RhymeFlow: Training-Free Acceleration for Video Generation with Asynchronous Denoising Flow Scheduling

Chensheng Dai[1⋆], Shengjun Zhang[1⋆], Yifan Li[1], Zhang Zhang[1], Zheng Zhu[2], and Yueqi Duan[1✉]

[1] Tsinghua University, Beijing, China
{dcs23, zhangsj23}@mails.tsinghua.edu.cn
[2] GigaAI, Beijing, China

Abstract. Video generation models based on Diffusion Transformers (DiTs) have achieved remarkable performance in video synthesis, yet they suffer from high inference latency and computational costs due to the quadratic complexity of 3D attention. Existing acceleration methods primarily reduce computational complexity within each individual denoising steps through techniques such as sparse attention and KV-caching. However, they rigidly adhere to the inherent constraint of the standard diffusion pipeline: every frame in the target video sequence must be subjected to a complete, dense denoising process across all diffusion timesteps. We observe that due to the corresponding contents and motions among adjacent frames, when keyframes with critical semantic transitions are anchored, the intermediate states of others often follow more predictable trajectories, which indicates that such uniform, dense denoising process is inherently redundant for natural video data. To this end, we introduce **RhymeFlow**, a training-free framework that decouples the denoising trajectories of different frames. Specifically, we first identify a sparse set of pivotal key frames that dominate the latent semantic evolution. Then, only these keyframes undergo dense, step-by-step denoising to ensure structural integrity, while non-keyframes progressively skip denoising steps to minimize computational cost. Since skipped intermediate states of non-keyframes break the temporal coherence in keyframe denoising steps, leading to visual degradation, we further introduce a latent trajectory projection module, which enables keyframes to interact with a complete and temporally consistent sequence representation. Extensive experiments on current DiT-based video generation models demonstrate our method outperforms existing acceleration baselines with higher inference speed and better visual quality.

Keywords: Video Diffusion · Acceleration · Asynchronous Denoising

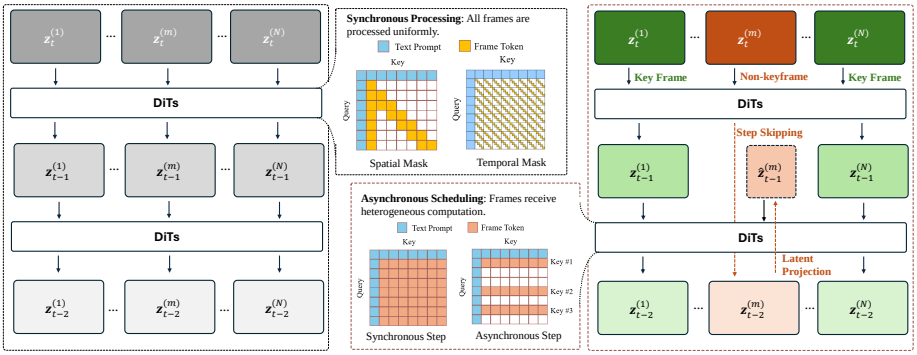
⋆ Equal contribution.
✉ Corresponding author.

1 Introduction

Video diffusion models [5, 12, 39], particularly those based on DiT architectures [10, 26, 28], have achieved remarkable success in high-fidelity video synthesis. However, their practical deployment is hampered by a inherent bottleneck: the quadratic complexity of 3D spatiotemporal attention combined with dozens of denoising steps creates prohibitive inference costs [7, 19]. To mitigate this, training-free acceleration methods [24, 33, 42] have garnered significant attention due to their compatibility with pre-trained models, which broadly fall into three categories: KV-cache management [16, 25, 27], model compression via quantization or token pruning [2, 8, 44], and sparse attention mechanisms [36, 37, 40]. While these approaches effectively lower the complexity of every individual denoising step, every video frame is subjected to a dense, stepwise denoising procedure across the full set of diffusion timesteps.

In this work, we observe that such uniform computational allocation is inherently redundant. Due to the overlapping visual contents and continuous motion trajectory among neighbor frames, video sequences natively exhibit strong spatiotemporal coherence. Building upon this, we find that once a sparse set of pivotal keyframes—those capturing critical structural or semantic transitions—is anchored via step-by-step full attention, the intermediate states of adjacent non-keyframes often follow highly predictable trajectories in the latent space. Driven by this predictability, allowing non-keyframes to appropriately skip certain denoising steps does not incur noticeable degradation in visual quality. This observation indicates that applying a uniform, dense denoising schedule to every single frame is fundamentally unnecessary. Consequently, designing heterogeneous, frame-specific denoising schedules presents a promising, yet largely untapped, potential for video diffusion acceleration.

To this end, we introduce **RhymeFlow**, a training-free framework that establishes a new paradigm for video diffusion acceleration with **Asynchronous Denoising Flow Scheduling**. Our core insight is to decouple the denoising trajectories of individual frames and assign heterogeneous schedules accordingly. Specifically, we first identify a sparse set of pivotal keyframes by detecting semantic shifts in the latent space, ensuring that frames capturing critical transitions are prioritized for high-fidelity generation. Second, we assign heterogeneous denoising schedules: only these keyframes execute the complete, step-by-step denoising process to ensure structural integrity and high visual fidelity. In contrast, non-keyframes skip designated denoising steps to minimize computational costs. Third, recognizing that skipped intermediate states invariably break the temporal coherence required for 3D attention, we introduce a lightweight latent trajectory projection module. This mechanism analytically estimates the skipped intermediate latents of non-keyframes, which ensures keyframes can attend to a complete and temporally consistent sequence representation at every step, thereby preserving global consistency without incurring the penalty of full network evaluation.



(a) Conventional Synchronous Scheduling (b) **RhymeFlow**: Asynchronous Denoising Scheduling

Fig. 1: Different sparse attention-based acceleration methods. (a) Synchronous Scheduling: Conventional acceleration methods operate within a synchronous framework where all frames are jointly denoised at each timestep. These approaches achieve efficiency by exploiting intra-step sparsity through techniques such as spatial, temporal, or semantic attention masking. (b) Our Asynchronous Scheduling: Our work introduces an orthogonal acceleration dimension. Instead of processing all frames uniformly, we differentiate between keyframes and non-keyframes. Keyframes undergo a full, step-by-step denoising process to preserve high fidelity, while non-keyframes are updated asynchronously, effectively skipping computation at certain timesteps.

