 2023. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*.

Castells, T.; Song, H.-K.; Kim, B.-K.; and Choi, S. 2024. Ld-pruner: Efficient pruning of latent diffusion models using task-agnostic insights. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 821–830.

Chen, P.; Shen, M.; Ye, P.; Cao, J.; Tu, C.; Bouganis, C.-S.; Zhao, Y.; and Chen, T. 2024. Delta-DiT: A Training-Free Acceleration Method Tailored for Diffusion Transformers. *arXiv preprint arXiv:2406.01125*.

Child, R.; Gray, S.; Radford, A.; and Sutskever, I. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Ding, H.; Li, D.; Su, R.; Zhang, P.; Deng, Z.; Stoica, I.; and Zhang, H. 2025. Efficient-vDiT: Efficient Video Diffusion Transformers With Attention Tile. *arXiv preprint arXiv:2502.06155*.

Esser, P.; Kulal, S.; Blattmann, A.; Entezari, R.; Müller, J.; Saini, H.; Levi, Y.; Lorenz, D.; Sauer, A.; Boesel, F.; Podell, D.; Dockhorn, T.; English, Z.; Lacey, K.; Goodwin, A.; Marek, Y.; and Rombach, R. 2024. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis. arXiv:2403.03206.

Fang, G.; Ma, X.; and Wang, X. 2023. Structural pruning for diffusion models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23. Red Hook, NY, USA: Curran Associates Inc.

Hassani, A.; Walton, S.; Li, J.; Li, S.; and Shi, H. 2023. Neighborhood attention transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6185–6194.

He, H.; Zhang, Y.; Lin, L.; Xu, Z.; and Pan, L. 2025. Pre-Trained Video Generative Models as World Simulators. *arXiv preprint arXiv:2502.07825*.

He, Y.; Xia, M.; Chen, H.; Cun, X.; Gong, Y.; Xing, J.; Zhang, Y.; Wang, X.; Weng, C.; Shan, Y.; et al. 2023. Animate-a-story: Storytelling with retrieval-augmented video generation. *arXiv preprint arXiv:2307.06940*.

Hu, L. 2024. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8153–8163.

Huang, Z.; He, Y.; Yu, J.; Zhang, F.; Si, C.; Jiang, Y.; Zhang, Y.; Wu, T.; Jin, Q.; Chanpaisit, N.; et al. 2024. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21807–21818.

Jiang, H.; Li, Y.; Zhang, C.; Wu, Q.; Luo, X.; Ahn, S.; Han, Z.; Abdi, A.; Li, D.; Lin, C.-Y.; et al. 2024. Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention. *Advances in Neural Information Processing Systems*, 37: 52481–52515.

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

Lai, X.; Lu, J.; Luo, Y.; Ma, Y.; and Zhou, X. 2025. Flexprefill: A context-aware sparse attention mechanism for efficient long-sequence inference. *arXiv preprint arXiv:2502.20766*.

Lin, B.; Ge, Y.; Cheng, X.; Li, Z.; Zhu, B.; Wang, S.; He, X.; Ye, Y.; Yuan, S.; Chen, L.; et al. 2024. Open-sora plan: Open-source large video generation model. *arXiv preprint arXiv:2412.00131*.

Lin, S.; Xia, X.; Ren, Y.; Yang, C.; Xiao, X.; and Jiang, L. 2025. Diffusion adversarial post-training for one-step video generation. *arXiv preprint arXiv:2501.08316*.

Liu, F.; Zhang, S.; Wang, X.; Wei, Y.; Qiu, H.; Zhao, Y.; Zhang, Y.; Ye, Q.; and Wan, F. 2024. Timestep Embedding Tells: It's Time to Cache for Video Diffusion Model. *arXiv preprint arXiv:2411.19108*.

Liu, S.; Tan, Z.; and Wang, X. 2024. CLEAR: Conv-Like Linearization Revs Pre-Trained Diffusion Transformers Up. *arXiv preprint arXiv:2412.16112*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Lv, Z.; Si, C.; Pan, T.; Chen, Z.; Wong, K.-Y. K.; Qiao, Y.; and Liu, Z. 2025. Dual-Expert Consistency Model for Efficient and High-Quality Video Generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14983–14993.

Lv, Z.; Si, C.; Song, J.; Yang, Z.; Qiao, Y.; Liu, Z.; and Wong, K.-Y. K. 2024. Fastercache: Training-free video diffusion model acceleration with high quality. *arXiv preprint arXiv:2410.19355*.

Ma, X.; Fang, G.; and Wang, X. 2024. Deepcache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 15762–15772.

Meng, F.; Liao, J.; Tan, X.; Shao, W.; Lu, Q.; Zhang, K.; Cheng, Y.; Li, D.; Qiao, Y.; and Luo, P. 2024. Towards world simulator: Crafting physical commonsense-based benchmark for video generation. *arXiv preprint arXiv:2410.05363*.

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

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Shen, M.; Chen, P.; Ye, P.; Xia, G.; Chen, T.; Bouganis, C.-S.; and Zhao, Y. 2024. MD-DiT: Step-aware Mixture-of-Depths for Efficient Diffusion Transformers. In *Adaptive Foundation Models: Evolving AI for Personalized and Efficient Learning*.

