FIS-DiT: Breaking the Few-Step Video Inference Barrier via Training-Free Frame Interleaved Sparsity

Anonymous authors
Paper under double-blind review

Abstract

While the overall inference latency of Video Diffusion Transformers (DiTs) can be substantially reduced through model distillation, per-step inference latency remains a critical bottleneck. Existing acceleration paradigms primarily exploit redundancy across the denoising trajectory; however, we identify a limitation where these step-wise strategies encounter diminishing returns in few-step regimes. In such scenarios, fewer intermediate states can limit opportunities for feature reuse or predictive modeling, motivating complementary forms of acceleration. To overcome this, we propose Frame Interleaved Sparsity DiT (FIS-DiT), a training-free and operator-agnostic framework that shifts the optimization focus from the temporal trajectory to the latent frame dimension. Our approach is motivated by an intrinsic duality within this dimension: the existence of frame-wise sparsity that permits reduced computation, coupled with a structural consistency that calls for maintaining coverage of frame positions in the global spatiotemporal context. Leveraging this insight, we implement Frame Interleaved Sparsity (FIS) as an execution strategy that manipulates frame subsets across the model hierarchy, refreshing all latent positions without requiring full-scale block computation. Empirical evaluations on Wan 2.2 and HunyuanVideo 1.5 demonstrate that FIS-DiT reaches 2.4$\times$ speedup while maintaining comparable performance on VBench, providing a scalable and robust pathway toward real-time high-definition video generation.

1 Introduction

Recent advances in diffusion transformers (DiTs) have substantially improved the fidelity, motion realism, and temporal coherence of video generation (Peebles & Xie, 2023; Ma et al., 2024a; Zheng et al., 2024; Yang et al., 2025; Kong et al., 2024; Wan Team, 2025). Meanwhile, model distillation has greatly reduced the number of denoising steps required for high-quality synthesis, making few-step video generation increasingly practical (Song et al., 2023; Luo et al., 2023; Wang et al., 2023a; 2024; Ding et al., 2025). However, fewer sampling steps do not eliminate the inference bottleneck of DiTs. For high-resolution video synthesis, each denoising step still requires expensive spatio-temporal Transformer computation over a large number of latent tokens. As the denoising trajectory is compressed to only a few steps, the cost of each remaining DiT evaluation becomes an increasingly important part of overall inference latency.

A prominent family of training-free acceleration methods exploits redundancy along the denoising trajectory. They reuse, predict, cache, or skip intermediate states across timesteps, with policies ranging from step-level reuse to block-level feature caching (Ma et al., 2024b; Zhao et al., 2025; Lv et al., 2025; Liu et al., 2025a; Chen et al., 2024b; Ma et al., 2025b; Chung et al., 2026; Liu et al., 2025b). Despite their different granularities, these methods often rely on nearby denoising states being sufficiently similar to support approximation of current-step computation.

Opportunities to exploit this redundancy may narrow in few-step distilled DiTs. When the denoising process is compressed to only a handful of steps, fewer intermediate states are available for reuse, and adjacent steps may be more widely separated. As a result, cross-step matching, prediction, and caching may offer less opportunity for additional acceleration in this regime. Together, these constraints create a practical barrier to further trajectory-based acceleration in few-step settings.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To break this barrier, we shift the acceleration perspective from the denoising trajectory to the latent frame dimension. The latent frame dimension refers to the temporal axis of the video latent representation, where each latent frame contains a complete spatial token grid. Inspired by the strong correlation between neighboring video frames in video frame interpolation (Jiang et al., 2018; Bao et al., 2019; Huang et al., 2022; Reda et al., 2022), we ask whether similar redundancy also exists inside the intermediate latent representations of DiTs. Rather than generating a low-frame-rate video followed by pixel-space interpolation, we explore whether DiTs can sparsely compute only a subset of latent frames and restore the full latent sequence within the network hierarchy.

Our analysis reveals three key properties of latent-frame dynamics in few-step DiTs. First, adjacent latent frames within the same Transformer block exhibit structured local temporal redundancy, suggesting that omitted frame states can be approximated from neighboring ones in suitable blocks. Second, this redundancy is block-dependent: many middle blocks are relatively stable, while some early and late blocks show larger frame-wise variations and call for protection from sparse computation. Third, lightweight feature-level interpolation can approximate omitted latent states at low computational cost, although its error depends on the block and frame position.

Based on these observations, we propose **Frame Interleaved Sparsity DiT (FIS-DiT)**, a training-free acceleration framework for few-step DiTs. Unlike cache-based methods that reuse historical states across timesteps, FIS-DiT reduces computation within each forward pass by sparsely evaluating latent frames and reconstructing the full sequence via lightweight interpolation.

Our main contributions are summarized as follows:

- **New Acceleration Perspective:** We observe that trajectory-based reuse may offer diminishing returns in few-step regimes and shift the focus from cross-step redundancy to within-step latent-frame sparsity.
- **FIS-DiT:** We propose a training-free framework that sparsely computes complete latent-frame slices in stable blocks while preserving full computation in sensitive blocks, interleaves the computed frame subsets across layers to periodically refresh all frame positions, avoids additional feature-cache memory, and can be plugged into existing DiT inference pipelines without retraining.
- **Empirical Performance:** On the challenging 4-step distilled Wan 2.2 setting at 720p, FIS-DiT achieves a 2.11×–2.41× inference speedup while maintaining comparable performance on VBench.

2 Related Work

2.1 Diffusion Models for Video Synthesis

Diffusion models have become foundational in generative modeling due to their ability to produce high-quality and diverse samples (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021; Nichol & Dhariwal, 2021; Dhariwal & Nichol, 2021). Latent diffusion models improve scalability by performing denoising in a compact latent space (Rombach et al., 2022), while early video diffusion models extend image diffusion backbones with temporal modules for temporally coherent generation (Ho et al., 2022b; Singer et al., 2022; Ho et al., 2022a; Blattmann et al., 2023; Zhou et al., 2022; Guo et al., 2024).

Recently, diffusion transformers (DiTs) (Peebles & Xie, 2023) have been increasingly adopted for video generation due to their scalability and modeling capacity. Modern video diffusion systems operate over dense spatio-temporal latent tokens and achieve strong fidelity, motion realism, and temporal coherence (Wang et al., 2023b; Chen et al., 2023a; Xing et al., 2024; Ma et al., 2024a; Zheng et al., 2024; Yang et al., 2025; Kong et al., 2024; Wan Team, 2025; Polyak et al., 2024). However, this improved generation quality comes with substantial inference cost, as each denoising evaluation requires expensive Transformer computation over a large number of video latent tokens.

2.2 Efficiency Improvements in Diffusion Models

Existing efficiency improvements for diffusion models can be broadly categorized into reducing the number of sampling steps and reducing the computational cost per step. For sampling step reduc-

tion, fast SDE/ODE solvers and improved samplers accelerate diffusion inference without changing model weights (Karras et al., 2022; Lu et al., 2022a;b), while distillation and consistency-based methods compress the denoising trajectory into only a few steps (Salimans & Ho, 2022; Song et al., 2023; Luo et al., 2023). These ideas have also been extended to video generation (Wang et al., 2023a; 2024; Ding et al., 2025). However, few-step generation mainly shortens the denoising trajectory and does not directly reduce the cost of each remaining DiT forward pass.