2 Related Works

2.1 Video Diffusion Models

Recent years have witnessed the remarkable success of diffusion models [11, 35] in generating high-fidelity and diverse content. This paradigm has been successfully extended from images to the video domain, with Diffusion Transformers (DiTs) [10, 26, 28] emerging as the dominant architecture. State-of-the-art open-source models like Wan 2.1 [34] , CogVideoX [13, 41], and HunyuanVideo [18], as well as closed-source systems like Sora [3] and Kling, have demonstrated unprecedented capabilities in generating temporally consistent and visually stunning videos from text or image prompts. These models typically adapt the 2D spatial attention mechanism to a more computationally intensive 3D spatiotemporal attention to capture both the appearance within frames and the motion across them [1, 18, 41].

However, this success comes at a great cost. The inference process of these models is notoriously slow, often requiring hundreds of sequential passes through a massive transformer network. The 3D attention mechanism, in particular, exhibits quadratic complexity with respect to the number of tokens (pixels $\times$ frames), making the generation of long, high-resolution videos prohibitively expensive. This significant computational barrier motivates a strong need for efficient video generation techniques.

2.2 Efficient Video Generation

To combat the prohibitive inference costs, researchers have explored several avenues for acceleration. We categorize these efforts and position our work in relation to them.

Decreasing the Denoising Steps. A primary strategy for accelerating diffusion models is to reduce the number of function evaluations (NFE) required during sampling. Early models based on stochastic differential equations (SDEs) required thousands of steps. The introduction of ordinary differential equation (ODE) solvers, such as Denoising Diffusion Implicit Models (DDIM) [31], and more advanced solvers like DPM-Solver [22, 23, 48], significantly reduced the required steps. More recently, methods like Consistency Models [32] and progressive distillation [30] have been proposed to enable high-quality synthesis in just a few steps. However, these approaches have two major limitations in the context of large-scale video models. First, many of them, such as distillation and consistency training, require costly retraining or fine-tuning, which is impractical for models with billions of parameters. Second, they apply a uniform step-reduction strategy to all frames, treating them as equally important throughout the denoising process.

Diffusion Model Compression. An orthogonal line of research focuses on compressing the model itself to reduce the computational cost of each forward pass. This includes quantization techniques that reduce the bit-width of weights and activations, such as the W8A8 strategy in Q-Diffusion [21], or even more aggressive 4-bit schemes [20]. Other approaches involve designing more efficient model architectures from the ground up or using more compact autoencoders.

Training-free Acceleration via Caching. A recent and related trend in training-free acceleration involves caching and reusing intermediate results [4, 25, 27, 49, 50]. Methods like DeepCache [27] and FasterCache [25] leverage the high feature similarity between adjacent denoising steps, avoiding redundant computation by reusing cached features (e.g., from the UNet's deeper layers or KV-cache). These methods are training-free and effectively reduce computation.

Sparse Attention in LLMs and Video. The quadratic complexity of the attention mechanism has motivated extensive research into sparse attention, particularly in Large Language Models (LLMs). Methods like StreamingLLM [38] and H2O [46] identify that attention scores are often concentrated on a small subset of "heavy-hitter" or local tokens. This concept has been adapted for video diffusion. For instance, the proposed frameworks SVG [36] and SAP [40], perform an in-depth analysis of attention patterns in Video DiTs, classifying heads into Spatial and Temporal types and applying structured sparse masks accordingly. While highly effective, these methods focus on optimizing the computation within a single denoising step by reducing the number of token-to-token interactions. Instead of making the attention matrix sparse, RhymeFlow skips the entire forward pass for certain frames at certain steps, optimizing the denoising trajectory over time. This makes our asynchronous denoising flow scheduling paradigm orthogonal to intra-step sparsity.

3 Method

In this section, we preform a detailed introduction of **RhymeFlow**, a training-free acceleration framework for video diffusion models. Our approach re-envisions the denoising process by assigning different denoising schedule to different frames. We first classify keyframes and non-keyframes based on the latent semantic evolution. After the warm-up denoising stages, we propose a progressive skip strategy, where non-keyframes skip less steps at high noise levels and skip more steps at low noise levels. To maintain the temporal coherence of natural video data, we further present a latent trajectory projection mechanism to enable keyframes to interact with non-keyframes at the denoising steps that they skip.

3.1 Sequential Keyframe Selection

To alleviate the prohibitive inference latency and excessive computational cost of prevailing video generation models while preserving the temporal continuity and semantic integrity of the generated sequence, we design a lightweight, content-aware sequential key frame selection scheme to identify representative frames that dominate the semantic transitions of the entire video.

Supposing that the video representation in the diffusion latent space consist of N latent frames, we denote the noisy latent of the i-th frame at the current denoising timestep t as $\mathbf{z}_t^{(i)}(1 \leq t \leq N)$. Since the raw noisy latents are heavily corrupted by noise and lack distinct structural information, we perform a single-step denoising prediction to estimate the fully denoised clean latent, denoted as $\hat{\mathbf{z}}_0^{(i)}$. These predicted clean latents provide a structurally clearer and more robust proxy for evaluating frame-wise correlations.

Based on the clean latent frames, we conduct the selection procedure in a strictly sequential manner along the temporal axis of the target video. Previous studies [6, 9, 17, 29] have comprehensively demonstrated that the initial frame of a video sequence acts as the fundamental semantic and visual anchor for subsequent frame synthesis, and plays an irreplaceable role in maintaining long-range temporal consistency and semantic fidelity throughout the generated video. In light of this, we initialize the set of selected key frames $\mathcal{K} = \{\hat{\mathbf{z}}_0^{(i)}\}$ with the initial frame of the target sequence. Subsequently, we traverse all subsequent candidate frames in chronological order. For each candidate latent frame $\hat{\mathbf{z}}_0^{(j)}$, we compute the pairwise similarity $\mathrm{sim}(\hat{\mathbf{z}}_0^{(j)}, \hat{\mathbf{z}}_0^{(k)})$ between the candidate frame and the identified nearest preceding key frame $\hat{\mathbf{z}}_0^{(k)} \in \mathcal{K}$. The update rule of the key frame set is written as:

$$\mathcal{K} \leftarrow \begin{cases} \mathcal{K} \cup \{\hat{\mathbf{z}}_0^{(j)}\}, & \text{if } \psi_{\text{sim}}(\hat{\mathbf{z}}_0^{(j)}, \hat{\mathbf{z}}_0^{(k)})) < \tau \\ \mathcal{K}, & \text{otherwise} \end{cases} \tag{1}$$

where ψ_{sim} is defined as the cosine similarity function, and τ is a pre-defined threshold.

