HEAVY-FIRST, LIGHT-LATE: REVERSING CAPACITY SCHEDULING FOR EFFICIENT VIDEO DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion transformers achieve state-of-the-art video generation quality but incur prohibitive inference latency due to repeated full-capacity denoiser evaluations across sampling steps. Many multi-capacity methods follow a coarse-to-fine paradigm, allocating higher computational budgets to late-stage refinement. In contrast, we show that capacity demands peak during early denoising: **early steps determine global spatiotemporal trajectories, while late steps perform localized texture synthesis.** We formalize this property as **trajectory anchoring**, showing that executing a brief prefix of early steps with a large model locks the generative path, after which a compact model can complete the remaining steps with minimal fidelity loss. Motivated by this observation, we propose **ANCHOR** (**A**daptive **N**etwork **C**apacity **H**andoff for **O**ptimal **R**outing), a training-free cross-scale inference framework that dynamically routes computation between heterogeneous diffusion transformers. ANCHOR introduces a coordinate alignment mechanism to reconcile latent representations across model scales without fine-tuning, alongside an adaptive handoff predictor that determines sample-specific transition points from early trajectory dynamics. Comprehensive evaluations on Wan2.1, CogVideoX, and SkyReels-V2 demonstrate up to 3.7× speedup while preserving high frame-aligned similarity to the full-Large reference. Furthermore, our adaptive routing exposes superior fidelity-speed Pareto operating points by dynamically assigning longer prefixes to complex video samples.



Figure 1: **Model-capacity allocation comparison.** Conventional Small-first (right, ☹) schedules cause noticeable content and style drift, whereas our Large-first (left, ☺) schedule maintains high matched-reference fidelity to the full-Large baseline. Prompts are listed in Appendix H.

1 INTRODUCTION

Diffusion transformers (DiTs) achieve state-of-the-art video generation quality, yet their multi-step sampling incurs prohibitive latency by repeatedly evaluating billion-parameter models at full capacity. While existing caching and token-skipping methods reduce intra-step computation (Ma et al., 2024; Zhao et al., 2025b; Zhang et al., 2025b; Zhao et al., 2025a), an orthogonal yet fundamental question remains unaddressed: **how should model capacity itself be dynamically allocated across the temporal denoising trajectory?**

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.

Existing temporal-allocation paradigms mainly follow a coarse-to-fine heuristic: using lightweight or structure-forming computation during early noisy phases and reserving more expressive or specialized processing for late-stage refinement. T-Stitch executes a Small denoiser before a Large one, T-GATE retains late self-attention in its "fidelity-improving" phase, and SDXL sends base-model latents to a specialized low-noise refiner (Pan et al., 2024; Liu et al., 2025b; Podell et al., 2023). Despite their different mechanisms, these designs share the premise that later computation is especially valuable.

Early-Stage Dominance. Contrary to the coarse-to-fine consensus, we reveal that *model capacity exhibits the opposite temporal utility*. Under a fixed compute budget, *allocating large-model execution to a contiguous early prefix achieves substantially higher fidelity* to full-scale reference videos (matching prompt, seed, and initial noise) than late-stage placement. As illustrated in Figure 1, conventional small-first schedules cause severe trajectory drift in subject identity, spatial layout, and motion continuity. Conversely, *prioritizing large-model capacity early firmly establishes the global spatiotemporal manifold*, while subsequent steps remain remarkably resilient to capacity downscaling.

Key Insight. We formalize this mechanism as **trajectory anchoring**. Early denoising steps dictate the global semantic topology of the video; once an under-parameterized model steers generation into a divergent basin, late-stage high-capacity models cannot recover the intended trajectory. Furthermore, contiguous early allocation consistently outperforms sparse or uniform interleaving under identical compute budgets, confirming that early steps act collectively to lock the generative path. This motivates a cross-scale execution schedule, denoted as $L^k S^{N-k}$, where a large foundation model L executes k prefix steps before handing off to a lightweight model S for the remaining $N - k$ steps.

Idea. Building on these insights, we propose **ANCHOR** (**A**daptive **N**etwork **C**apacity **H**andoff for **O**ptimal **R**outing), a training-free cross-scale inference framework that dynamically routes computation across heterogeneous DiTs. ANCHOR resolves two fundamental deployment bottlenecks: (1) *Effective-SNR coordinate alignment*, a training-free mechanism that reconciles latent coordinate and variance mismatches across distinct noise parameterizations without fine-tuning; and (2) *Difficulty-aware adaptive handoff*, a lightweight diagnostic that evaluates early trajectory convergence to dynamically budget sample-specific prefix lengths along the speed-fidelity Pareto frontier (Figure 5).

Our main contributions are summarized as follows:

- We discover and formalize **trajectory anchoring**, demonstrating that an early contiguous large-model prefix is far more critical for video fidelity than late-stage or sparse capacity allocation across multiple foundation models.
- We propose a training-free **effective-SNR coordinate alignment** mechanism that enables seamless, zero-shot latent handoffs between structurally heterogeneous diffusion models.
- We design a lightweight **adaptive handoff predictor** that dynamically budgets model capacity based on sample difficulty, navigating superior fidelity-speed trade-offs across Wan2.1, CogVideoX, and SkyReels-V2.

2 Background and Related Work

2.1 Video Diffusion and Multi-Step Sampling

Video Diffusion Transformers (DiTs) represent video latents as 3D spatiotemporal token grids of shape $T \times H \times W$ and iteratively denoise them across a discrete schedule of timesteps (Peebles & Xie, 2023; Wan Team, 2025a; Yang et al., 2025). Flattening these tokens into unified sequences enables comprehensive cross-frame attention, but incurs quadratic $O((THW)^2)$ complexity in computation and activation footprint. Furthermore, repeatedly evaluating the full DiT backbone across all diffusion steps scales serving latency and memory traffic linearly with the number of function evaluations (Figure 3). This severe multi-step overhead makes executing uniform, heavy models across the entire trajectory computationally prohibitive.

2.2 Efficient Diffusion Inference

Prior acceleration schemes target either sampling trajectory length or intra-step computation.

Trajectory Distillation. Methods such as DMD2 and rCM (Yin et al., 2024; Zheng et al., 2026) distill multi-step schedules into few-step samplers. However, distillation incurs prohibitive retraining overheads and frequently degrades fine-grained motion dynamics and text adherence in complex video synthesis.

Intra-Step Caching and Pruning. Rather than shortening the trajectory, orthogonal methods reduce single-step FLOPs by caching temporal representations (Ma et al., 2024; Liu et al., 2025a), sparsifying spatiotemporal attention (Zhang et al., 2025a;b), or dynamically pruning layer width (Zhao et al., 2025a).