Sun, W.; Tu, R.-C.; Ding, Y.; Jin, Z.; Liao, J.; Liu, S.; and Tao, D. 2025. VORTA: Efficient Video Diffusion via Routing Sparse Attention. *arXiv preprint arXiv:2505.18809*.

Tian, R.; Dai, Q.; Bao, J.; Qiu, K.; Yang, Y.; Luo, C.; Wu, Z.; and Jiang, Y.-G. 2024. REDUCIO! Generating 1024$times$1024 Video within 16 Seconds using Extremely Compressed Motion Latents. *arXiv preprint arXiv:2411.13552*.

Tillet, P.; Kung, H.-T.; and Cox, D. 2019. Triton: an intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, 10–19.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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

Wang, J.; Pu, J.; Qi, Z.; Guo, J.; Ma, Y.; Huang, N.; Chen, Y.; Li, X.; and Shan, Y. 2024. Taming rectified flow for inversion and editing. *arXiv preprint arXiv:2411.04746*.

Wang, Z.; and Bovik, A. C. 2002. A universal image quality index. *IEEE signal processing letters*, 9(3): 81–84.

Wu, J.; Hou, L.; Yang, H.; Tao, X.; Tian, Y.; Wan, P.; Zhang, D.; and Tong, Y. 2025. VMoBA: Mixture-of-Block Attention for Video Diffusion Models. *arXiv preprint arXiv:2506.23858*.

Wu, J.; Wang, H.; Shang, Y.; Shah, M.; and Yan, Y. 2024. Ptq4dit: Post-training quantization for diffusion transformers. *arXiv preprint arXiv:2405.16005*.

Xi, H.; Yang, S.; Zhao, Y.; Xu, C.; Li, M.; Li, X.; Lin, Y.; Cai, H.; Zhang, J.; Li, D.; et al. 2025. Sparse VideoGen: Accelerating Video Diffusion Transformers with Spatial-Temporal Sparsity. *arXiv preprint arXiv:2502.01776*.

Xiao, G.; Tang, J.; Zuo, J.; Guo, J.; Yang, S.; Tang, H.; Fu, Y.; and Han, S. 2024. Duoattention: Efficient long-context llm inference with retrieval and streaming heads. *arXiv preprint arXiv:2410.10819*.

Xiao, G.; Tian, Y.; Chen, B.; Han, S.; and Lewis, M. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

Xie, E.; Chen, J.; Chen, J.; Cai, H.; Tang, H.; Lin, Y.; Zhang, Z.; Li, M.; Zhu, L.; Lu, Y.; et al. 2024. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*.

Xing, J.; Xia, M.; Zhang, Y.; Chen, H.; Yu, W.; Liu, H.; Liu, G.; Wang, X.; Shan, Y.; and Wong, T.-T. 2024. Dynamicrafter: Animating open-domain images with video diffusion priors. In *European Conference on Computer Vision*, 399–417. Springer.

Yang, S.; Xi, H.; Zhao, Y.; Li, M.; Zhang, J.; Cai, H.; Lin, Y.; Li, X.; Xu, C.; Peng, K.; et al. 2025. Sparse VideoGen2: Accelerate Video Generation with Sparse Attention via Semantic-Aware Permutation. *arXiv preprint arXiv:2505.18875*.

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

Yuan, Z.; Zhang, H.; Pu, L.; Ning, X.; Zhang, L.; Zhao, T.; Yan, S.; Dai, G.; and Wang, Y. 2024. Ditfastattn: Attention compression for diffusion transformer models. *Advances in Neural Information Processing Systems*, 37: 1196–1219.

Zhang, C.; Feng, C.; Yan, F.; Zhang, Q.; Zhang, M.; Zhong, Y.; Zhang, J.; and Ma, L. 2025a. InstructVEdit: A Holistic Approach for Instructional Video Editing. *arXiv preprint arXiv:2503.17641*.

Zhang, H.; Su, R.; Yuan, Z.; Chen, P.; Fan, M. S. Y.; Yan, S.; Dai, G.; and Wang, Y. 2025b. DiTFastAttnV2: Head-wise Attention Compression for Multi-Modality Diffusion Transformers. *arXiv preprint arXiv:2503.22796*.

Zhang, P.; Chen, Y.; Huang, H.; Lin, W.; Liu, Z.; Stoica, I.; Xing, E. P.; and Zhang, H. 2025c. Faster video diffusion with trainable sparse attention. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Zhang, P.; Chen, Y.; Su, R.; Ding, H.; Stoica, I.; Liu, Z.; and Zhang, H. 2025d. Fast video generation with sliding tile attention. *arXiv preprint arXiv:2502.04507*.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.

Zhao, M.; Chen, P.; Yu, C.; Wen, Y.; Tan, X.; and Chen, T. 2025. Pioneering 4-Bit FP Quantization for Diffusion Models: Mixup-Sign Quantization and Timestep-Aware Fine-Tuning. arXiv:2505.21591.

Zhao, X.; Jin, X.; Wang, K.; and You, Y. 2024. Real-time video generation with pyramid attention broadcast. *arXiv preprint arXiv:2408.12588*.

Zhu, L.; Huang, Z.; Liao, B.; Liew, J. H.; Yan, H.; Feng, J.; and Wang, X. 2024. Dig: Scalable and efficient diffusion models with gated linear attention. *arXiv preprint arXiv:2405.18428*.