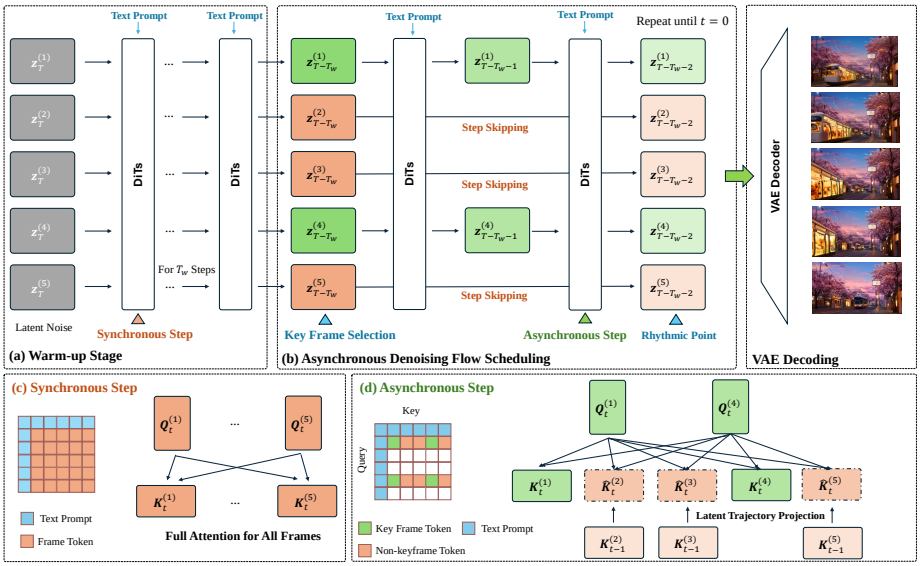


Fig. 2: Pipeline. (a) **Warm-up Stage:** All frames are processed uniformly via synchronous updates for the initial T_w steps. (b) **Asynchronous Scheduling:** After warm-up, keyframes are selected. The pipeline transitions to an asynchronous schedule: keyframes are fully updated to preserve critical details, while non-keyframes are sparsely processed to accelerate generation. (c) **Synchronous Step:** A standard full-attention mechanism is applied during warm-up and at "Rhythmic Points" to synchronize all latent frame representations. (d) **Asynchronous Step:** We introduce latent trajectory projection to efficiently estimate the skipped states of non-keyframes, providing the temporal context required for keyframes to attend to the full sequence.

3.2 Asynchronous Denoising Flow Scheduling

We design a heterogeneous denoising schedule where keyframes and non-keyframes are updated at different time-steps.

Initial Synchronous Warm-up. The generation process commences with a synchronous warm-up stage that spans the initial T_w denoising steps. This foundational phase is critical, as the early steps of the diffusion process are primarily responsible for establishing the global compositional structure, motion priors, and color palette of the nascent video. Bypassing full computation during this period can introduce severe, often irrecoverable, artifacts and quality degradation. Consequently, for the initial T_w steps (i.e., from timestep $t = T$ down to $t = T - T_w$), all N latent frames are updated simultaneously. The update for each latent frame $\mathbf{z}_t^i$ to $\mathbf{z}_{t-1}^i$ proceeds using the standard, unmodified denoising function with full attention, as illustrated in Figure 2 (a).

Progressive Asynchronous Scheduling. The reverse diffusion process exhibit strong non-uniformity along the timestep axis. The early stages, character-

ized by high noise levels (large t), are dedicated to establishing the fundamental, low-frequency structure of the video, such as global composition and object forms. In this phase, the denoising trajectory is highly sensitive and less predictable, making these steps foundational to the final output quality. Conversely, the later stages (small t) primarily involve the refinement of high-frequency details, where the latent trajectory becomes significantly smoother and thus more predictable.

A static step-skipping stride cannot adapt to optimally navigate this evolving dynamic. An aggressive stride risks irrecoverable damage to the foundational structure in the early stages, while a conservative stride would forgo significant efficiency gains in the predictable later stages. To strike a deliberate balance between computational efficiency and generative fidelity, we introduce a Progressive Asynchronous Scheduling strategy. This approach adaptively increases the skipping stride for non-keyframes as the denoising process advances.

We implement this strategy with a simple, yet effective, piecewise schedule for the skip stride, n_{skip}, as a function of the current timestep t:

$$n_{skip}(t) = \begin{cases} n_{small}, & \text{if } T_{mid} < t \leq T - T_w \\ n_{large}, & \text{if } t \leq T_{mid} \end{cases} \tag{2}$$

where T_{mid} is a mid-point hyperparameter (e.g., $T/2$), and $n_{small} < n_{large}$ are integer stride values (e.g., $n_{small} = 2$, $n_{large} = 3$). This schedule intelligently allocates computational resources where they are most critical, thereby achieving a near-optimal trade-off between acceleration and the perceptual quality of the final generated video.

Asynchronous Denoising Step. Following the warm-up stage, RhymeFlow transitions into an asynchronous denoising schedule characterized by heterogeneous computational trajectories for different frame categories:

- **Keyframes** are updated via a standard single-step denoising process, advancing their state from $\mathbf{z}_t^{(k)}$ to $\mathbf{z}_{t-1}^{(k)}$. This ensures they maintain fidelity.
- **Non-keyframes** are projected forward multiple steps in a single computation. Their state is advanced directly from $\mathbf{z}_t^{(k)}$ to $\mathbf{z}_{t-n_{\text{skip}}}^{(k)}$, thereby bypassing the intermediate denoising steps and accelerating the generation process.

Since keyframes and non-keyframes progress with staggered strides, they periodically coincide at specific timesteps, which we define as *Rhythmic Points*. At these points, the framework executes a **full synchronous update**, wherein the latent representations of all frames engage in global 3D attention. This periodic re-synchronization provides a comprehensive contextual foundation, critical for maintaining global video coherence. Moreover, Rhythmic Points function as computational checkpoints where the trajectories of non-keyframes are recalibrated against the high-fidelity keyframe anchors. Such a mechanism effectively bounds the error accumulation inherent in prolonged step-skipping, ensuring that acceleration does not compromise temporal consistency or perceptual quality.