Our Distinction. These paradigms optimize computation within an unvarying architectural scale. In contrast, our approach introduces an orthogonal axis: *dynamically modulating model capacity across sampling timesteps via cross-scale switching without modifying or retraining the underlying backbones.*

2.3 Capacity Allocation across Denoising Stages

The Coarse-to-Fine Convention. Existing stage-wise methods rely on a coarse-to-fine intuition, allocating smaller capacity to early noisy steps and reserving heavy computation for late detail synthesis (Podell et al., 2023; Pan et al., 2024; Liu et al., 2025b). Moving beyond static pairings, Wan2.2 (Wan Team, 2025b) introduces noise-specialized temporal MoE experts; however, both experts retain identical 14B parameter footprints, addressing functional specialization rather than capacity reduction.

Cross-Scale Handoff Limitations. SRDiffusion (Cheng et al., 2025) explores early-large, late-small scheduling, but exhibits two vulnerabilities: (i) it assumes identical latent shapes ensure compatibility, ignoring coordinate and variance misalignments across disparate backbones, and (ii) its heuristic handoff relies on relative update thresholds that fail under open-domain prompt shifts.

Our Paradigm. We depart from coarse-to-fine conventions by establishing that **early trajectory anchoring** prevents irreversible manifold divergence. We resolve cross-network latent drift via training-free **effective-SNR coordinate alignment** and govern dynamic transitions with an **adaptive handoff predictor**, yielding consistent speed-fidelity trade-offs across video foundation models.

3 Understanding Trajectory Anchoring in Video Diffusion

3.1 Problem Formulation

Consider a multi-step video diffusion or flow-based generative process governed by an N-step discretization schedule $\{t_n\}_{n=0}^{N}$. Let x_{t_n} denote the latent state at step n, and let $v_\theta(x_{t_n}, t_n, c)$ represent the denoiser prediction conditioned on text prompt c. We investigate cross-scale scheduling between a high-capacity foundation model L (Large) and a compact counterpart S (Small). Under a uniform schedule, full-scale generation is denoted by L^N, while S^N represents the low-capacity baseline. We define the hybrid execution schedule as $L^k S^{N-k}$, where model L executes an initial prefix of k steps ($0 \le k \le N$) before handing off intermediate generation to model S for the remaining $N-k$ steps.

After handoff, the canonical Large-model scheduler state is preserved; only the denoiser backbone switches. From a system perspective, we sequentially offload model L before loading model S into device memory. This non-overlapping execution ensures that peak GPU memory never exceeds that of running model L alone, with the one-time weight swapping overhead consuming less than 2% of the overall end-to-end inference latency.

To quantify trajectory preservation relative to the full-capacity baseline under identical prompts, seeds, and initial Gaussian noise, we formulate the *matched-reference fidelity* as:

$$F(k) = \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \cos\left(\Phi\left(\hat{I}_\tau^{(k)}\right), \Phi\left(I_\tau^{(N)}\right)\right), \tag{1}$$

where $\hat{I}_\tau^{(k)}$ and $I_\tau^{(N)}$ denote the synthesized video frames at temporal index $\tau \in \mathcal{T}$ generated by $L^k S^{N-k}$ and L^N, respectively, and $\Phi(\cdot)$ denotes the normalized visual embedding function of CLIP

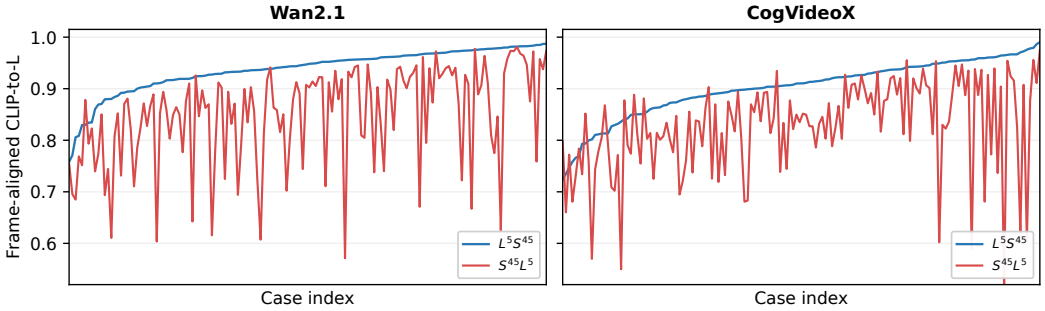


Figure 2: **Paired placement comparison.** Wan2.1 (left) and CogVideoX (right) compare a contiguous 10% Large prefix against an equal-budget Large suffix under identical prompts, seeds, and initial noise. Cases are sorted by $F(k)$.

ViT-B/32 (Radford et al., 2021). We report the temporal average across all aligned frames, with consistent fidelity trends confirmed across alternative vision-language encoders in Appendix E.

3.2 Temporal Placement: Early vs. Late Capacity Allocation

Early-Stage Dominance. To investigate temporal capacity sensitivity, we compare execution schedules under identical Large-model compute budgets but opposing temporal placements. As shown in Table 1, allocating capacity to a late Large-model suffix fails to recover the generative trajectory diverted by initial Small-model steps. In contrast, an early Large-model prefix preserves trajectory alignment across both Wan2.1 and CogVideoX (Figure 2). This demonstrates that *early denoising steps govern generative dynamics*, rendering late-stage high-capacity corrections ineffective.

Table 1: Prefix vs. suffix at matched Large-model budgets.

Family	Prefix	Suffix	Prefix F	Suffix F
Wan2.1	L^5S^{45}	$S^{45}L^5$	0.937	0.851
CogVideoX	L^5S^{45}	$S^{45}L^5$	0.902	0.827
SkyReels-V2	L^3S^{27}	$S^{27}L^3$	0.870	0.705

3.3 Temporal Contiguity: Continuous Prefix vs. Periodic Refreshing

Contiguity over Sparsity. Having established the advantage of early placement, we next examine whether Large-model capacity must form a continuous temporal prefix or can be distributed as isolated refresh steps. Such periodic activation is widely employed in activation-caching frameworks such as DeepCache and FORA (Ma et al., 2024; Selvaraju et al., 2024). We evaluate three periodic schedules against our contiguous prefix $L^{10}S^{40}$ under a matched budget of 10 Large-model steps.

Table 2: Contiguous vs. periodic on Wan2.1.

Schedule	CLIP-to-L
strategy-1	0.9488
strategy-2	0.9505
strategy-3	0.9544
$L^{10}S^{40}$	**0.9635**

Cooperative Anchoring. As reported in Table 2, the contiguous schedule $L^{10}S^{40}$ outperforms all periodic interleaving strategies. The exact Large-step placements used by the three periodic strategies are listed in Appendix F. This indicates that *trajectory anchoring is not governed by isolated critical timesteps, but emerges from consecutive early Large-model updates acting collectively to lock the trajectory within the correct basin*.

3.4 Prefix Scaling and Capacity Allocation Principle

Monotonic Convergence. We evaluate the impact of scaling anchoring prefix length k across all three foundation model families. As reported in Table 4(b), extending the contiguous Large prefix from 10% to 30% of total sampling steps improves fidelity $F(k)$ and perceptual quality. This trend confirms that *additional early Large-model steps progressively constrain the cross-scale latent trajectory toward the uncompressed full-Large realization*, providing an explicit control lever for navigating the speed-fidelity trade-off.