Training-free acceleration methods reduce inference cost by reusing, caching, broadcasting, or predicting computation across denoising timesteps (Ma et al., 2024b; Li et al., 2024; Zhao et al., 2025; Lv et al., 2025; Liu et al., 2025a; Chen et al., 2024b; Kahatapitiya et al., 2025; Ma et al., 2025a; Wang et al., 2026; Selvaraju et al., 2024; Ma et al., 2025b). Representative designs include feature reuse or broadcasting (Ma et al., 2024b; Zhao et al., 2025), DiT-oriented caching (Chen et al., 2024b; Lv et al., 2025; Liu et al., 2025a; Kahatapitiya et al., 2025; Ma et al., 2025a; Selvaraju et al., 2024; Wang et al., 2026; Ma et al., 2025b; Chung et al., 2026), and feature forecasting (Liu et al., 2025b; Feng et al., 2025). Their reuse opportunities shrink when distillation compresses the trajectory to only a few steps. DisCa targets this regime with a learned feature-cache predictor for step-distilled video DiTs (Zou et al., 2026); FIS-DiT instead requires no additional training and reduces computation within each remaining step. Within-step acceleration has also been studied through token pruning, pooling, learning, and merging (Rao et al., 2021; Liang et al., 2022; Ryoo et al., 2021; Yin et al., 2022; Marin et al., 2023; Bolya et al., 2023; Chen et al., 2023b), token reduction for diffusion or multimodal inference (Bolya & Hoffman, 2023; Chen et al., 2024a), and sparse or linear attention (Zhang et al., 2026). Fine-grained token reduction can introduce irregular indexing, alter spatial token structure, or complicate compatibility with efficient attention kernels such as FlashAttention (Dao et al., 2022). FIS-DiT is also related to video frame interpolation (Jiang et al., 2018; Bao et al., 2019; Huang et al., 2022; Reda et al., 2022), but reconstructs intermediate latent states inside the DiT hierarchy rather than interpolating output frames as post-processing.

Differences with Previous Methods. Prior cache-based methods exploit redundancy across denoising timesteps, an assumption that becomes fragile in few-step regimes where adjacent steps contain large semantic changes. FIS-DiT shifts the acceleration axis from cross-timestep reuse to within-step latent-frame sparsity, preserving the spatial token grid and the full-frame module interface. It is also complementary to timestep-oriented caching: caching reduces redundant steps along the trajectory, while FIS-DiT reduces frame-wise computation inside each remaining step.

3 METHOD

3.1 PRELIMINARIES: VIDEO DIFFUSION TRANSFORMERS

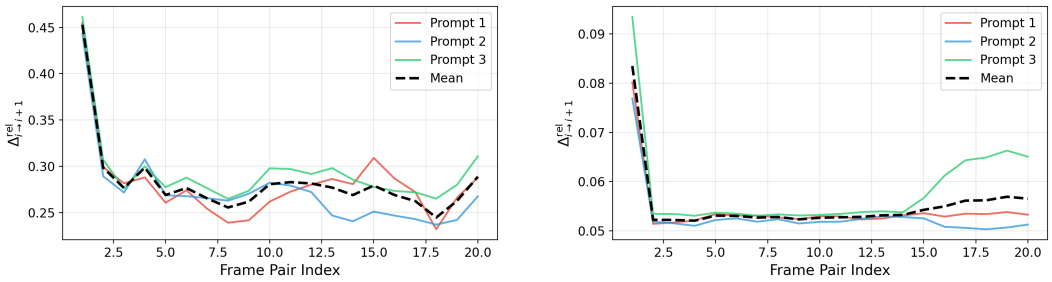
Latent Video Generation. Modern video diffusion models operate in a compressed latent space (Rombach et al., 2022). A pre-trained 3D VAE encodes a video into $z_0 \in \mathbb{R}^{F\times C\times H\times W}$, and the generative process denoises $z_T \sim \mathcal{N}(0, \mathbf{I})$ stepwise until z_0 is recovered.

DiT Architecture. The denoising network $\epsilon_\theta(z_t, t)$ is parameterized by a Transformer (DiT) with L stacked blocks $\{\Phi_l\}_{l=0}^{L-1}$. Each block processes patchified tokens via sub-modules Φ_l^m (e.g., Spatial-Temporal Attention and FFNs). The global attention complexity scales quadratically with the sequence length $\mathcal{O}((F \cdot H \cdot W)^2)$, making full-frame evaluation prohibitively expensive for large F.

3.2 EMPIRICAL OBSERVATIONS: TEMPORAL REDUNDANCY IN DITS

In this section, we empirically characterize the temporal redundancy of intermediate representations in video DiTs. In particular, we show that: **(1)** after an initial change, adjacent-frame relative-change profiles are comparatively even across time and show similar trends across three prompt types, **(2)** temporal redundancy is highly heterogeneous across network depth, and **(3)** interpolated-frame relative-error profiles show no apparent temporal drift on either backbone despite different mean levels.

Let $X_i^{(b,s,p)} \in \mathbb{R}^{H\times W\times D}$ denote the feature tensor of the i-th frame at transformer block b, denoising step s, and prompt p, where H and W are the spatial dimensions and D is the channel dimension.



(a) Wan 2.2 (40 blocks, 4 steps) (b) HunyuanVideo 1.5 (54 blocks, 8 steps)

Figure 1: Adjacent-frame relative changes $\Delta^{\text{rel}}_{i\to i+1}$ drop after the first frame pair and then vary more mildly across time, with similar trends across three prompt types on both models.

For notational simplicity, we omit the superscript (b, s, p) in the following definitions and assume that all quantities are computed at a fixed block, a fixed denoising step, and a fixed prompt.

To quantify frame-to-frame change, we first reduce the channel dimension by computing the L_2 norm of each spatial token:

$$V_i(h, w) = \|X_i(h, w)\|_2, \tag{1}$$

where $X_i(h, w) \in \mathbb{R}^D$ denotes the feature vector at spatial location (h, w), and $V_i \in \mathbb{R}^{H\times W}$ is the resulting spatial magnitude map of frame i.

We then define the *absolute adjacent-frame distance* as the L_2 distance between two consecutive spatial magnitude maps:

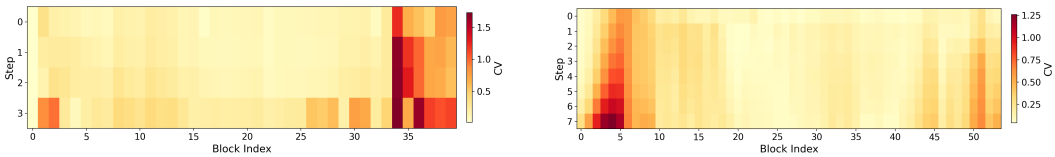
$$\Delta^{\text{abs}}_{i\to i+1} = \|V_{i+1} - V_i\|_2. \tag{2}$$

To obtain a scale-invariant measure, we normalize by the magnitude of the preceding frame, yielding the *relative change*:

$$\Delta^{\text{rel}}_{i\to i+1} = \frac{\Delta^{\text{abs}}_{i\to i+1}}{\|V_i\|_2 + \epsilon}, \qquad \epsilon = 10^{-8}. \tag{3}$$

This normalized quantity measures the fractional change of the spatial feature magnitude from frame i to frame $i + 1$, making the comparison less sensitive to feature scale differences across blocks and denoising steps.

Adjacent-Frame Dynamics Across Prompts. We evaluate Eq. 3 on two architecturally distinct video DiTs: Wan 2.2 (40 blocks, 4 denoising steps) and HunyuanVideo 1.5 (54 blocks, 8 denoising steps). As shown in Fig. 1, the temporal curves $\{\Delta^{\text{rel}}_{i\to i+1}\}_{i=0}^{F-2}$ fall after the first frame pair and are comparatively even over the remaining pairs, so no later transition consistently dominates feature evolution. The three prompt types span fast aquatic motion, non-rigid animal motion, and slower stylized motion; their exact text is given in Appendix C. Their curves follow similar overall trends on both models, although their levels and local fluctuations differ. These measurements describe intermediate latent-feature magnitude maps after 3D VAE compression and temporal mixing, rather than pixel-level motion. This regularity supports a static evenly spaced sparse policy in the evaluated settings.