However, a fundamental challenge arises during the intermediate steps $\tau \in [t - n_{skip} + 1, t - 1]$, a keyframe $\mathbf{z}_\tau^{(k)} \in \mathcal{K}$ must be updated. This update requires attention context from all other frames at the target timestep τ. The central challenge arises here: a non-keyframe $\mathbf{z}_\tau^{(j)}$ has been fast-forwarded to $t - n_{skip}$, meaning its latent states for any intermediate timestep τ are not explicitly computed and do not exist.

Latent Trajectory Projection. Inspired by the linear nature of ODE trajectories in Rectified Flow, we analytically project an approximation of the missing latent state. For a non-keyframe j that was last updated at t_{start} to obtain $\mathbf{z}_{t_{end}}^{(j)}$, we can project its latent state $\hat{\mathbf{z}}_\tau^{(j)}$ at the intermediate time t via linear interpolation in the latent space:

$$\hat{\mathbf{z}}_\tau^{(j)} = (1 - \alpha) \cdot \mathbf{z}_{t_{start}}^{(j)} + \alpha \cdot \mathbf{z}_{t_{end}}^{(j)}, \quad \alpha = \frac{t_{start} - \tau}{t_{start} - t_{end}} \tag{3}$$

This projection is computationally trivial yet provides a robust estimate of the intermediate state. Following this latent projection, we can then compute the required Key and Value vectors, $\hat{\mathbf{K}}_\tau^{(j)}$ and $\hat{\mathbf{V}}_\tau^{(j)}$, on-the-fly.

With this mechanism, we define the logic for a keyframe update: it attends to the true states of other keyframes and the projected states of non-keyframes. A non-keyframe only performs its update at its designated, less frequent steps.

KV-Cache Management. Our asynchronous denoising schedule necessitates a specialized KV-cache mechanism fundamentally distinct from causal autoregressive approaches. We introduce a **per-layer rolling cache** $\mathcal{C}_\ell^{(f)}$ for each DiT layer ℓ that stores the two most recent attention outputs for each non-keyframe $\hat{\mathbf{z}}$: $\mathbf{h}_{t_1}^{(\ell,\hat{\mathbf{z}})}$ and $\mathbf{h}_{t_2}^{(\ell,\hat{\mathbf{z}})}$ at timesteps $t_1 > t_2$, along with their temporal indices. Critically, we cache the post-attention hidden states rather than input key-value pairs, as these representations have already integrated full-frame contextual information. When a keyframe requires context from a non-keyframe at a skipped timestep $\tau \in (t_1, t_2)$, we employ latent trajectory projection. These interpolated representations serve as transient key-value providers for the current attention operation and are immediately discarded thereafter, eliminating persistent storage of intermediate states.

3.3 Orthogonality to Intra-step Sparsity

A notable advantage of RhymeFlow is its orthogonality to existing intra-step sparse attention techniques. This orthogonality stems from operating on different dimensions of the computational workload. Methods like SVG [36] target **intra-step efficiency**, optimizing how tokens interact within a single attention computation, while our method addresses **inter-step efficiency**, determining which frames participate in the computation at each denoising step. This fundamental distinction allows the two approaches to be synergistically combined for enhanced acceleration performance.

Table 1: Quantitative evaluation of the efficiency-fidelity trade-off on Wan 2.1. We investigate the impact of warm-up timesteps (T_w) and the number of keyframes (M). Our optimal default configuration is highlighted in bold.

Method		Similarity Metrics			Video Quality		Efficiency	
T_w	M	PSNR ↑	SSIM ↑	LPIPS ↓	SubCon. ↑	ImgQual. ↑	Lat.(s) ↓	Speedup ↑
Original		-	-	-	0.9102	0.6946	993.5	-
6	3	20.998	0.615	0.206	0.8460	0.6099	607.2	1.64×
6	4	22.588	0.617	0.196	0.8478	0.6362	642.0	1.55×
6	5	25.669	0.657	0.184	0.8536	0.6883	682.0	1.46×
8	3	23.742	0.721	0.182	0.8601	0.6344	621.6	1.60×
8	**4**	**26.291**	**0.783**	**0.168**	**0.8831**	**0.6706**	**650.4**	**1.53**×
8	5	27.707	0.812	0.162	0.8896	0.6902	712.8	1.39×
10	3	24.006	0.742	0.169	0.8818	0.6714	630.8	1.57×
10	4	26.636	0.817	0.164	0.8838	0.6819	670.3	1.48×
10	5	28.061	0.816	0.161	0.8911	0.6919	738.1	1.35×

To illustrate, consider an intra-step method like SAP that constructs a token-level sparse attention mask, $\mathbf{M}_{\text{SAP}}$. Concurrently, our framework generates a distinct frame-level mask, $\mathbf{M}_{\text{Rhyme}}$, which dictates the set of active and inactive frames for a given step. A hybrid, highly efficient attention mechanism can be formulated by computing the element-wise product of these two masks:

$$\mathbf{M}_{\text{combined}} = \mathbf{M}_{\text{Rhyme}} \odot \mathbf{M}_{\text{SAP}} \tag{4}$$

The resulting mask, $\mathbf{M}_{\text{combined}}$, imposes a hierarchical sparsity. It first prunes the extensive computations corresponding to skipped non-keyframes (as dictated by $\mathbf{M}_{\text{Rhyme}}$) and then further sparsifies the attention calculations among the remaining active frames based on the token-level patterns in $\mathbf{M}_{\text{SAP}}$. This fusion of inter-step and intra-step strategies unlocks compounded acceleration gains, paving the way for even more efficient video generation models.

4 Experiments

4.1 Experimental Settings

Models. To validate the effectiveness of our proposed framework, we integrate it into two prominent open-source text-to-video (T2V) generation models: Wan2.1-T2V-v1.3B-Diffusers [34] and CogVideoX-v1.5-T2V [41]. Both models are configured to produce videos of 81 frames at 720p resolution. The architectural specifics of these models are noteworthy: in its latent space, Wan 2.1 operates on 21 latent frames, with each frame represented by 3600 tokens post 3D-VAE processing. In contrast, CogVideoX-v1.5 utilizes a 3D full attention mechanism over 11 frames, where each frame corresponds to 4080 tokens after its 3D-VAE.