The Heavy-First, Light-Late Principle. These findings formalize trajectory anchoring into a unified scheduling rule: *Heavy-First, Light-Late*. Under a fixed compute budget, capacity provides maximal marginal fidelity during initial denoising rather than late refinement. Furthermore, this capacity must

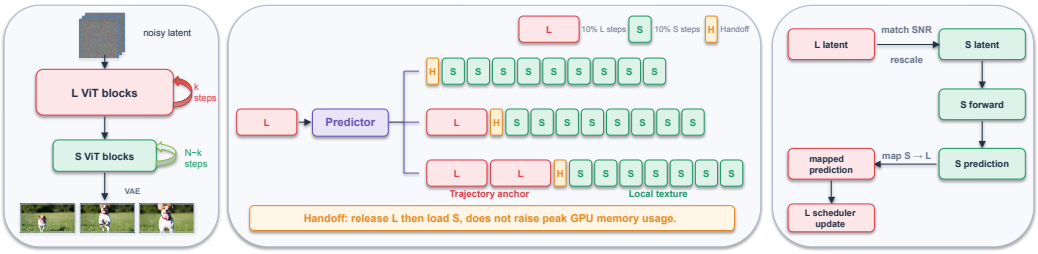


Figure 3: **Overview of the ANCHOR cross-scale video diffusion framework.** Left: Cross-scale execution pipeline, where model L anchors the early trajectory for k steps before model S completes the remaining $N - k$ steps prior to VAE decoding. Middle: Sample-adaptive capacity allocation, where a lightweight predictor diagnoses early trajectory dynamics to determine budget k, coupled with non-overlapping model swapping. Right: Zero-shot coordinate alignment, reconciling effective-SNR discrepancies and remapping predictions into the canonical scheduler frame.

be concentrated into a continuous initial prefix rather than dispersed across periodic refresh steps. Once anchored, compact models complete the trajectory with negligible drift.

4 ANCHOR: Cross-Scale Dynamic Inference Framework

4.1 Framework Overview

We introduce **ANCHOR**, a training-free cross-scale inference framework that dynamically routes computation between heterogeneous diffusion transformers. As illustrated in Figure 3, ANCHOR couples a high-capacity foundation model L with a compact counterpart S under an execution schedule $L^k S^{N-k}$. To transition from theoretical principles to robust deployment, ANCHOR resolves two critical challenges:

- **Latent coordinate mismatch:** Heterogeneous architectures parameterize noise schedules and latent spaces differently. We introduce a training-free *effective-SNR coordinate alignment* mechanism that dynamically reconciles cross-model representations with near-zero runtime overhead (Section 4.2).
- **Sample capacity heterogeneity:** Video generation instances exhibit varied structural complexity. We design an *adaptive handoff predictor* that diagnoses early trajectory dynamics to assign sample-specific prefix budgets along the speed-fidelity Pareto frontier (Section 4.3).

4.2 Zero-Shot Effective-SNR Coordinate Alignment

Representation Incompatibility. While the $L^k S^{N-k}$ schedule delegates late-stage synthesis to model S, direct state transfer naively assumes that identical numeric latent tensors represent equivalent physical noise states across different model scales. Formally, for a diffusion model $m \in \{L, S\}$, the perturbed latent state at timestep t is parameterized as:

$$x_t^{(m)} = \alpha_m(t) s_m z_0 + \sigma_m(t)\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{2}$$

where z_0 represents the clean unscaled latent, $s_m \in \mathbb{R}^+$ denotes the model-specific latent scaling factor, and $\alpha_m(t), \sigma_m(t)$ define the noise schedule. Due to divergent pretraining setups, different model scales frequently employ distinct noise formulations and latent scalings. For instance, between CogVideoX-2B and CogVideoX-5B, identical numeric values represent entirely different signal-to-noise ratios, causing catastrophic trajectory collapse under naive latent swapping.

Effective-SNR Formulation. To bridge this representation gap without auxiliary fine-tuning or retraining, we define an *effective signal-to-noise ratio* coordinate $\lambda_m(t)$ incorporating the latent scale s_m:

$$\lambda_m(t) = \log\left(\frac{\alpha_m^2(t) s_m^2}{\sigma_m^2(t)}\right). \tag{3}$$

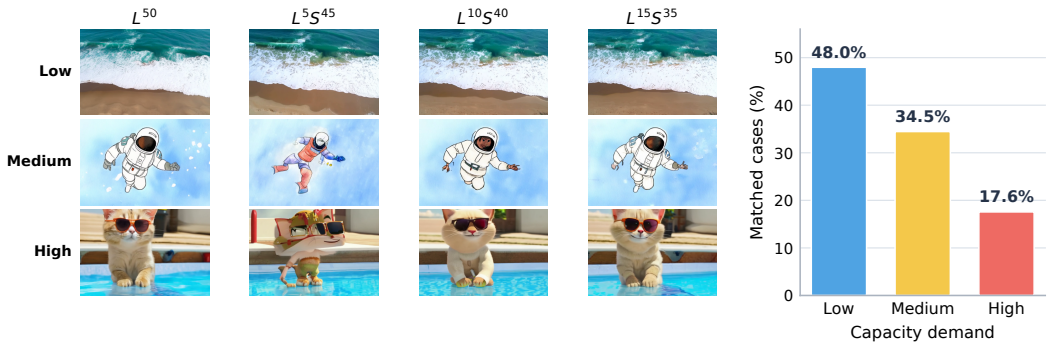


Figure 4: **Heterogeneous capacity demands across video instances.** Left: Representative middle frames generated under varied scheduling budgets. Right: Distribution of sample-wise capacity requirements on Wan2.1. Prompts are listed in Appendix H.

For each Small-model evaluation at Large-model timestep t_L, we identify the closest Small-model coordinate t_S by minimizing the effective-SNR discrepancy:

$$t_S = \arg\min_j \left|\lambda_S(j) - \lambda_L(t_L)\right|. \tag{4}$$

To harmonize variance discrepancies, the input latent is rescaled prior to evaluation by model S:

$$x_S = \frac{\sigma_S(t_S)}{\sigma_L(t_L)} x_L. \tag{5}$$

Closed-Loop Canonical Remapping. Let p_S denote the native prediction of model S. We first recover its clean-latent and noise coordinates using the analytical inverse associated with its scheduler:

$$(\hat{z}_0, \hat{\epsilon}) = \mathcal{D}_S(x_S, t_S, p_S). \tag{6}$$

We then express the same pair in the canonical prediction coordinates of model L:

$$p_L^* = \mathcal{E}_L(\hat{z}_0, \hat{\epsilon}, t_L), \tag{7}$$

after which the Large-model scheduler advances the trajectory using p_L^*. Both mappings are analytical and scheduler-specific. Their closed forms are given in Appendix A.