(a) Wan 2.2 (40 blocks, 4 steps) (b) HunyuanVideo 1.5 (54 blocks, 8 steps)

Figure 2: Block-aware temporal heterogeneity. CV heatmaps show temporally stable middle blocks and larger fluctuations in head/tail blocks on (a) Wan 2.2 and (b) HunyuanVideo 1.5.

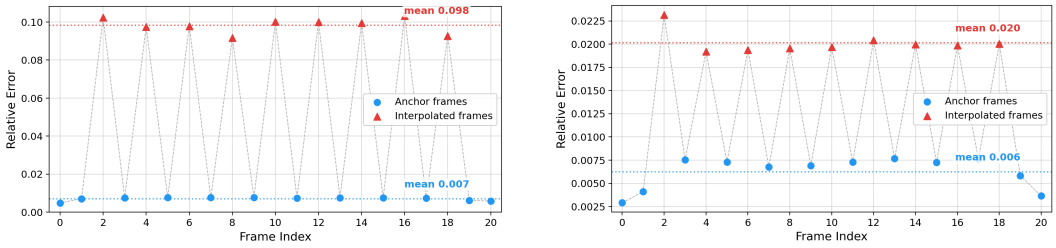
Block-Aware Heterogeneity. Although adjacent-frame changes are broadly uniform over time, their stability varies substantially across network depth. To quantify this effect, for each block b and

denoising step s, we summarize the flatness of the temporal curve $\{\Delta^{\text{rel}}_{i\to i+1}(b,s)\}_{i=0}^{F-2}$ using the coefficient of variation:

$$\mu_{b,s} = \frac{1}{F-1}\sum_{i=0}^{F-2}\Delta^{\text{rel}}_{i\to i+1}(b,s), \qquad \sigma_{b,s} = \left(\frac{1}{F-1}\sum_{i=0}^{F-2}\left(\Delta^{\text{rel}}_{i\to i+1}(b,s)-\mu_{b,s}\right)^2\right)^{1/2}, \tag{4}$$

$$\text{CV}_{b,s} = \frac{\sigma_{b,s}}{\mu_{b,s}+\epsilon}, \qquad \overline{\text{CV}}_{b,s} = \frac{1}{P}\sum_{p=1}^{P}\text{CV}^{(p)}_{b,s}, \qquad P = \text{number of prompts}. \tag{5}$$

As shown in Fig. 2, the resulting heatmaps on both models reveal a consistent block-dependent pattern: middle blocks exhibit low CV values, indicating highly uniform adjacent-frame changes, while head and tail blocks display noticeably larger fluctuations. This heterogeneity directly motivates our block-aware design: sparse approximation is applied only to the stable middle block set $\mathcal{M}$, while sensitive head and tail blocks retain full-frame computation.



(a) Wan 2.2 (b) HunyuanVideo 1.5

Figure 3: Per-frame relative magnitude-map deviation. Anchor frames (blue) remain below 1%; interpolated frames (red) show no apparent temporal drift on either model despite different mean levels.

Interpolation Error Diagnostics. To characterize the effect of reconstructing skipped frames, we measure the per-frame relative error under sparse evaluation (n=2). Let Y_i and $\hat{Y}_i$ denote the full-computation and sparse-evaluation outputs for frame i, respectively. The per-frame relative error is:

$$\mathcal{E}_i = \frac{\|\hat{V}_i - V_i\|_2}{\|V_i\|_2 + \epsilon}, \tag{6}$$

where $V_i(h,w) = \|Y_i(h,w)\|_2$ and $\hat{V}_i(h,w) = \|\hat{Y}_i(h,w)\|_2$. Figure 3 shows deviations below 1% at anchors and mean deviations of 9.8% on Wan and 2.0% on HunyuanVideo at interpolated frames. These same-input, single-block probes measure intermediate magnitude maps, not decoded-video errors or additive layer-wise increments. Because the backbones and block–step locations differ, the numerical gap is descriptive, not a controlled attribution to step count or VAE design. Both profiles show no temporal drift; Appendix C reports the depth profile.

These findings show structured, block-dependent intra-step temporal redundancy with similar trends across the evaluated prompts and backbones, forming the basis of our inference strategy in Section 3.3.

3.3 The FIS-DiT Framework

FIS-DiT is a training-free sparse inference framework that replaces redundant full-frame computation with a sparse-compute-and-reconstruct paradigm. In safe regions, it computes only anchor frames, reconstructs skipped frames, interleaves anchor sets across layers, and disables sparsity in high-risk blocks or timesteps.

3.3.1 Interpolation-based Frame Reconstruction

Let X_l be the input feature sequence of module Φ_l^m in block l, and let $\mathcal{F} = \{0,\dots,F-1\}$. Given anchor set $\mathcal{A}_l \subseteq \mathcal{F}$, FIS-DiT performs:

$$X_l^{\text{anc}} = \texttt{Gather}(X_l, \mathcal{A}_l), \tag{7}$$

$$Y_l^{\text{anc}} = \Phi_l^m(X_l^{\text{anc}}), \tag{8}$$

$$X_{l+1} = \mathcal{R}(Y_l^{\text{anc}}, \mathcal{A}_l, \mathcal{F}), \tag{9}$$

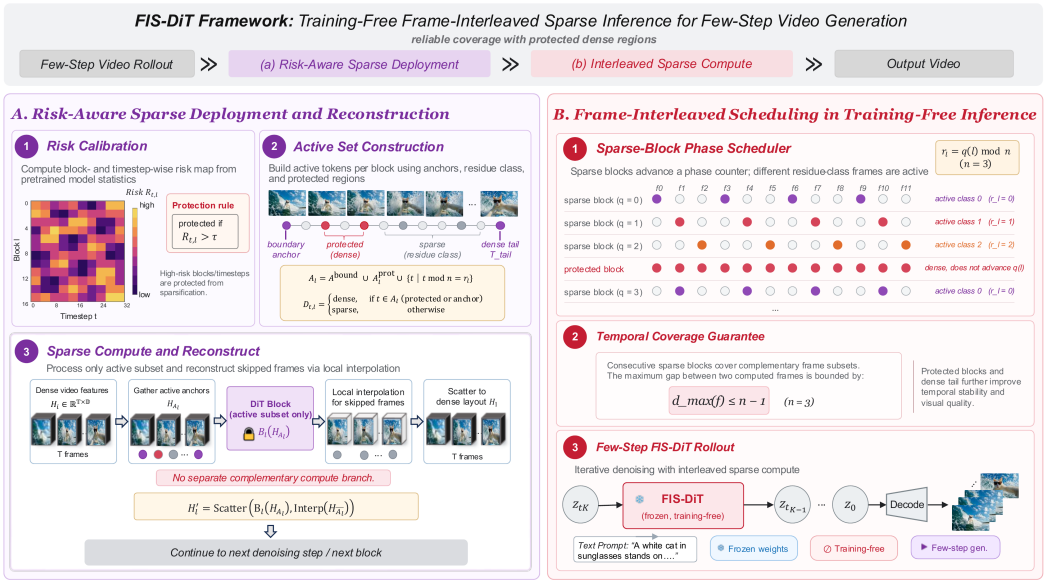


Figure 4: Overview of FIS-DiT.

where $\mathcal{R}$ reconstructs the full frame sequence from sparse anchor outputs.

The operator $\mathcal{R}$ is generic; since few-step video generation requires a training-free, low-overhead, and stable operator, we use local linear interpolation by default. For a skipped frame f_u between neighboring anchors f_a and f_b, we compute:

$$X_{l+1}(f_u) = \frac{f_b - f_u}{f_b - f_a} Y_l^{\text{anc}}(f_a) + \frac{f_u - f_a}{f_b - f_a} Y_l^{\text{anc}}(f_b). \tag{10}$$

Anchor frames are directly copied from Y_l^{anc}. We always include the first and last frames as anchors, since boundary frames lack two-sided context and empirically show larger variations.