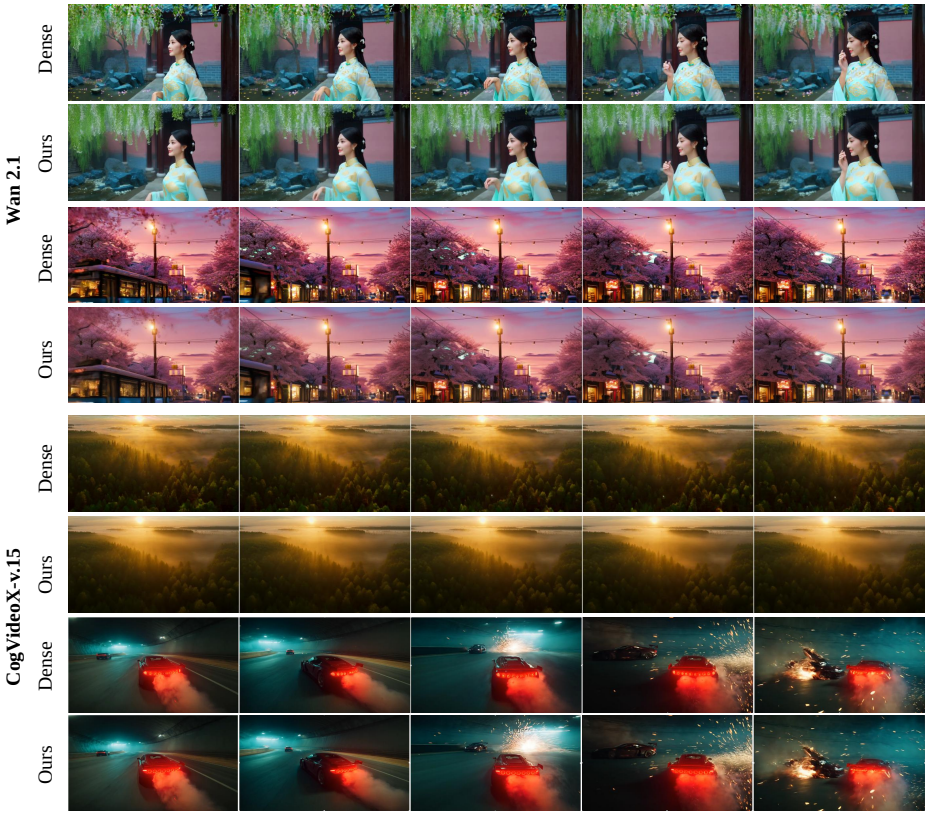


Fig. 3: Qualitative visual comparisons. Frame sequences generated by the standard dense attention baseline versus our RhymeFlow. The top two examples are produced by Wan 2.1, while the bottom two are from CogVideoX-v1.5.

Metrics. We evaluate our method across two primary dimensions: generation fidelity and computational efficiency. To quantify fidelity preservation, we compare the accelerated outputs against the unaccelerated baselines using standard pixel-level and perceptual metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS). Additionally, we assess the visual quality of the generated videos using VBench [14], focusing on critical aspects such as subject consistency and imaging quality. Finally, to demonstrate computational efficiency, we report the average inference latency of diffusion process alongside the corresponding speed-up ratio.

Baselines. To contextualize the performance of RhymeFlow, we conduct a comparative analysis against several state-of-the-art (SOTA) training-free acceleration methods. Our baselines include SpargeAttn [45], MInference [15], Pyramid Attention Broadcast (PAB) [47], Sparse VideoGen (SVG) [36], and Semantic-Aware Permutation (SAP) [40]. For a fair comparison, we adopt the

Table 2: Quality and efficiency benchmarking on Wan 2.1 and CogVideoX-v1.5.

Method	Quality					Efficiency	
	PSNR ↑	SSIM ↑	LPIPS ↓	SubCon. ↑	ImgQual. ↑	Lat.(s) ↓	Speedup ↑
Wan 2.1	-	-	-	0.9102	0.6946	993.5	-
SpargeAttn [45]	20.399	0.613	0.393	0.8632	0.7118	719.7	1.38×
SVG [36]	22.419	0.694	0.290	0.8758	0.6913	708.0	1.40×
SAP [40]	24.454	0.730	0.223	0.8789	0.6837	608.5	1.63×
Ours	**26.291**	**0.783**	**0.168**	**0.8831**	**0.6706**	**650.4**	**1.53**×
Ours + SAP	**24.586**	**0.737**	**0.221**	**0.8792**	**0.6806**	**596.8**	**1.66**×
CogVideoX-v1.5	-	-	-	0.987	0.624	625.0	-
MInference [15]	22.490	0.743	0.264	0.874	0.589	422.3	1.48×
PAB [47]	23.230	0.782	0.145	0.978	0.573	443.3	1.41×
SVG [36]	24.130	0.811	0.171	0.982	0.597	385.8	1.62×
Ours	**26.890**	**0.852**	**0.142**	**0.986**	**0.623**	**351.1**	**1.78**×
Ours + SAP	**25.574**	**0.821**	**0.157**	**0.972**	**0.609**	**323.8**	**1.93**×

Table 3: Quality and efficeincy benchmarking on HunyuanVideo.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	Latency (s) ↓	Speedup ↑
SVG [36]	21.17	0.684	0.392	3459	1.92 ×
SAP [40]	24.64	0.904	0.068	2634	2.52 ×
EasyCache [50]	23.51	0.861	0.119	2850	2.33 ×
DiCache [4]	23.54	0.860	0.114	2814	2.36 ×
VGDFR [43]	19.49	0.779	0.199	3019	2.20 ×
Ours	**26.34**	**0.918**	**0.060**	2939	2.26 ×
Ours+SAP	25.01	0.910	0.068	**2555**	**2.60** ×

official hyperparameter configurations provided by the authors for their respective text-to-video generation tasks. All experiments are executed on a single NVIDIA A800 GPU to ensure a consistent hardware environment.

Implementation Details. Our text-to-video generation experiments utilize prompts sourced from the Penguin Benchmark, which have been refined through the prompt optimization process detailed in VBench [14]. The core hyperparameters are configured as follows: the duration of the warm-up stage, T_w, is set to 8 steps, and the total number of keyframes, M, is 4. For the progressive asynchronous scheduling, the update strides are defined as $n_{\text{small}} = 2$ and $n_{\text{large}} = 3$.