4.3 Difficulty-Aware Adaptive Handoff Predictor

Capacity Demand Variance. While the anchoring behavior is consistently observed across model families, the minimum required prefix length k varies substantially across samples (Figure 4). Simple visual scenes stabilize within a compact prefix ($k = 0.1N$), whereas complex geometries and rapid motion require extended anchoring ($k = 0.3N$). A static handoff schedule either risks visual collapse on complex instances or incurs redundant execution on simple ones.

Because the optimal trade-off space concentrates within the initial 30% of sampling steps, we define candidate handoff boundaries as:

$$k_j = 0.1jN, \qquad j \in \{1, 2, 3\}. \tag{8}$$

To determine the optimal boundary on the fly, our predictor evaluates trajectory dynamics accumulated strictly within the initial window $i \in [1, k_1]$ using two complementary metrics:

Trajectory Curvature Diagnostic. Let x_{i+1}^{solver} denote the state produced by the numerical solver, and let $x_{i+1}^{\text{Euler}} = x_i + (\sigma_{i+1} - \sigma_i) v_L(x_i, t_i, c)$ represent the first-order linear extrapolation under the Large-model velocity field. We formulate relative trajectory curvature as:

$$C_{\text{solver}} = \frac{1}{k_1} \sum_{i=1}^{k_1} \frac{\|x_{i+1}^{\text{solver}} - x_{i+1}^{\text{Euler}}\|_2}{\|x_{i+1}^{\text{solver}} - x_i\|_2 + \varepsilon}. \tag{9}$$

$\varepsilon > 0$ ensures numerical stability. Higher C_{solver} is associated with lower capacity demand, suggesting that pronounced solver corrections accompany faster trajectory stabilization.

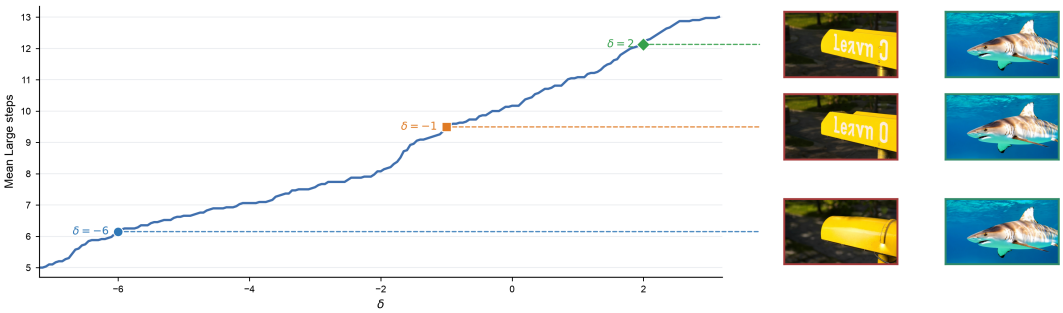


Figure 5: **Pareto trade-off control via threshold bias δ on Wan2.1.** Modulating δ arbitrates between throughput and fidelity. Complex samples (left) automatically receive higher capacity allocations to preserve detail, whereas simple samples (right) remain at minimal prefix budgets.

Spatiotemporal Spectral Energy. Let u_i denote the unconditional velocity prediction of model L at step i, and let $\mathcal{F}_{3D}$ denote the 3D discrete Fourier transform evaluated across temporal and spatial latent dimensions. We measure high-frequency spectral concentration within high-pass band Ω_H:

$$C_{\text{spectral}} = \max_{1 \le i \le k_1} \frac{\sum_{\omega \in \Omega_H} |\mathcal{F}_{3D}(u_i)(\omega)|^2}{\sum_{\omega} |\mathcal{F}_{3D}(u_i)(\omega)|^2}. \tag{10}$$

High-frequency energy reflects spatial edges and frame-to-frame temporal changes. Elevated C_{spectral} indicates unresolved geometric or motion structures, necessitating extended Large-model allocation.

Probabilistic Handoff Routing. We standardize both metrics into normalized scores z_{sol} and z_{spec} using the fixed statistics in Appendix B, and define the composite difficulty score:

$$d = z_{\text{spec}} - z_{\text{sol}}. \tag{11}$$

We assign class logits $a = (-d, 0, d)$ over $\{k_1, k_2, k_3\}$, deriving categorical probabilities via softmax:

$$\pi_j = \frac{\exp(a_j)}{\sum_{\ell=1}^{3} \exp(a_\ell)}. \tag{12}$$

We compute the expected budget $\bar{k} = \sum_{j=1}^{3} \pi_j k_j$ and project it onto discrete steps via adjustable threshold bias $\delta \in \mathbb{R}$:

$$\hat{k}_\delta = \begin{cases} k_1, & \bar{k} + \delta \le (k_1 + k_2)/2, \\ k_2, & (k_1 + k_2)/2 < \bar{k} + \delta \le (k_2 + k_3)/2, \\ k_3, & \bar{k} + \delta > (k_2 + k_3)/2. \end{cases} \tag{13}$$

Adjusting δ allows continuous traversal of the speed-fidelity Pareto frontier (Figure 5).

Predictor Evaluation. Step-wise heuristic thresholds, such as the relative update criterion in SRDiffusion (Cheng et al., 2025), frequently trigger false-stop errors due to transient local minima. Because cross-scale handoff is irreversible, premature exits induce severe visual collapse. Table 3 compares the resulting routing errors with ANCHOR, whose predictor evaluates curvature and spectral properties across the initial window.

Table 3: Routing errors on Wan2.1. Severe failure denotes hard cases receiving < 10 Large steps; redundant allocation denotes easy cases receiving > 10.

Method	Severe failure	Redundant allocation
SRDiffusion	50.0%	50.7%
ANCHOR	**3.8%**	**19.7%**

4.4 System Execution and Memory Management

Non-Overlapping Execution. To maximize memory efficiency during cross-scale inference, ANCHOR manages GPU memory through sequential device transfers. During the handoff boundary at step $\hat{k}_\delta$, we offload the weights of model L to host memory before loading model S. This non-overlapping execution guarantees that peak GPU memory never exceeds the memory footprint of