Hardware efficiency. Frame-level sparsity preserves contiguous spatial tokens within each selected frame, avoiding the fragmented layouts caused by unstructured token pruning. FIS-DiT can therefore reuse optimized dense kernels such as FlashAttention (Dao et al., 2022) without custom CUDA kernels.

3.3.2 Interleaved Anchor Scheduling

A fixed anchor set repeatedly excludes the same frames from nonlinear modules and can concentrate approximation error across layers. FIS-DiT instead interleaves anchor sets across sparse blocks.

Let κ be the target anchor ratio and $n = \lceil 1/\kappa \rceil$ be the anchor stride. For sparse block l, let $q(l) = |\{j \in \mathcal{M} : j < l\}|$ count preceding sparse blocks, and define:

$$r_l = q(l) \bmod n, \tag{11}$$

The anchor set is:

$$\mathcal{A}_l = \{f \in \mathcal{F} \mid (f - r_l) \bmod n = 0\} \cup \{0, F-1\}. \tag{12}$$

Across any n consecutive sparse blocks, every non-boundary frame becomes an anchor exactly once, so no frame is reconstructed in every block of the cycle. Direct evaluation still receives features carrying earlier approximation error; it does not reset the frame to the dense trajectory.

3.3.3 Block- and Step-Aware Sparse Deployment

Since approximation risk is heterogeneous, FIS-DiT applies sparse computation only when both the block and timestep are safe. We partition DiT blocks into stable middle blocks $\mathcal{M}$ and sensitive

head/tail blocks $\mathcal{S}$:

$$\mathcal{M} \cup \mathcal{S} = \{0, \ldots, L-1\}, \qquad \mathcal{M} \cap \mathcal{S} = \emptyset. \tag{13}$$

Sparse reconstruction is enabled only for $l \in \mathcal{M}$, while $l \in \mathcal{S}$ retains full-frame computation. The protected set is determined automatically by a one-time calibration pass: we compute the CV profile in Eq. 5 and apply a fixed threshold, without manual heatmap inspection or per-prompt adjustment at inference time. The resulting sets can differ across backbones under the same selection rule.

For timestep-level risk control, FIS-DiT disables sparsity in the final denoising steps, where low-noise refinement is sensitive to approximation errors. With zero-based step index $t \in \{0, \ldots, T-1\}$ and full-frame tail length T_{tail}, the global gate is:

$$G(l,t) = \mathbf{1}[l \in \mathcal{M}] \cdot \mathbf{1}[t < T - T_{\text{tail}}]. \tag{14}$$

The module computation becomes:

$$X_{l+1} = \begin{cases} \mathcal{R}\left(\Phi_l^m(\texttt{Gather}(X_l, \mathcal{A}_l)), \mathcal{A}_l, \mathcal{F}\right), & G(l,t) = 1, \\ \Phi_l^m(X_l), & G(l,t) = 0. \end{cases} \tag{15}$$

This gate allows FIS-DiT to accelerate stable regions while preserving full-frame computation in sensitive blocks and final denoising steps.

4 Experiments

4.1 Experimental Settings

Models and Evaluation. We evaluate FIS-DiT on two publicly available few-step video diffusion backbones: Wan 2.2 (Wan Team, 2025) (14B, 4-step T2V at 480p/720p) and HunyuanVideo 1.5 (Kong et al., 2024) (8B, 8-step I2V at 480p). These models cover different scales, generation paradigms, and few-step regimes. The main comparison spans step caching (MagCache (Ma et al., 2025b) and SeaCache (Chung et al., 2026)), step prediction (TaylorSeer (Liu et al., 2025b)), token merging (ToMe (Bolya & Hoffman, 2023)), and sparse attention (SLA (Zhang et al., 2026)). We use the recent MagCache as a representative caching baseline and report its forced configuration to obtain nontrivial acceleration in the few-step setting. We likewise report relaxed SeaCache and TaylorSeer configurations. We report FLOPs, inference latency, and speedup for efficiency; for quality, we use VBench (Huang et al., 2024) Quality Score, Semantic Score, and Total Score for Wan T2V, and VBench-I2V (Huang et al., 2024; 2025) Quality Score, I2V Score, and Total Score for HunyuanVideo I2V.

Implementation Details. All experiments are conducted on a single GPU with FlashAttention enabled: NVIDIA H20 for Wan 2.2 and NVIDIA L20 for HunyuanVideo 1.5. We evaluate 81-frame generation using 100 diverse prompts sampled from VBench (Huang et al., 2024); for HunyuanVideo, each prompt is paired with a reference image for I2V evaluation.

The anchor stride n controls the frame sparsity ratio, with $\kappa = 1/n$, and we evaluate $n \in \{2, 3, 4, 5\}$. We obtain the sensitive block set $\mathcal{S}$ from a one-time CV calibration with a fixed threshold and reuse it across prompts; the resulting head/tail protection follows the pattern in Fig. 2. The final denoising step is kept dense with $T_{\text{tail}} = 1$. Specifically, for Wan 2.2 (40 blocks, $T = 4$), we set $\mathcal{S} = \{0, 38, 39\}$ and apply sparsity to steps 0–2. For HunyuanVideo 1.5 (54 blocks, $T = 8$), we set $\mathcal{S} = \{0\text{–}5, 51\text{–}53\}$ and apply sparsity to steps 0–6. We denote the conservative setting $n = 3$ as **FIS-Balanced** and the aggressive setting $n = 5$ as **FIS-Fast**. We also evaluate **FIS-Balanced + MagCache-Force** to test complementarity between intra-step frame sparsity and inter-step caching.

4.2 Main Results and Visualization

Efficiency and Quality. Table 1 shows that FIS-DiT substantially accelerates all three few-step settings with little change in aggregate VBench scores, reporting task-specific VBench Semantic and Total Scores for Wan T2V, and I2V and Total Scores for HunyuanVideo I2V. FIS-Fast reaches 2.09×–2.41× speedup, with a maximum Total Score decrease of 0.0036 on Wan and no decrease on HunyuanVideo; FIS-Balanced reaches 1.86×–2.11×, reducing the maximum Wan decrease to 0.0014 and again showing no decrease on HunyuanVideo. Combining FIS-Balanced with

Table 1: Efficiency and quality comparison of FIS-DiT and competing methods on few-step Wan 2.2 T2V and HunyuanVideo 1.5 I2V.