4.2 Trade-off Analysis of Scheduling Parameters

Varying the warm-up duration (T_w) and keyframe budget (M) inherently trades acceleration for generation quality (Table 1). Increasing T_w strengthens the global structure, while a larger M reduces error accumulation via more frequent

Table 4: Ablation on key frame selection strategy.

Setup	PSNR ↑	SSIM ↑	LPIPS ↓	Latency ↓	Speedup ↑
Original	-	-	-	993.5	-
Random	20.630	0.525	0.383	650.0	1.53×
First	19.220	0.515	0.402	651.0	1.53×
Uniform	24.293	0.643	0.183	649.0	1.53×
Ours	26.291	0.783	0.168	650.4	1.53×

Table 5: Ablation study on Architectural Components.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	Latency ↓	Speedup ↑
Original	-	-	-	993.5	-
w/o Progressive	25.399	0.753	0.172	708.0	1.40×
w/o Projection	20.630	0.525	0.383	622.2	1.60×
Ours	26.291	0.783	0.168	650.4	1.53×

high-fidelity temporal anchors. For instance, at a fixed $T_w = 8$, raising M from 3 to 5 boosts PSNR from 23.742 to 27.707 and SSIM from 0.721 to 0.812, though speedup drops from 1.60× to 1.39×. We identify $T_w = 8$ and $M = 4$ as the optimal default configuration. This setting preserves strong fidelity (ImgQual. 0.6706, SubCon. 0.8831) comparable to the dense model, while delivering a substantial 1.53× inference speedup (650.4s vs. 993.5s).

4.3 Comparison with State-of-the-Art Methods

We benchmark RhymeFlow against several training-free acceleration methods on Wan 2.1 [34] and CogVideoX-v1.5 [41]. To ensure fairness, baselines are selected based on official architectural compatibility (e.g., SAP [40] on Wan 2.1; MInference [15] and PAB [47] on CogVideoX-v1.5).

Quantitative Evaluation. As detailed in Table 2, RhymeFlow establishes a superior efficiency-quality trade-off. On Wan 2.1, it achieves a high PSNR of 26.291 and SSIM of 0.783 with a 1.53× speedup, outperforming intra-step (SpargeAttn [45]) and standalone (SVG [36]) methods. On CogVideoX-v1.5, it attains a leading 1.78× speedup while preserving a 0.986 SubConsistency. Furthermore, on high-dynamic HunyuanVideo [18] clips (Table 3), our method outperforms cache-based alternatives (EasyCache [50], DiCache [4]) across all visual metrics. Combined with SAP (Ours+SAP), our approach achieves the lowest latency and highest speedup.

Orthogonality & Synergistic Combination. Our inter-step scheduling is inherently orthogonal to intra-step sparse attention methods (Section 3.3). We evaluate a hybrid *Ours + SAP* approach, relaxing SAP's [40] sparsity ratio from 0.3 to 0.5 to avoid information bottlenecks caused by concurrent acceleration.

Table 6: Ablation study on KV-Cache management.

Method	Latency (s) ↓	Speedup ↑	Average Memory (GB) ↓	Peak Memory (GB) ↓
Original	993.5	1.00×	24.4	44.3
w/o KV-Cache	653.6	1.52×	27.1	49.9
Ours	650.4	1.53×	21.5	42.6

Table 7: Latent-state analysis against dense sampler on Wan 2.1.

Variant	PSNR↑	$\mathcal{E}_{\text{keyframe}}$ ↓	$\mathcal{E}_{\text{non-keyframe}}$ ↓	$\mathcal{E}_{\text{projected-nonkeyframe}}$ ↓	Speedup↑
Ours (full)	**27.70**	**0.0019**	**0.0022**	**0.0018**	1.44×
w/o latent projection	24.44	0.0190	0.0029	0.0240	1.55×
w/ keyframe-only attention	24.37	0.0043	0.0045	0.0490	**1.69**×

This synergy yields the fastest speeds (1.66× on Wan 2.1, 1.93× on CogVideoX-v1.5) with better quality than the standalone SAP baseline.

Qualitative Results. Figure 3 visually confirms our quantitative superiority. Across diverse scenarios—including human portraits, cityscapes, and high-dynamic motion—RhymeFlow flawlessly maintains the intricate textures, lighting, and motion dynamics of the dense baseline, avoiding the flickering or blurring artifacts common in aggressive acceleration.

4.4 Ablation Study

We conduct extensive ablations to validate our design choices on Wan 2.1 [34].

Keyframe Selection. We compare our content-aware strategy against heuristic allocations (*Random*, *First*, and *Uniform*) with $M = 7$ (Table 4). While *Uniform* provides basic temporal coverage (PSNR 24.293), it ignores semantic shifts. By dynamically placing anchors where substantial latent changes occur, our method boosts SSIM (0.643 → 0.783) and lowers LPIPS (0.183 → 0.168).

Architectural Components & Projection. Table 5 isolates our core designs. Replacing progressive scheduling with a fixed skip (*w/o Progressive*) diminishes both speedup (1.53× → 1.40×) and PSNR (26.291 → 25.399). Omitting trajectory projection (*w/o Projection*) deprives the model of dense temporal context, triggering a severe quality collapse (PSNR drops to 20.630). As plotted in Figure 4 and Table 7, the near-straight trajectories of flow matching keep linear-projection errors small; removing this projection or non-keyframe context sharply increases latent errors.

Efficacy of KV-Cache Management. Table 6 highlights the necessity of our per-layer rolling cache. While native skipping (*w/o KV-Cache*) matches our 1.53× speedup, it retains historical states for future projections, bloating peak VRAM to 49.9 GB (exceeding the dense model's 44.3 GB). Our KV-cache management instantly discards transient features, bounding peak memory to an accessible 42.6 GB without sacrificing acceleration.

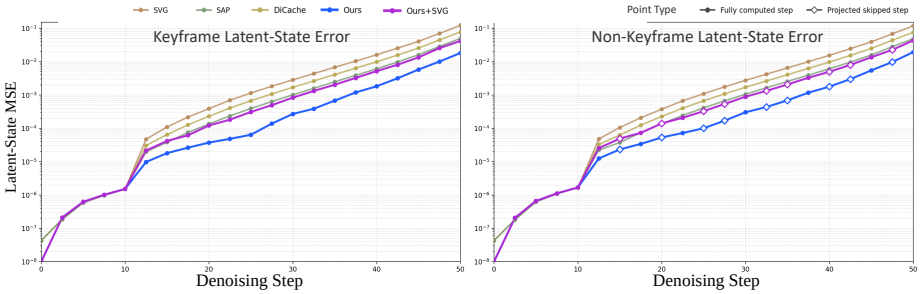


Fig. 4: The error analysis of the latent projection on high-dynamic videos.