Table 4: **Main results across video DiT architectures.**

Family	Method / Schedule	Latency (s)	Speedup	LPIPS↓	SSIM↑	PSNR↑	CLIP-to-L↑	VBench↑
(a) Method Comparison and Composition								
Wan2.1	TeaCache (slow)	1094.8	1.54×	0.189	0.777	20.29	0.958	0.808
Wan2.1	TeaCache (fast)	777.7	2.16×	0.336	0.664	16.46	0.914	0.800
Wan2.1	MagCache	672.1	2.50×	0.069	0.911	27.50	0.988	0.816
Wan2.1	ANCHOR	590.4	2.85×	0.177	0.789	21.14	0.959	0.827
Wan2.1	ANCHOR + TeaCache	437.8	3.84×	0.172	0.799	21.57	0.961	0.826
Wan2.1	ANCHOR + MagCache	438.3	3.83×	0.182	0.792	21.31	0.958	0.828
(b) Capacity Scaling								
Wan2.1	10% prefix	454.3	3.70×	0.264	0.721	18.50	0.937	0.807
Wan2.1	20% prefix	591.8	2.84×	0.162	0.805	21.97	0.965	0.814
Wan2.1	30% prefix	750.4	2.24×	0.110	0.855	24.56	0.976	0.820
CogVideoX	10% prefix	191.6	2.11×	0.324	0.684	18.95	0.902	0.765
CogVideoX	20% prefix	200.2	2.02×	0.205	0.777	23.05	0.939	0.774
CogVideoX	30% prefix	224.6	1.80×	0.128	0.851	26.57	0.962	0.770
SkyReels-V2	10% prefix	706.6	3.49×	0.344	0.567	17.26	0.870	0.750
SkyReels-V2	20% prefix	906.7	2.72×	0.252	0.631	19.45	0.911	0.803
SkyReels-V2	30% prefix	1105.9	2.23×	0.201	0.677	20.95	0.939	0.809
(c) Adaptive budget control								
Wan2.1	$\delta = -6$	484.9	3.47×	0.238	0.740	19.34	0.945	0.809
Wan2.1	$\delta = -1$	576.5	2.92×	0.174	0.793	21.65	0.963	0.811
Wan2.1	$\delta = 2$	648.8	2.59×	0.139	0.827	23.17	0.971	0.816

running model L alone. The one-time model swapping introduces minimal overhead, accounting for less than 2% of total inference latency, which is readily amortized by the up to $3.7\times$ speedup achieved during the Small-model propagation phase. Detailed handoff overhead measurements are reported in Appendix K.

5 Experiments

5.1 Experimental Setup

Model Families and Paired Baselines. We evaluate ANCHOR across three representative open-source video DiT families spanning over 2,400 video generations: (1) **Wan2.1** (Wan Team, 2025a): pairing Wan2.1-14B (L) with Wan2.1-1.3B (S) under a 50-step flow-matching schedule at 720×1280 resolution (81 frames); (2) **CogVideoX** (Yang et al., 2025): pairing CogVideoX-5B (L) with CogVideoX-2B (S) across a 50-step schedule at 480×720 resolution (49 frames); and (3) **SkyReels-V2** (Chen et al., 2025): pairing SkyReels-V2-14B (L) with SkyReels-V2-1.3B (S) over a 30-step schedule at 540×960 resolution (97 frames). Every paired cross-scale generation strictly freezes the prompt, random seed, initial Gaussian noise, resolution, scheduler, and total step count N against its full-scale reference L^N. Latency profiles are benchmarked on dedicated NVIDIA H20 GPUs, with weight-swapping overheads fully incorporated into end-to-end measurements.

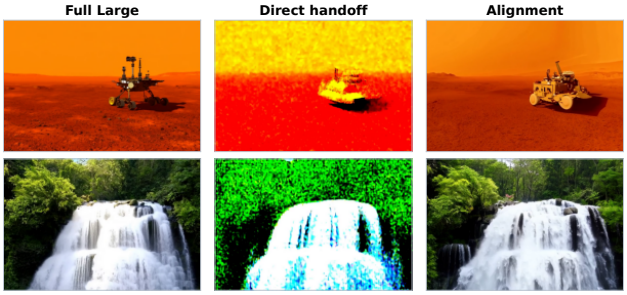
Metrics. We evaluate reference-trajectory fidelity via $F(k)$ (Equation (1)) using frame-aligned CLIP ViT-B/32 cosine similarity, with robustness across DINOv2 and SigLIP confirmed in Appendix E (Radford et al., 2021; Oquab et al., 2024; Zhai et al., 2023). Generation fidelity against the full-scale teacher is measured via frame-aligned LPIPS (AlexNet) (Zhang et al., 2018), SSIM (Wang et al., 2004), and PSNR (Huynh-Thu & Ghanbari, 2008), alongside standard perceptual quality benchmarks from VBench (Huang et al., 2024). System efficiency is quantified via end-to-end wall-clock speedup (inclusive of VAE decoding and tensor migration) and peak GPU memory footprint.

5.2 Main Results

Comparison with Existing DiT Acceleration Methods. Table 4(a) compares ANCHOR with representative training-free DiT acceleration methods on Wan2.1. ANCHOR achieves a $2.85\times$ end-to-end speedup while retaining a CLIP-to-L similarity of 0.959 and a VBench score of 0.827. Compared with TeaCache and MagCache, these results demonstrate that cross-scale capacity scheduling provides a competitive efficiency–fidelity operating point through a distinct acceleration mechanism.

Table 5: Ablation of coordinate alignment on CogVideoX under $L^{0.1N}S^{0.9N}$. Alignment resolves cross-scale coordinate mismatch and enables seamless state transfer.

Handoff Scheme	CLIP-to-L
Direct Handoff	0.693
Coordinate Alignment	0.901



Complementarity with Existing Acceleration Methods. ANCHOR operates along a different efficiency dimension from intra-step acceleration methods and can therefore be composed with them directly. As shown in Table 4(a), combining ANCHOR with MagCache increases the inference speedup from $2.85\times$ to $3.83\times$, while essentially preserving ANCHOR's matched-reference fidelity (0.959 vs. 0.958 CLIP) and standalone generation quality (0.827 vs. 0.828 VBench). These results show that cross-stage capacity allocation is complementary to intra-step acceleration methods rather than competing against them.

Adaptive Budget Control. ANCHOR's predictor supports different compute budgets. As reported in Table 4(c), shifting δ allows the same predictor to operate at different fidelity-efficiency targets without retraining or modifying the underlying models.

Impact of Latent Alignment. Table 5 confirms that direct handoff without reconciliation degrades generation fidelity, collapsing CLIP similarity to 0.693 despite identical spatial dimensions. Our training-free effective-SNR alignment recovers this to 0.901, confirming that cross-architecture handoff requires explicit coordinate and variance alignment.

Table 6: C_{spec} reduces under/over-allocation errors relative to a fixed prefix, while combining it with C_{sol} retains the lowest error rates, demonstrating that the two signals are complementary.

	Under-allocation	Over-allocation
Fixed L10	17.6%	48.0%
C_{sol} only	18.9%	43.2%
C_{spec} only	14.9%	39.9%
ANCHOR ($\delta = -0.2$)	14.9%	37.2%

Handoff Predictor. Table 6 evaluates trajectory diagnostics under matched compute budgets. Premature handoff (under-allocation) introduces trajectory drift by truncating the high-capacity prefix, whereas late handoff (over-allocation) yields diminishing fidelity returns at substantial latency costs. The proposed predictor better approximates the oracle transition point across varied generation prompts.

6 Conclusion

This work explores temporal capacity allocation as a pathway to efficient video diffusion. Across Wan2.1, CogVideoX, and SkyReels-V2, our findings confirm that high-capacity computation is most critical during an early contiguous prefix. Once the global spatiotemporal trajectory is established, a compact model can complete generation with minimal drift, establishing the empirical basis for our Heavy-First, Light-Late principle and the ANCHOR framework. ANCHOR realizes this principle via training-free coordinate alignment across heterogeneous latent spaces and sample-adaptive stage handoff. Across diverse open-source DiTs, ANCHOR provides a tunable speed-fidelity frontier, achieving up to $3.7\times$ end-to-end acceleration while composing orthogonally with intra-step methods such as feature caching. These results demonstrate that efficient diffusion inference requires optimizing not only overall FLOPs, but also the dynamic distribution of model capacity across distinct generative phases.