Method	Axis	Efficiency			VBench Evaluation		
		FLOPs (P) ↓	Latency (s) ↓	Speedup ↑	Quality Score ↑	Semantic / I2V Score ↑	Total Score ↑
(a) Wan 2.2-T2V-A14B + Lightning LoRA (81 frames, 720p, $T = 4$; Quality + Semantic)							
Baseline	–	26.09	269.1	1.00×	0.8846	0.7678	0.8612
MagCache-Fast (Ma et al., 2025b)	Step	26.09 [↓ 0.0%]	269.1 [↓ 0.0%]	1.00×	0.8846 (+0.0000)	0.7678 (+0.0000)	0.8612 (+0.0000)
MagCache-Force	Step	19.57 [↓ 25.0%]	203.6 [↓ 24.3%]	**1.32**×	0.8803 (−0.0043)	0.7646 (−0.0032)	0.8572 (−0.0040)
SeaCache-Relaxed† (Chung et al., 2026)	Step	19.33‡ [↓ 25.9%]	199.3 [↓ 25.9%]	**1.35**×	0.8673 (−0.0173)	0.7468 (−0.0210)	0.8432 (−0.0180)
TaylorSeer-Relaxed† (Liu et al., 2025b)	Step	12.92‡ [↓ 50.5%]	133.2 [↓ 50.5%]	**2.02**×	0.8692 (−0.0154)	0.7480 (−0.0198)	0.8450 (−0.0162)
ToMe ($r = 0.3$) (Bolya & Hoffman, 2023)	Token	15.17‡ [↓ 41.9%]	156.5 [↓ 41.9%]	**1.72**×	0.8525 (−0.0321)	0.6888 (−0.0790)	0.8198 (−0.0414)
SLA (Zhang et al., 2026)	Attn.	11.91‡ [↓ 54.3%]	122.9 [↓ 54.3%]	**2.19**×	0.8660 (−0.0186)	0.7115 (−0.0563)	0.8351 (−0.0261)
FIS-Balanced	Frame	11.55 [↓ 55.7%]	127.6 [↓ 52.6%]	**2.11**×	**0.8845** (−0.0001)	0.7610 (−0.0068)	0.8598 (−0.0014)
FIS-Fast	Frame	10.39 [↓ 60.2%]	111.7 [↓ 58.5%]	**2.41**×	0.8837 (−0.0009)	0.7546 (−0.0132)	0.8579 (−0.0033)
FIS-Bal. + Mag.-Force	Hybrid	**9.87** [↓ 62.2%]	**108.7** [↓ 59.6%]	**2.48**×	0.8844 (−0.0002)	0.7663 (−0.0015)	0.8608 (−0.0004)
(b) Wan 2.2-T2V-A14B + Lightning LoRA (81 frames, 480p, $T = 4$; Quality + Semantic)							
Baseline	–	6.71	66.8	1.00×	0.8778	0.7340	0.8490
MagCache-Fast (Ma et al., 2025b)	Step	6.71 [↓ 0.0%]	66.8 [↓ 0.0%]	1.00×	0.8778 (+0.0000)	0.7340 (+0.0000)	0.8490 (+0.0000)
MagCache-Force	Step	5.03 [↓ 25.0%]	51.1 [↓ 23.5%]	**1.31**×	0.8732 (−0.0046)	0.7340 (+0.0000)	0.8454 (−0.0036)
SeaCache-Relaxed† (Chung et al., 2026)	Step	4.76‡ [↓ 29.1%]	47.4 [↓ 29.1%]	**1.41**×	0.8640 (−0.0138)	0.7212 (−0.0128)	0.8354 (−0.0136)
TaylorSeer-Relaxed† (Liu et al., 2025b)	Step	3.21‡ [↓ 52.2%]	32.0 [↓ 52.2%]	**2.09**×	0.8596 (−0.0182)	0.6935 (−0.0405)	0.8264 (−0.0226)
ToMe ($r = 0.3$) (Bolya & Hoffman, 2023)	Token	4.25‡ [↓ 36.7%]	42.3 [↓ 36.7%]	**1.58**×	0.8370 (−0.0408)	0.6672 (−0.0668)	0.8030 (−0.0460)
SLA (Zhang et al., 2026)	Attn.	5.05‡ [↓ 24.8%]	50.2 [↓ 24.8%]	**1.33**×	0.8361 (−0.0417)	0.7115 (−0.0225)	0.8112 (−0.0378)
FIS-Balanced	Frame	3.17 [↓ 52.8%]	36.0 [↓ 46.1%]	**1.86**×	**0.8780** (+0.0002)	0.7299 (−0.0041)	0.8484 (−0.0006)
FIS-Fast	Frame	2.83 [↓ 57.8%]	32.0 [↓ 52.1%]	**2.09**×	0.8753 (−0.0025)	0.7257 (−0.0083)	0.8454 (−0.0036)
FIS-Bal. + Mag.-Force	Hybrid	**2.67** [↓ 60.2%]	**30.7** [↓ 54.0%]	**2.18**×	0.8766 (−0.0012)	0.7371 (+0.0031)	0.8487 (−0.0003)
(c) HunyuanVideo 1.5-8B + Step Distilled (I2V, 81 frames, 480p, $T = 8$; VBench-I2V Quality Score + I2V Score)							
Baseline	–	5.47	51.1	1.00×	0.7955	0.9184	0.8570
MagCache-Fast (Ma et al., 2025b)	Step	5.47 [↓ 0.0%]	51.1 [↓ 0.0%]	1.00×	0.7955 (+0.0000)	0.9184 (+0.0000)	0.8570 (+0.0000)
MagCache-Force	Step	3.42 [↓ 37.5%]	32.6 [↓ 36.2%]	**1.57**×	0.7914 (−0.0041)	0.9012 (−0.0172)	0.8463 (−0.0107)
SeaCache-Relaxed† (Chung et al., 2026)	Step	2.72‡ [↓ 50.2%]	25.4 [↓ 50.2%]	**2.01**×	0.7925 (−0.0030)	**0.9200** (+0.0016)	0.8563 (−0.0007)
TaylorSeer-Relaxed† (Liu et al., 2025b)	Step	3.30‡ [↓ 39.8%]	30.8 [↓ 39.8%]	**1.66**×	0.7928 (−0.0027)	0.9137 (−0.0047)	0.8533 (−0.0037)
ToMe ($r = 0.3$) (Bolya & Hoffman, 2023)	Token	2.29‡ [↓ 58.2%]	21.4 [↓ 58.2%]	**2.39**×	0.6929 (−0.1026)	0.8219 (−0.0965)	0.7574 (−0.0996)
SLA (Zhang et al., 2026)	Attn.	3.72‡ [↓ 32.0%]	34.8 [↓ 32.0%]	**1.47**×	0.8039 (+0.0084)	0.9176 (−0.0008)	0.8608 (+0.0038)
FIS-Balanced	Frame	2.33 [↓ 57.4%]	24.9 [↓ 51.3%]	**2.05**×	**0.8058** (+0.0103)	0.9188 (+0.0004)	**0.8623** (+0.0054)
FIS-Fast	Frame	**2.06** [↓ 62.3%]	**21.3** [↓ 58.3%]	**2.40**×	0.8041 (+0.0086)	0.9179 (−0.0005)	0.8610 (+0.0041)
FIS-Bal. + Mag.-Force	Hybrid	2.14 [↓ 60.9%]	21.8 [↓ 57.3%]	**2.34**×	0.8049 (+0.0094)	0.9183 (−0.0001)	0.8616 (+0.0047)

Notes. Wan reports VBench Quality, Semantic, and Total Scores ($\text{Total} = 0.8\,\text{Quality} + 0.2\,\text{Semantic}$); HunyuanVideo reports VBench-I2V Quality, I2V, and Total Scores ($\text{Total} = 0.5\,\text{Quality} + 0.5\,\text{I2V}$). Relative to each matched dense baseline, blue brackets denote efficiency reductions, while green/red/gray parentheses denote score improvements, degradations, or negligible changes ($|\Delta| \leq 0.001$). Score changes are computed before rounding. † denotes a relaxed baseline configuration; ‡ denotes idealized FLOPs estimated as $C_{\text{base}}/\text{Speedup}$ when traces are unavailable.

MagCache-Force reaches 2.18×–2.48× while keeping Total Scores close to the dense baselines, confirming complementarity with timestep caching.



Figure 5: Qualitative comparison on 4-step Wan 2.2 at 720p across three prompts.

Visual Comparison. Figure 5 compares three representative prompts—misty wilderness, surfing, and blooming flower—spanning atmospheric scenes, large motion, and fine-grained details. MagCache-Force introduces reduced scene layering, structural distortion, and texture/color degradation, whereas FIS-Balanced remains close to the baseline and FIS-Fast preserves clear structures and details at over 2× speedup. FIS-Balanced + MagCache-Force also remains visually close at 2.48×, consistent with the complementarity in Table 1.