Table 8: Double-blind user study results (%).

Metric	Ours v.s. SVG				Ours v.s. SAP				Ours v.s. Dense			
	Ours	Tie	Opp.	p-Value	Ours	Tie	Opp.	p-Value	Ours	Tie	Opp.	p-Value
Visual Quality	**51.2**	18.3	30.5	.025	**74.4**	14.6	11.0	<.001	18.3	**53.7**	28.0	.256
Temporal Coherence	**53.7**	25.6	20.7	<.001	**78.0**	12.2	9.8	<.001	18.3	**58.5**	23.2	.608
Perceptual Preference	**51.2**	23.2	25.6	.006	**79.3**	12.2	8.5	<.001	15.9	**56.1**	28.0	.133

4.5 User Study

We conducted a double-blind user study with 82 participants for human preference evaluation (Table 8). We computed p-values using binomial tests (excluding ties), applying one-sided tests against baselines and a two-sided test against Dense. Our method outperforms both SVG and SAP ($p < .05$) across visual quality, temporal coherence, and perceptual preference. Against the Dense model, differences are statistically insignificant (all $p > .10$), confirming no perceived loss of generation quality compared to the unaccelerated baseline.

5 Conclusion

We present **RhymeFlow**, a training-free acceleration framework for video diffusion models. By replacing uniform denoising with intelligent, content-aware scheduling, RhymeFlow enables scalable and accessible high-resolution video generation. Furthermore, its inherent orthogonality to existing token-level compression methods (e.g., sparse attention, quantization) allows for compounded efficiency gains. Future work will explore learned, scene-adaptive scheduling functions to further optimize this paradigm.

Acknowledgments

This work was supported in part by the Beijing Natural Science Foundation under Grant L252011, by the National Natural Science Foundation of China under Grant 62576185, and by the Young Elite Scientist Sponsorship Program by CAST under Grant YESS20240544.

References

1. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: Vivit: A video vision transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6836–6846 (October 2021)
2. Bolya, D., Hoffman, J.: Token merging for fast stable diffusion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4599–4603 (2023)
3. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., Ramesh, A.: Video generation models as world simulators (2024)
4. Bu, J., Ling, P., Zhou, Y., Wang, Y., Zang, Y., Lin, D., Wang, J.: Dicache: Let diffusion model determine its own cache. arXiv preprint arXiv:2508.17356 (2025)
5. Chen, H., Zhang, Y., Cun, X., Xia, M., Wang, X., Weng, C., Shan, Y.: Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7310–7320 (2024)
6. Chen, J., Li, Z., Liu, Z., Shi, G., Wu, X., Liu, F., Fermuller, C., Feng, B.Y., Aloimonos, Y.: First frame is the place to go for video content customization. arXiv preprint arXiv:2511.15700 (2025)
7. Croitoru, F.A., Hondru, V., Ionescu, R.T., Shah, M.: Diffusion models in vision: A survey. IEEE transactions on pattern analysis and machine intelligence **45**(9), 10850–10869 (2023)
8. Deng, H., Pan, T., Diao, H., Luo, Z., Cui, Y., Lu, H., Shan, S., Qi, Y., Wang, X.: Autoregressive video generation without vector quantization. arXiv preprint arXiv:2412.14169 (2024)
9. Girdhar, R., Singh, M., Brown, A., Duval, Q., Azadi, S., Rambhatla, S.S., Shah, A., Yin, X., Parikh, D., Misra, I.: Factorizing text-to-video generation by explicit image conditioning. In: European Conference on Computer Vision. pp. 205–224. Springer (2024)
10. Gupta, A., Yu, L., Sohn, K., Gu, X., Hahn, M., Li, F.F., Essa, I., Jiang, L., Lezama, J.: Photorealistic video generation with diffusion models. In: European Conference on Computer Vision. pp. 393–411. Springer (2024)
11. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 6840–6851. Curran Associates, Inc. (2020)
12. Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems. vol. 35, pp. 8633–8646. Curran Associates, Inc. (2022)
13. Hong, W., Ding, M., Zheng, W., Liu, X., Tang, J.: Cogvideo: Large-scale pretraining for text-to-video generation via transformers. arXiv preprint arXiv:2205.15868 (2022)
14. 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., Liu, Z.: VBench: Comprehensive benchmark suite for video generative models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)
15. Jiang, H., Li, Y., Zhang, C., Wu, Q., Luo, X., Ahn, S., Han, Z., Abdi, A.H., Li, D., Lin, C.Y., et al.: Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention. Advances in Neural Information Processing Systems **37**, 52481–52515 (2024)