AI USE STATEMENT

In this work, we use generative AI tools for searching related work. We do not use AI tools for other research tasks requiring disclosure. We review all AI-assisted content and take full responsibility for the final paper.

REFERENCES

Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, et al. Skyreels-v2: Infinite-length film generative model. *arXiv preprint arXiv:2504.13074*, 2025. URL https://arxiv.org/abs/2504.13074.

Shenggan Cheng, Yuanxin Wei, Lansong Diao, Yong Liu, Bujiao Chen, Lianghua Huang, Yu Liu, Wenyuan Yu, Jiangsu Du, Wei Lin, and Yang You. SRDiffusion: Accelerate video diffusion inference via sketching-rendering cooperation. *arXiv preprint arXiv:2505.19151*, 2025. URL https://arxiv.org/abs/2505.19151.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2311.17982.

Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of PSNR in image/video quality assessment. *Electronics Letters*, 44(13):800–801, 2008. doi: 10.1049/el:20080522.

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. *arXiv preprint arXiv:2411.19108*, 2025a. URL https://arxiv.org/abs/2411.19108.

Haozhe Liu, Wentian Zhang, Jinheng Xie, Francesco Faccio, Mengmeng Xu, Tao Xiang, Mike Zheng Shou, Juan-Manuel Perez-Rua, and Jürgen Schmidhuber. Faster diffusion via temporal attention decomposition. *Transactions on Machine Learning Research*, 2025b. URL https://arxiv.org/abs/2404.02747.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2312.00858.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2304.07193.

Zizheng Pan, Bohan Zhuang, De-An Huang, Weili Nie, Zhiding Yu, Chaowei Xiao, Jianfei Cai, and Anima Anandkumar. T-stitch: Accelerating sampling in pre-trained diffusion models with trajectory stitching. *arXiv preprint arXiv:2402.14167*, 2024. URL https://arxiv.org/abs/2402.14167.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. URL https://arxiv.org/abs/2212.09748.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. URL https://arxiv.org/abs/2307.01952.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. URL https://arxiv.org/abs/2103.00020.

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. URL https://arxiv.org/abs/2407.01425.

Wan Team. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025a. URL https://arxiv.org/abs/2503.20314.

Wan Team. Wan2.2: Open and advanced large-scale video generative models. Official GitHub repository and model documentation, 2025b. URL https://github.com/Wan-Video/Wan2.2. Accessed 2026-09-21.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. doi: 10.1109/TIP.2003.819861.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2408.06072.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.14867.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. URL https://arxiv.org/abs/2303.15343.

Jintao Zhang, Haoxu Wang, Kai Jiang, Shuo Yang, Kaiwen Zheng, Haocheng Xi, Ziteng Wang, Hongzhou Zhu, Min Zhao, Ion Stoica, Joseph E. Gonzalez, Jun Zhu, and Jianfei Chen. SLA: Beyond sparsity in diffusion transformers via fine-tunable sparse-linear attention. *arXiv preprint arXiv:2509.24006*, 2025a. URL https://arxiv.org/abs/2509.24006.

Peiyuan Zhang, Yongqi Chen, Haofeng Huang, Will Lin, Zhengzhong Liu, Ion Stoica, Eric P. Xing, and Hao Zhang. VSA: Faster video diffusion with trainable sparse attention. *arXiv preprint arXiv:2505.13389*, 2025b. URL https://arxiv.org/abs/2505.13389.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. URL https://arxiv.org/abs/1801.03924.

Wangbo Zhao, Yizeng Han, Jiasheng Tang, Kai Wang, Yibing Song, Gao Huang, Fan Wang, and Yang You. Dynamic diffusion transformer. In *International Conference on Learning Representations*, 2025a. URL https://arxiv.org/abs/2410.03456.

Xuanlei Zhao, Xiaolong Jin, Kai Wang, and Yang You. Real-time video generation with pyramid attention broadcast. In *International Conference on Learning Representations*, 2025b. URL https://arxiv.org/abs/2408.12588.

Kaiwen Zheng, Yuji Wang, Qianli Ma, Huayu Chen, Jintao Zhang, Yogesh Balaji, Jianfei Chen, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Large scale diffusion distillation via score-regularized continuous-time consistency. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.08431.

A Scheduler-Specific Handoff Mappings

For the VP velocity prediction, the analytical mappings are

$$\hat{z}_0 = \frac{\alpha_S x_S - \sigma_S v_S}{s_S}, \qquad \hat{\epsilon} = \sigma_S x_S + \alpha_S v_S, \tag{14}$$

$$v_L^* = \alpha_L \hat{\epsilon} - \sigma_L s_L \hat{z}_0. \tag{15}$$

For the linear-flow convention, where $x_t = (1-t)sz_0 + t\epsilon$ and $u_t = \epsilon - sz_0$, the mappings become

$$\hat{z}_0 = \frac{x_S - t_S u_S}{s_S}, \qquad \hat{\epsilon} = x_S + (1-t_S)u_S, \tag{16}$$

$$u_L^* = \hat{\epsilon} - s_L \hat{z}_0. \tag{17}$$

Table 7: Cumulative ablation of coordinate alignment on CogVideoX.

	CLIP-to-L↑
Direct handoff	0.693
+ SNR match	0.716
+ Latent rescale	0.682
+ Prediction remap (OURS)	0.901

B Predictor Details

The normalization is $z = (C - \mu)/\sigma$. μ and σ are computed from 30 unlabeled trajectories each family. No learned parameters or labels are used, preserving training-free routing. In C_{spectral}, Ω_H contains the 3D FFT bins whose radial frequency exceeds 0.25 after each temporal and spatial frequency axis is normalized by its Nyquist frequency. We define k^* as the minimum prefix in $\{0.1N, 0.2N, 0.3N\}$ whose CLIP-to-L^{50} reaches 0.95; the low-, medium-, and high-demand labels correspond to $k^* = 0.1N$, $0.2N$, and $0.3N$, respectively.

C MATCHED-BUDGET ADAPTIVE ROUTING

We compare ANCHOR with fixed Large-prefix schedules under the same average Large-model budget. ANCHOR improves fidelity by reducing sample-wise misallocation.

Table 8: Matched-average-budget comparison between fixed-prefix routing and ANCHOR across three video diffusion model families.

Family	Method	LPIPS↓	SSIM↑	PSNR↑	CLIP-to-L↑
Wan2.1	Fixed L10	0.162	0.805	21.97	0.965
	ANCHOR	0.156	0.812	22.29	0.967
CogVideoX	Fixed L10	0.205	0.777	23.05	0.939
	ANCHOR	0.184	0.800	23.74	0.947
SkyReels-V2	Fixed L6	0.252	0.631	19.45	0.911
	ANCHOR	0.241	0.641	19.66	0.917

D Prefix and Suffix Placement Quality

Table 9: Matched-reference quality on Wan2.1 with the same budget of ten Large-model and forty Small-model steps.