4.3 Ablation Studies

We ablate interleaved frame scheduling and risk-aware sparse deployment on Wan 2.2-T2V-A14B + Lightning LoRA (4-step, 480p).

Effectiveness of Interleaved Frame Scheduling. Table 2 compares interleaved scheduling with a non-interleaved variant that uses the same computed frame subset across all sparse blocks. At matched strides and nearly identical speedups, this isolates the effect of interleaving. It raises CLIP-Score from 27.42 to 28.53 and VBench Quality Score from 0.8621 to 0.8780 at $n = 3$; at $n = 5$, it retains 28.50 and 0.8753, versus 26.08 and 0.8497 without interleaving, supporting its role in preventing approximation error from repeatedly concentrating on the same frames.

Table 2: Ablation on interleaved frame scheduling and stride n.

Frame Scheduling	Stride n	Latency (s) ↓	Speedup ↑	CLIP-Score ↑	VBench Quality ↑
Baseline	–	66.8	1.00×	**28.56**	0.8778
w/o Interleaving	3	35.7	1.87×	27.42	0.8621
w/ Interleaving	3	36.0	1.86×	28.53	**0.8780**
w/o Interleaving	4	33.4	2.00×	26.61	0.8558
w/ Interleaving	4	33.8	1.98×	28.51	0.8777
w/o Interleaving	5	31.6	2.11×	26.08	0.8497
w/ Interleaving	5	32.0	2.09×	28.50	0.8753

Impact of Risk-Aware Sparse Deployment. Table 3 evaluates the two protection terms. Removing block-aware protection raises speedup from 1.86× to 2.04× and lowers VBench Quality Score from 0.8780 to 0.8731, consistent with the larger head/tail fluctuations in Fig. 2; removing step-aware protection reaches 2.36× and 0.8586, confirming the sensitivity of final denoising steps. Removing both reaches 2.72× and 24.6s but lowers CLIP-Score to 27.35 and VBench Quality Score to 0.8427, showing that both protections are needed to balance efficiency and visual fidelity.

Table 3: Ablation on risk-aware sparse deployment ($n = 3$).

Configuration	Speedup ↑	Latency (s) ↓	CLIP-Score ↑	VBench Quality ↑
Baseline	1.00×	66.8	**28.56**	0.8778
w/o Block-aware Protection	2.04×	32.8	28.35	0.8731
w/o Step-aware Protection	2.36×	28.3	27.94	0.8586
w/o Risk-aware Deployment	**2.72**×	**24.6**	27.35	0.8427
FIS-Balanced	1.86×	36.0	28.53	**0.8780**

5 Conclusion and Future Work

We present FIS-DiT, a training-free framework that explores latent-frame sparsity as a complementary direction for accelerating few-step video DiTs. FIS-DiT achieves over 2× inference speedup in few-step video generation while maintaining comparable VBench performance. As a sparse inference method, it naturally reduces computation and memory use without retraining, and its orthogonality to timestep-oriented acceleration enables further speedup with caching, as verified in our experiments.

We demonstrate these results on two representative publicly available video DiT backbones, Wan 2.2 for T2V and HunyuanVideo 1.5 for I2V. Our evaluation relies on automatic metrics and does not include human-preference studies, while broader validation across video architectures remains future work. Together, these results establish latent-frame sparsity as an effective and complementary acceleration axis for few-step video generation.

Reproducibility statement

The method and evaluation settings are described in Sections 3 and 4. Appendix A gives the comparison protocol and deployment rule, and Appendix B states the assumptions and proof for the local analysis.

AI use statement

Large language models were used for language editing, formatting assistance, calculation checking, and drafting revision wording. All AI-assisted content was reviewed by the authors, who take responsibility for the final manuscript, including its technical claims, experimental results, and citations.

References

Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *CVPR*, 2019.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023.

Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *CVPR Workshop*, 2023.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *ICLR*, 2023.

Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023a.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *ECCV*, 2024a.

Mengzhao Chen, Wenqi Shao, Peng Xu, Mingbao Lin, Kaipeng Zhang, Fei Chao, Rongrong Ji, Yu Qiao, and Ping Luo. Diffrate: Differentiable compression rate for efficient vision transformers. In *ICCV*, 2023b.

Pengtao Chen, Mingzhu Shen, Peng Ye, Jianjian Cao, Chongjun Tu, Christos-Savvas Bouganis, Yiren Zhao, and Tao Chen. δ-dit: A training-free acceleration method tailored for diffusion transformers. *arXiv preprint arXiv:2406.01125*, 2024b.

Jiwoo Chung, Sangeek Hyun, MinKyu Lee, Byeongju Han, Geonho Cha, Dongyoon Wee, Youngjun Hong, and Jae-Pil Heo. Seacache: Spectral-evolution-aware cache for accelerating diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14283–14294, 2026.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *NeurIPS*, 2022.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021.

Zihan Ding, Chi Jin, Difan Liu, Haitian Zheng, Krishna Kumar Singh, Qiang Zhang, Yan Kang, Zhe Lin, and Yuchen Liu. Dollar: Few-step video generation via distillation and latent reward optimization. In *ICCV*, 2025.

Liang Feng, Shikang Zheng, Jiacheng Liu, Yuqi Lin, Qinming Zhou, Peiliang Cai, Xinyu Wang, Junjie Chen, Chang Zou, Yue Ma, and Linfeng Zhang. Hicache: Training-free acceleration of diffusion models via hermite polynomial-based feature caching, 2025.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *ICLR*, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022a.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *NeurIPS*, 2022b.

Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *ECCV*, 2022.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Vbench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. doi: 10.1109/TPAMI.2025.3633890.

Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *CVPR*, 2018.

Kumara Kahatapitiya, Haozhe Liu, Sen He, Ding Liu, Menglin Jia, Chenyang Zhang, Michael S. Ryoo, and Tian Xie. Adaptive caching for faster video generation with diffusion transformers. In *ICCV*, 2025.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022.

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

Senmao Li, Taihang Hu, Joost van de Weijer, Fahad Shahbaz Khan, Tao Liu, Linxuan Li, Shiqi Yang, Yaxing Wang, Ming-Ming Cheng, and Jian Yang. Faster diffusion: Rethinking the role of the encoder for diffusion model inference. In *NeurIPS*, 2024.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *ICLR*, 2022.

Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. In *CVPR*, 2025a.

Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Junjie Chen, and Linfeng Zhang. From reusing to forecasting: Accelerating diffusion models with taylorseers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15853–15863, 2025b.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *NeurIPS*, 2022a.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022b.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

Zhengyao Lv, Chenyang Si, Junhao Song, Zhenyu Yang, Yu Qiao, Ziwei Liu, and Kwan-Yee K. Wong. Fastercache: Training-free video diffusion model acceleration with high quality. In *ICLR*, 2025.

Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024a.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *CVPR*, 2024b.

Xuran Ma, Yexin Liu, Yaofu Liu, Xianfeng Wu, Mingzhe Zheng, Zihao Wang, Ser-Nam Lim, and Harry Yang. Model reveals what to cache: Profiling-based feature reuse for video diffusion models. In *ICCV*, 2025a.

Zehong Ma, Longhui Wei, Feng Wang, Shiliang Zhang, and Qi Tian. Magcache: Fast video generation with magnitude-aware cache. *arXiv preprint arXiv:2506.09045*, 2025b.

Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. Token pooling in vision transformers for image classification. In *WACV*, 2023.

Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, 2021.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

Adam Polyak et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *NeurIPS*, 2021.

Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. Film: Frame interpolation for large motion. In *ECCV*, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Michael S. Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: What can 8 learned tokens do for images and videos? In *NeurIPS*, 2021.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022.