16. Kahatapitiya, K., Liu, H., He, S., Liu, D., Jia, M., Zhang, C., Ryoo, M.S., Xie, T.: Adaptive caching for faster video generation with diffusion transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15240–15252 (2025)
17. Khachatryan, L., Movsisyan, A., Tadevosyan, V., Henschel, R., Wang, Z., Navasardyan, S., Shi, H.: Text2video-zero: Text-to-image diffusion models are zero-shot video generators. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15954–15964 (2023)
18. 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)
19. Li, C., Huang, D., Lu, Z., Xiao, Y., Pei, Q., Bai, L.: A survey on long video generation: Challenges, methods, and prospects. arXiv preprint arXiv:2403.16407 (2024)
20. Li*, M., Lin*, Y., Zhang*, Z., Cai, T., Li, X., Guo, J., Xie, E., Meng, C., Zhu, J.Y., Han, S.: Svdquant: Absorbing outliers by low-rank components for 4-bit diffusion models. In: The Thirteenth International Conference on Learning Representations (2025)
21. Li, X., Liu, Y., Lian, L., Yang, H., Dong, Z., Kang, D., Zhang, S., Keutzer, K.: Q-diffusion: Quantizing diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 17535–17545 (October 2023)
22. Lu, C., Zhou, Y., Bao, F., Chen, J., LI, C., Zhu, J.: Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems. vol. 35, pp. 5775–5787. Curran Associates, Inc. (2022)
23. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. Machine Intelligence Research **22**(4), 730–751 (Jun 2025). https://doi.org/10.1007/s11633-025-1562-4
24. Lu, Y., Liang, Y., Zhu, L., Yang, Y.: Freelong: Training-free long video generation with spectralblend temporal attention. Advances in Neural Information Processing Systems **37**, 131434–131455 (2024)
25. Lv, Z., Si, C., Song, J., Yang, Z., Qiao, Y., Liu, Z., Wong, K.Y.K.: Fastercache: Training-free video diffusion model acceleration with high quality. In: arxiv (2024)
26. Ma, X., Wang, Y., Jia, G., Chen, X., Liu, Z., Li, Y.F., Chen, C., Qiao, Y.: Latte: Latent diffusion transformer for video generation. arXiv preprint arXiv:2401.03048 (2024)
27. Ma, X., Fang, G., Wang, X.: Deepcache: Accelerating diffusion models for free (2023)
28. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4195–4205 (October 2023)
29. Ren, W., Yang, H., Zhang, G., Wei, C., Du, X., Huang, W., Chen, W.: Consisti2v: Enhancing visual consistency for image-to-video generation. arXiv preprint arXiv:2402.04324 (2024)
30. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. In: International Conference on Learning Representations (2022)
31. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: International Conference on Learning Representations (2021)

32. Song, Y., Dhariwal, P., Chen, M., Sutskever, I.: Consistency models. arXiv preprint arXiv:2303.01469 (2023)
33. Tewel, Y., Kaduri, O., Gal, R., Kasten, Y., Wolf, L., Chechik, G., Atzmon, Y.: Training-free consistent text-to-image generation. ACM Transactions on Graphics (TOG) **43**(4), 1–18 (2024)
34. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., Wang, J., Zhang, J., Zhou, J., Wang, J., Chen, J., Zhu, K., Zhao, K., Yan, K., Huang, L., Feng, M., Zhang, N., Li, P., Wu, P., Chu, R., Feng, R., Zhang, S., Sun, S., Fang, T., Wang, T., Gui, T., Weng, T., Shen, T., Lin, W., Wang, W., Wang, W., Zhou, W., Wang, W., Shen, W., Yu, W., Shi, X., Huang, X., Xu, X., Kou, Y., Lv, Y., Li, Y., Liu, Y., Wang, Y., Zhang, Y., Huang, Y., Li, Y., Wu, Y., Liu, Y., Pan, Y., Zheng, Y., Hong, Y., Shi, Y., Feng, Y., Jiang, Z., Han, Z., Wu, Z.F., Liu, Z.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
35. Wijmans, J.G., Baker, R.W.: The solution-diffusion model: a review. Journal of membrane science **107**(1-2), 1–21 (1995)
36. Xi, H., Yang, S., Zhao, Y., Xu, C., Li, M., Li, X., Lin, Y., Cai, H., Zhang, J., Li, D., et al.: Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity. arXiv preprint arXiv:2502.01776 (2025)
37. Xia, Y., Ling, S., Fu, F., Wang, Y., Li, H., Xiao, X., Cui, B.: Training-free and adaptive sparse attention for efficient long video generation. arXiv preprint arXiv:2502.21079 (2025)
38. Xiao, G., Tian, Y., Chen, B., Han, S., Lewis, M.: Efficient streaming language models with attention sinks. arXiv preprint arXiv:2309.17453 (2023)
39. Xing, Z., Feng, Q., Chen, H., Dai, Q., Hu, H., Xu, H., Wu, Z., Jiang, Y.G.: A survey on video diffusion models. ACM Computing Surveys **57**(2), 1–42 (2024)
40. Yang, S., Xi, H., Zhao, Y., Li, M., Zhang, J., Cai, H., Lin, Y., Li, X., Xu, C., Peng, K., et al.: Sparse videogen2: Accelerate video generation with sparse attention via semantic-aware permutation. arXiv preprint arXiv:2505.18875 (2025)
41. 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)
42. Yu, J., Wang, Y., Zhao, C., Ghanem, B., Zhang, J.: Freedom: Training-free energy-guided conditional diffusion model. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 23174–23184 (2023)
43. Yuan, Z., Xie, R., Shang, Y., Zhang, H., Wang, S., Yan, S., Dai, G., Wang, Y.: Vgdfr: Diffusion-based video generation with dynamic latent frame rate (2025)
44. Zhang, E., Tang, J., Ning, X., Zhang, L.: Training-free and hardware-friendly acceleration for diffusion models via similarity-based token pruning. In: Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence. AAAI'25/IAAI'25/EAAI'25, AAAI Press (2025). https://doi.org/10.1609/aaai.v39i9.33071
45. Zhang, J., Huang, H., Zhang, P., Wei, J., Zhu, J., Chen, J.: Sageattention2: Efficient attention with thorough outlier smoothing and per-thread int4 quantization. In: International Conference on Machine Learning (ICML) (2025)
46. Zhang, Z., Sheng, Y., Zhou, T., Chen, T., Zheng, L., Cai, R., Song, Z., Tian, Y., Ré, C., Barrett, C., Wang, Z.A., Chen, B.: H2o: Heavy-hitter oracle for efficient

generative inference of large language models. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 34661–34710. Curran Associates, Inc. (2023)
47. Zhao, X., Jin, X., Wang, K., You, Y.: Real-time video generation with pyramid attention broadcast. arXiv preprint arXiv:2408.12588 (2024)
48. Zheng, K., Lu, C., Chen, J., Zhu, J.: Dpm-solver-v3: Improved diffusion ode solver with empirical model statistics. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 55502–55542. Curran Associates, Inc. (2023)
49. Zheng, S., Feng, L., Wang, X., Zhou, Q., Cai, P., Zou, C., Liu, J., Lin, Y., Chen, J., Ma, Y., et al.: Forecast then calibrate: Feature caching as ode for efficient diffusion transformers. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 40, pp. 13449–13457 (2026)
50. Zhou, X., Liang, D., Chen, K., Feng, T., Chen, X., Lin, H., Ding, Y., Tan, F., Zhao, H., Bai, X.: Less is enough: Training-free video diffusion acceleration via runtime-adaptive caching. arXiv preprint arXiv:2507.02860 (2025)