Schedule	LPIPS↓	SSIM↑	PSNR↑	CLIP-to-L↑	VBench↑
$S^{40}L^{10}$	0.506	0.533	13.21	0.850	0.753
$L^{10}S^{40}$	0.162	0.805	21.97	0.965	0.814

E Alternative-Embedding Robustness

The Large-prefix advantage is not specific to CLIP ViT-B/32 (Radford et al., 2021). Table 10 re-evaluates the 148 CogVideoX cases with three additional visual encoders.

Although each encoder has its own cosine-similarity range, all encoders show the same trend as CLIP ViT-B/32, supporting our Heavy-First, Light-Late Principle.

Table 10: Mean matched-reference similarity under alternative image encoders on CogVideoX.

Embedding	S^{50}	$S^{45}L^{5}$	$L^{5}S^{45}$	$L^{15}S^{35}$
CLIP ViT-B/32	0.826	0.827	0.902	0.962
CLIP ViT-L/14	0.748	0.748	0.834	0.922
DINO ViT-B/16	0.537	0.538	0.751	0.915
SigLIP ViT-B/16	0.811	0.812	0.890	0.955

F Periodic Capacity-Allocation Schedules

The three periodic strategies in Table 2 place their ten Large-model steps at $\{0\text{–}4, 10, 20, 30, 40, 45\}$, $\{0\text{–}5, 15, 25, 35, 45\}$, and $\{0\text{–}6, 20, 35, 45\}$, respectively. All remaining denoising steps use the Small model.

G Exploratory Cross-Family Extension

The main experiments consider Large and Small models within the same model family. We additionally investigate whether the same heavy-first principle can extend to a harder cross-family setting, CogVideoX-5B $\rightarrow$ Wan2.1-1.3B, where the two models use different VAEs and denoising coordinates. Because their noisy latent states are not directly compatible, we transfer through an estimated clean state rather than reusing the source latent.

Specifically, we first recover the CogVideoX clean-state estimate $\hat{x}_0^C$, map it into the Wan latent space using a lightweight adapter G_ϕ trained on paired frozen-VAE latents, and then re-noise the mapped state at the closest Wan noise coordinate before resuming generation with the Wan scheduler and denoiser. The video denoisers and VAEs remain frozen throughout this procedure.

In this pilot experiment, clean-state restart avoids the severe degradation observed with direct cross-family state reuse: it achieves a mean CLIP similarity of 0.868 to the source video and preserves the main semantics. We therefore view this result as preliminary evidence that heavy-first cross-scale execution may extend beyond within-family pairs, rather than as part of ANCHOR's training-free alignment mechanism.

H Prompts for Main Qualitative Comparisons

Figure 1: Bear: "a bear sniffing the air for scents of food."; Snowy campfire: "Campfire at night in a snowy forest with starry sky in the background."; Stop sign: "a stop sign and a parking meter."

Figure 4: Waves: "A beautiful coastal beach in spring, waves lapping on sand, zoom in."; Watercolor astronaut: "An astronaut flying in space, watercolor painting."; Lifeguard cat: "A cat wearing sunglasses and working as a lifeguard at a pool."

I External Acceleration Settings

We use the released TeaCache and MagCache inference rules without changing model weights. TeaCache uses the released Wan2.1 polynomial rescaling rule. MagCache keeps the first 20% of its trajectory uncached and limits each reuse run to at most $K = 4$ consecutive steps. Thresholds are selected with reference to the recommended settings of each method and calibrated for budget-matched comparison, such that the mean compute budget is near 40% of the full-model budget in the caching range (60% for TeaCache-slow). Table 11 lists the exact settings used in Table 4(a).

Table 11: Threshold of TeaCache and MagCache.

Configuration	Threshold
TeaCache (slow)	0.1442
TeaCache (fast)	0.2723
MagCache	0.15637
TeaCache with ANCHOR	0.1115
MagCache with ANCHOR	0.17052

J Evaluation Cohorts for Table 4

Table 4(a) evaluates every method on the same 30 prompt–seed cases, with identical full-Large references. Each row in Parts (b) and (c) is evaluated on 148 cases.

K Handoff Overhead

Table 12: Relative overhead is the handoff overhead divided by the denoising latency of each schedule.

Family	Schedule	Latency (s)	Relative overhead (%)
Wan2.1	L^5S^{45}	1.067	0.24
Wan2.1	$L^{10}S^{40}$	1.067	0.18
Wan2.1	$L^{15}S^{35}$	1.067	0.15
CogVideoX	L^5S^{45}	0.945	0.54
CogVideoX	$L^{10}S^{40}$	0.945	0.47
CogVideoX	$L^{15}S^{35}$	0.945	0.42
SkyReels-V2	L^3S^{27}	11.582	1.71
SkyReels-V2	L^6S^{24}	11.582	1.32
SkyReels-V2	L^9S^{21}	11.582	1.07

L Additional Qualitative Results

We provide additional frame-level comparisons using the same prompt, seed, and initial noise within each row.

M Peak GPU Memory

Table 13: Peak GPU memory footprint across the three evaluated model families.

Model family	Full Large (GiB)	ANCHOR (GiB)
Wan2.1	64.81	64.81
CogVideoX	22.75	22.75
SkyReels-V2	49.83	49.83

ANCHOR releases the Large model before loading the Small model and therefore introduces no additional peak GPU memory footprint.

N Supplementary Material

The supplementary material includes the core implementation code of our method and comparison videos. The videos are organized into four directories: composition with caching, high demand, low demand, and prefix vs suffix.