Pratheba Selvaraju, Tianyu Ding, Tianyi Chen, Ilya Zharkov, and Luming Liang. Fora: Fast-forward caching in diffusion transformer acceleration. *arXiv preprint arXiv:2407.01425*, 2024.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, 2023.

Wan Team. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Fu-Yun Wang, Zhaoyang Huang, Xiaoyu Shi, Weikang Bian, Guanglu Song, Yu Liu, and Hongsheng Li. Animatelcm: Accelerating the animation of personalized diffusion models and adapters with decoupled consistency learning. *arXiv preprint arXiv:2402.00769*, 2024.

Jiangshan Wang, Kang Zhao, Jiayi Guo, Jiayu Wang, Hang Guo, Chenyang Zhu, Xiu Li, and Xiangyu Yue. Precisecache: Precise feature caching for efficient and high-fidelity video generation. *arXiv preprint arXiv:2603.00976*, 2026.

Xiang Wang, Shiwei Zhang, Han Zhang, Yu Liu, Yingya Zhang, Changxin Gao, and Nong Sang. Videolcm: Video latent consistency model. *arXiv preprint arXiv:2312.09109*, 2023a.

Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, Yuwei Guo, Tianxing Wu, Chenyang Si, Yuming Jiang, Cunjian Chen, Chen Change Loy, Bo Dai, Dahua Lin, Yu Qiao, and Ziwei Liu. Lavie: High-quality video generation with cascaded latent diffusion models. *arXiv preprint arXiv:2309.15103*, 2023b.

Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Dynamicrafter: Animating open-domain images with video diffusion priors. In *ECCV*, 2024.

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

Hongxu Yin, Arash Vahdat, Jose Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *CVPR*, 2022.

Jintao Zhang, Haoxu Wang, Kai Jiang, Shuo Yang, Kaiwen Zheng, Haocheng Xi, Ziteng Wang, Hongzhou Zhu, Min Zhao, Ion Stoica, Joseph E. Gonzalez, Jianfei Chen, and Jun Zhu. Sla: Beyond sparsity in diffusion transformers via fine-tunable sparse–linear attention. In *International Conference on Learning Representations*, 2026.

Xuanlei Zhao, Xiaolong Jin, Kai Wang, and Yang You. Real-time video generation with pyramid attention broadcast. In *ICLR*, 2025.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022.

Chang Zou, Changlin Li, Yang Li, Patrol Li, Jianbing Wu, Xiao He, Songtao Liu, Zhao Zhong, Kailin Huang, and Linfeng Zhang. Disca: Accelerating video diffusion transformers with distillation-compatible learnable feature caching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

A Implementation and Evaluation Details

A.1 Generation and Comparison Protocol

Table 4 collects the three evaluated settings. Each method in a matched comparison uses the same prompt and random seed; I2V comparisons additionally use the same reference image. The main evaluation uses 100 VBench prompts. Calibration prompts are separate from evaluation prompts, and the resulting protected-block mask is reused across evaluation prompts for its backbone.

Table 4: Evaluated backbones and generation settings.

Backbone	Task	Frames	Resolution	Steps	GPU
Wan 2.2-T2V-A14B + Lightning LoRA	T2V	81	480p	4	H20
Wan 2.2-T2V-A14B + Lightning LoRA	T2V	81	720p	4	H20
HunyuanVideo 1.5-8B + Step Distilled	I2V	81	480p	8	L20

Baseline operating points. At their default settings, SeaCache triggers essentially no reuse at four or eight denoising steps, while TaylorSeer gives no acceleration at four steps and only weak acceleration at eight steps because of its warmup and correction-order requirements. We therefore report relaxed operating points for both methods in Table 5; neither method's original paper targets few-step generation. The corresponding rows in Table 1 are marked with †.

Table 5: Operating-point summary for the evaluated methods.

Method	Acceleration axis	Configuration used in this paper
MagCache-Fast	Step caching	Default setting
MagCache-Force	Step caching	Force-cache the eligible denoising step that best satisfies MagCache's threshold criterion, ensuring at least one cached step
SeaCache-Relaxed	Step caching	Wan: `threshold=10, warmup=1, end_step=3, verbose=True, boundary=0.0`; HunyuanVideo: `threshold=1.0, warmup=1, end_step=7`
TaylorSeer-Relaxed	Feature forecasting	Wan: `order=0, warmup=1, compute_interval=3, boundary=0.9`; HunyuanVideo: `warmup=2` (default order 1, interval 2)
ToMe	Token merging	Merge ratio $r = 0.3$
SLA	Sparse attention	Matched few-step setting
FIS-Balanced	Frame sparsity	Stride $n = 3$, calibrated protected mask, dense final step
FIS-Fast	Frame sparsity	Stride $n = 5$, same protected mask and dense final step

Efficiency accounting. Table 1 reports measured end-to-end latency separately from FLOPs. On Wan 480p, FIS-Balanced reduces FLOPs by 52.8% and latency by 46.1%; the difference reflects work outside the sparsified modules and other execution overhead. One-time CV calibration is a setup cost and is not included in the per-video latency.

A.2 One-Time Calibration and Deployment

The calibration rule is applied once to calibration prompts; it does not update model weights or require prompt-specific statistics at inference time. Let $\overline{\mathrm{CV}}_{b,s}$ be the prompt-averaged CV from Eq. 5. To retain both block-wise and step-wise risk information while obtaining structured masks,

we project this two-dimensional profile onto the block and denoising-step axes,

$$r_b = \frac{1}{T}\sum_{s=0}^{T-1} \overline{\mathrm{CV}}_{b,s}, \qquad u_s = \frac{1}{L}\sum_{b=0}^{L-1} \overline{\mathrm{CV}}_{b,s}, \qquad \widehat{r}_b = \frac{r_b}{\max_j r_j}, \qquad \widehat{u}_s = \frac{u_s}{\max_v u_v}. \tag{16}$$

The normalization makes both summaries insensitive to the absolute CV scale of a backbone. To encode the observed head–middle–tail structure along network depth, let $\delta_b = \min\{b, L-1-b\}/(L-1)$ be the normalized distance to the nearest network boundary and define the boundary-aware block score

$$B_b = \frac{\widehat{r}_b}{\delta_b} \quad (1 \le b \le L-2), \qquad B_0 = B_{L-1} = +\infty. \tag{17}$$

An interior block is eligible when $B_b \le \tau_B$. To avoid isolated sparse islands, we select the longest contiguous eligible block interval,

$$(a, c) = \operatorname*{arg\,max}_{1 \le i \le j \le L-2} (j - i + 1) \quad \text{s.t.} \quad B_b \le \tau_B \ \ \forall b \in \{i, \dots, j\}, \tag{18}$$

and apply the same structural constraint along the denoising trajectory by selecting the longest contiguous eligible step interval,

$$(p, q) = \operatorname*{arg\,max}_{0 \le i \le j \le T-1} (j - i + 1) \quad \text{s.t.} \quad \widehat{u}_s \le \tau_T \ \ \forall s \in \{i, \dots, j\}. \tag{19}$$

For either axis, ties are resolved first by the lower mean score and then by the smaller starting index. We set $\mathcal{M}_B = \{a, \dots, c\}$, $\mathcal{M}_T = \{p, \dots, q\}$, and use their Cartesian product as the structured sparse region. For visualizing the joint block–step risk before this structural projection, we define

$$R_{b,s} = \max\left\{\frac{B_b}{\tau_B}, \frac{\widehat{u}_s}{\tau_T}\right\}, \qquad G(b, s) = \mathbf{1}[b \in \mathcal{M}_B]\,\mathbf{1}[s \in \mathcal{M}_T], \tag{20}$$

so locations with $R_{b,s} > 1$ are protected, while the interval constraints prevent fragmented sparse masks.

Around the selected block boundaries, Wan has $B_1 = 9.795$, $B_{37} = 9.363$, and $B_{38} = 20.267$; HunyuanVideo has $B_5 = 10.600$, $B_6 = 6.613$, $B_{50} = 8.125$, and $B_{51} = 16.033$. Consequently, both block masks are unchanged for any common threshold $\tau_B \in [9.795, 10.600)$; we use the round default $\tau_B = 10$. Along the step axis, Wan has $\max_{s \le 2} \widehat{u}_s = 0.640$ and $\widehat{u}_3 = 1$, while HunyuanVideo has $\max_{s \le 6} \widehat{u}_s = 0.909$ and $\widehat{u}_7 = 1$. We use $\tau_T = 0.95$, which lies in the common stable interval between the largest eligible-step score and the final-step score. The resulting intervals are $\mathcal{M}_B = \{1, \dots, 37\}$ and $\mathcal{M}_T = \{0, 1, 2\}$ for Wan 2.2, and $\mathcal{M}_B = \{6, \dots, 50\}$ and $\mathcal{M}_T = \{0, \dots, 6\}$ for HunyuanVideo 1.5. Thus, both axes remain contiguous, and the selected step intervals correspond to $T_{\text{tail}} = 1$ in Section 4.1.

Algorithm 1 CV calibration and FIS-DiT deployment.

Input: calibration prompts $\mathcal{D}_{\text{cal}}$, stride n, fixed thresholds $\tau_B = 10$ and $\tau_T = 0.95$
Run one dense statistics pass on $\mathcal{D}_{\text{cal}}$ and calculate B_b and $\widehat{u}_s$ using Eqs. 16–17.
Find (a, c) and (p, q) using Eqs. 18–19; freeze $\mathcal{M}_B = \{a, \dots, c\}$ and $\mathcal{M}_T = \{p, \dots, q\}$.
for each denoising step t and block l **do**
    if $l \notin \mathcal{M}_B$ or $t \notin \mathcal{M}_T$ **then**
        Evaluate the block densely.
    else
        Set $r_l = q(l) \bmod n$; gather and evaluate anchors $\mathcal{A}_l$; reconstruct skipped-frame outputs.
    end if
end for

B Additional Method Analysis

B.1 Local Interpolation Error

Let $X_b(\tau)$ be a continuous interpolation of the dense feature trajectory over frame coordinate τ within block b. For neighboring anchors $a < c$, suppose X_b is twice differentiable on $[a, c]$ and

$\sup_{\tau\in[a,c]} \|X_b''(\tau)\|_2 \le M_b$. Linear interpolation of the *dense* anchors yields, for $a < f < c$,

$$\|X_b(f) - \widetilde{X}_b(f)\|_2 \le \frac{M_b}{2}(f-a)(c-f) \le \frac{M_b}{8}(c-a)^2. \tag{21}$$

FIS-DiT computes anchors under a sparse frame context. If their deviations from dense anchors are ϵ_a and ϵ_c, then with $\lambda = (f-a)/(c-a)$ its reconstructed output satisfies

$$\|X_b(f) - \widehat{X}_b(f)\|_2 \le \frac{M_b}{8}(c-a)^2 + (1-\lambda)\epsilon_a + \lambda\epsilon_c. \tag{22}$$

Thus the interpolation component grows quadratically with anchor spacing under this smoothness assumption, while sparse-context error is a separate term. These are local feature bounds; neither equation bounds final-video quality for the nonlinear denoising network.

What CV measures. CV in Eq. 5 measures how uneven adjacent-frame relative changes are within a block and step. A low CV means those changes have similar magnitudes; it does not imply small reconstruction error or provide an upper bound on M_b. The protected-block rule uses CV as an empirical indicator of temporal non-uniformity. This distinction also explains why the depth-wise relative errors in Table 6 can vary across blocks despite broadly low CV in middle blocks.

B.2 Interleaving and Reconstruction Depth

Let $e_l(f)$ be the deviation of frame f from the dense trajectory entering block l. A first-order local expansion gives

$$e_{l+1}(f) \approx J_l(f)e_l(f) + \delta_l^{\mathrm{ctx}}(f) + \delta_l^{\mathrm{int}}(f), \tag{23}$$

where J_l is the local Jacobian, δ_l^{ctx} is sparse-context perturbation, and δ_l^{int} is the new interpolation residual. Selecting f as an anchor removes $\delta_l^{\mathrm{int}}(f)$ for that block, though prior and context errors may remain.

Proposition 1 (Bounded consecutive reconstruction depth)**.** *Under the phase rule $r_l = q(l) \bmod n$ and anchor set in Eq. 12, each non-boundary frame is directly evaluated once in every n consecutive sparse blocks. It is therefore reconstructed in at most $n-1$ consecutive sparse blocks.*

Proof. The sparse-block counter $q(l)$ increases by one at each sparse block. Across n consecutive sparse blocks, $q(l) \bmod n$ visits every residue in $\{0,\dots,n-1\}$ exactly once. Any frame f matches one residue $f \bmod n$ and is an anchor for that phase. The other at most $n-1$ phases reconstruct it. Boundary frames are anchors in every phase. □

Indexing the phase by absolute block number would lose this property when protected blocks interrupt the sparse sequence. Proposition 1 limits consecutive *new interpolation* at each frame; it does not reset existing error or imply a global bound without assumptions on the Jacobian norms in Eq. 23.

B.3 Approximate Compute Accounting

An idealized sparse-block cost can be written as

$$c(n) \approx (1-\alpha-\beta) + \frac{\alpha}{n} + \frac{\beta}{n^2}, \tag{24}$$

where α and β represent cost components approximately linear and quadratic in the selected frame fraction, respectively. If a fraction ρ_B of blocks and ρ_T of denoising steps are eligible, the approximate normalized DiT cost is $1-\rho_B\rho_T(1-c(n))+C_{\mathrm{interp}}$, where C_{interp} accounts for gather, scatter, and reconstruction. This model describes why protected blocks and dense tail steps limit the gain from a larger stride. Measured latency also includes hardware and pipeline overhead, as reflected in Table 1.

C Empirical Diagnostics: Prompt Set and Depth-wise Error

Figure 1 uses three prompt types: fast aquatic motion (P0), non-rigid animal motion (P1), and slower motion with a stylized background (P2). The exact prompts are:

- P0: "A shark swimming in clear Caribbean ocean";
- P1: "a cat playing in park";
- P2: "A boat sailing leisurely along the Seine River with the Eiffel Tower in background by Vincent van Gogh".

Table 6 reports the prompt-wise relative deviation of intermediate feature-magnitude maps on Wan 2.2, averaged over sparse denoising steps. The error rises in the middle blocks and returns toward the entry level near the exit for all three prompts. On average, it rises from 4.1% at entry to 12.4% in the middle and returns to 4.4% near the exit, providing evidence against monotonic depth-wise growth in this rollout without claiming a global bound on final-video error. CV describes the uniformity of adjacent-frame changes, so a low CV need not imply a low reconstruction deviation.

Table 6: Depth-wise relative feature-magnitude-map deviation on Wan 2.2, averaged over sparse denoising steps.

Prompt	Entry (2–9)	Middle (18–25)	Exit (32–39)
P0 – shark	4.1%	12.6%	5.3%
P1 – cat	4.3%	12.2%	4.4%
P2 – painterly boat	4.0%	12.4%	3.7%
Mean	**4.1%**	**12.4%**	**4.4%**

Qualitative examples. Figure 5 uses the following exact prompts: "Wide cinematic shot of a lush wilderness at sunrise, mist over a valley"; "A professional surfer riding a massive wave in a tropical ocean, water splashing"; and "Time lapse of a blooming flower, petals unfurling in high detail." They represent an atmospheric scene, large motion, and continuous fine-grained change, respectively.