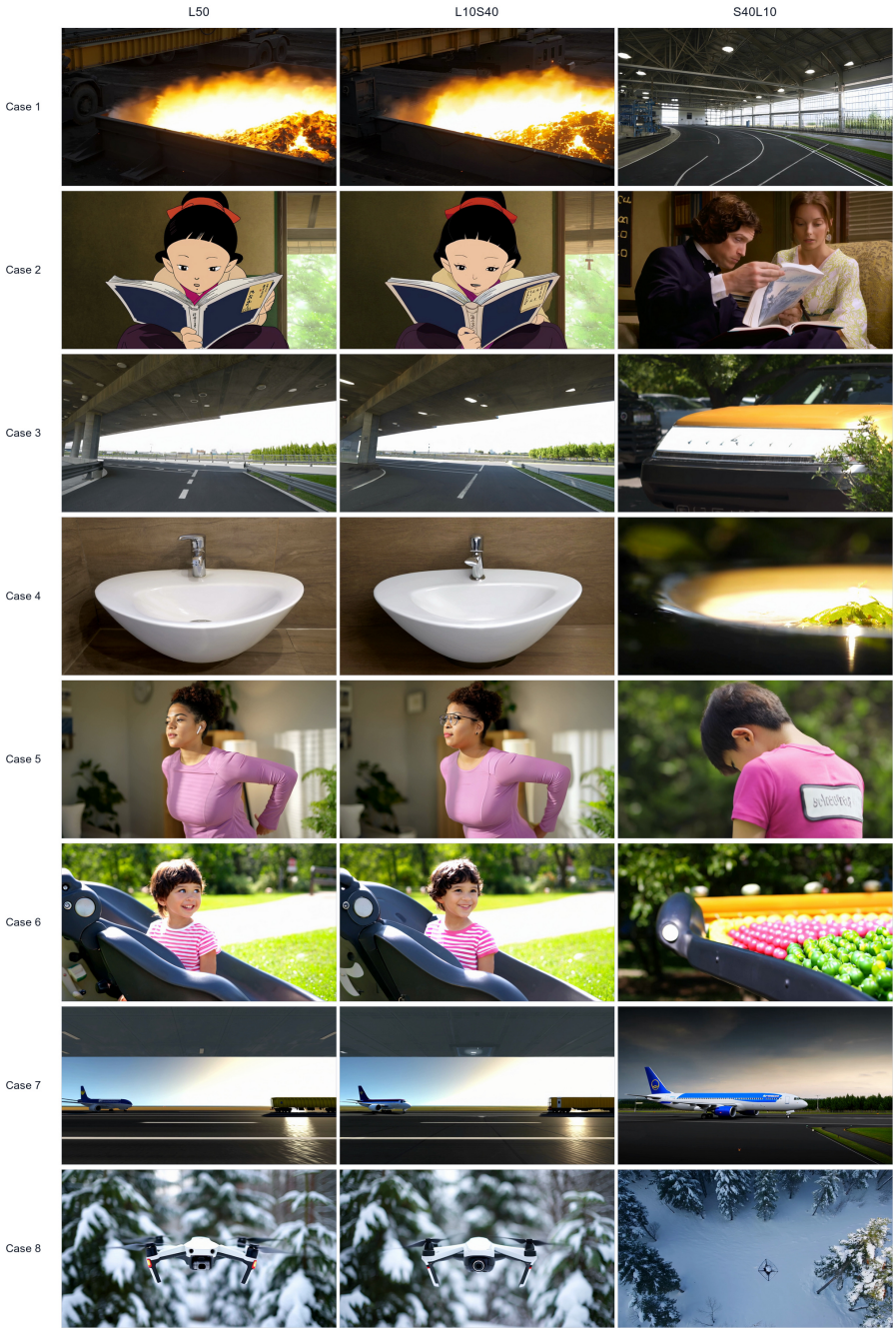


Figure 6: **Contiguous Large prefixes outperform same-budget Large suffixes.** Middle frames compare L^{50}, $L^{10}S^{40}$, and $S^{40}L^{10}$ across eight cases. From top to bottom, the prompts are "industrial area," "Gwen Stacy reading a book by Hokusai, in the style of Ukiyo," "parking lot," "a sink," "A person is bending back," "amusement park," "an airplane on the left of a train, front view," and "a drone flying over a snowy forest."

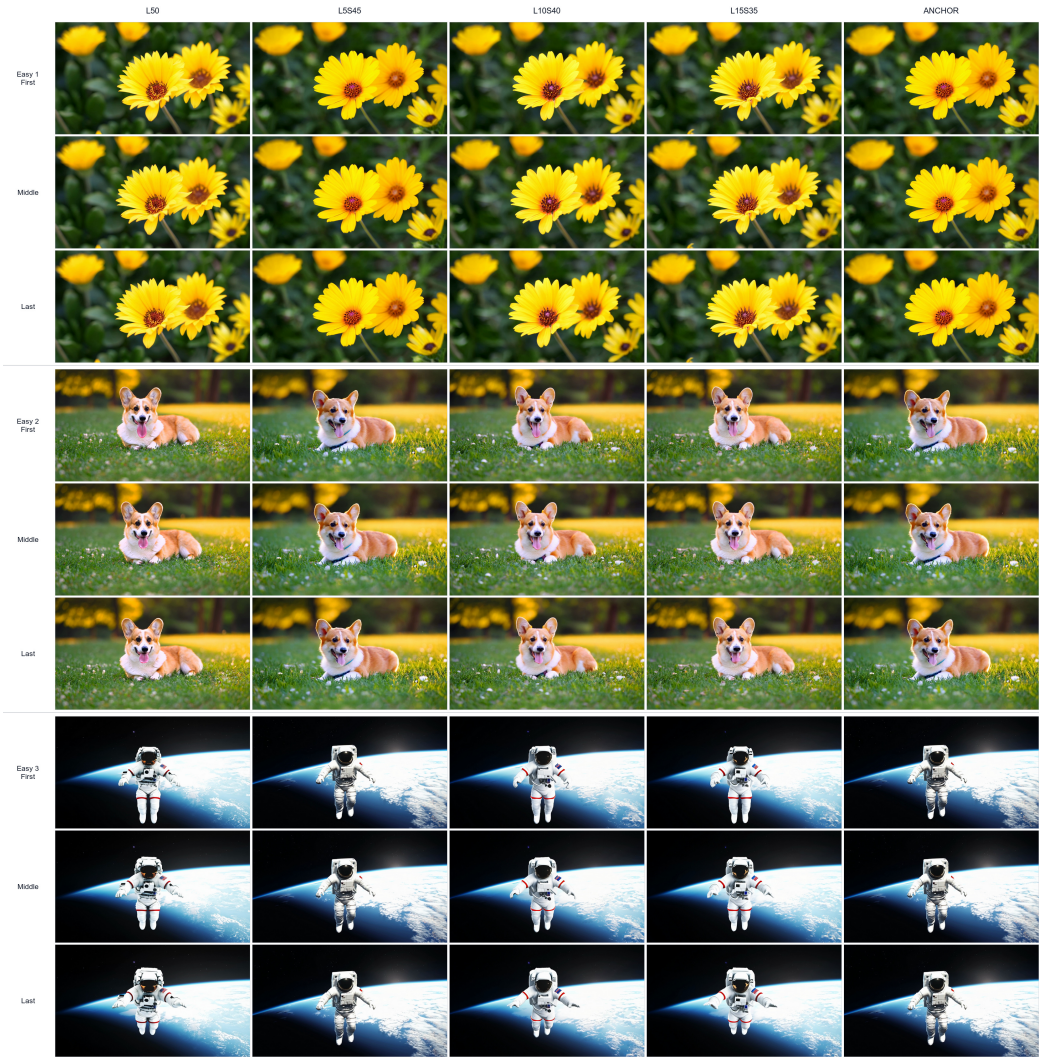


Figure 7: **ANCHOR on low-demand cases.** Each case occupies three rows (first, middle, and last frames). From top to bottom, the prompts are "Yellow flowers swing in the wind," "A cute happy Corgi playing in park, sunset, zoom out," and "An astronaut flying in space, with an intense shaking effect." ANCHOR selects L^5S^{45} for all three cases.

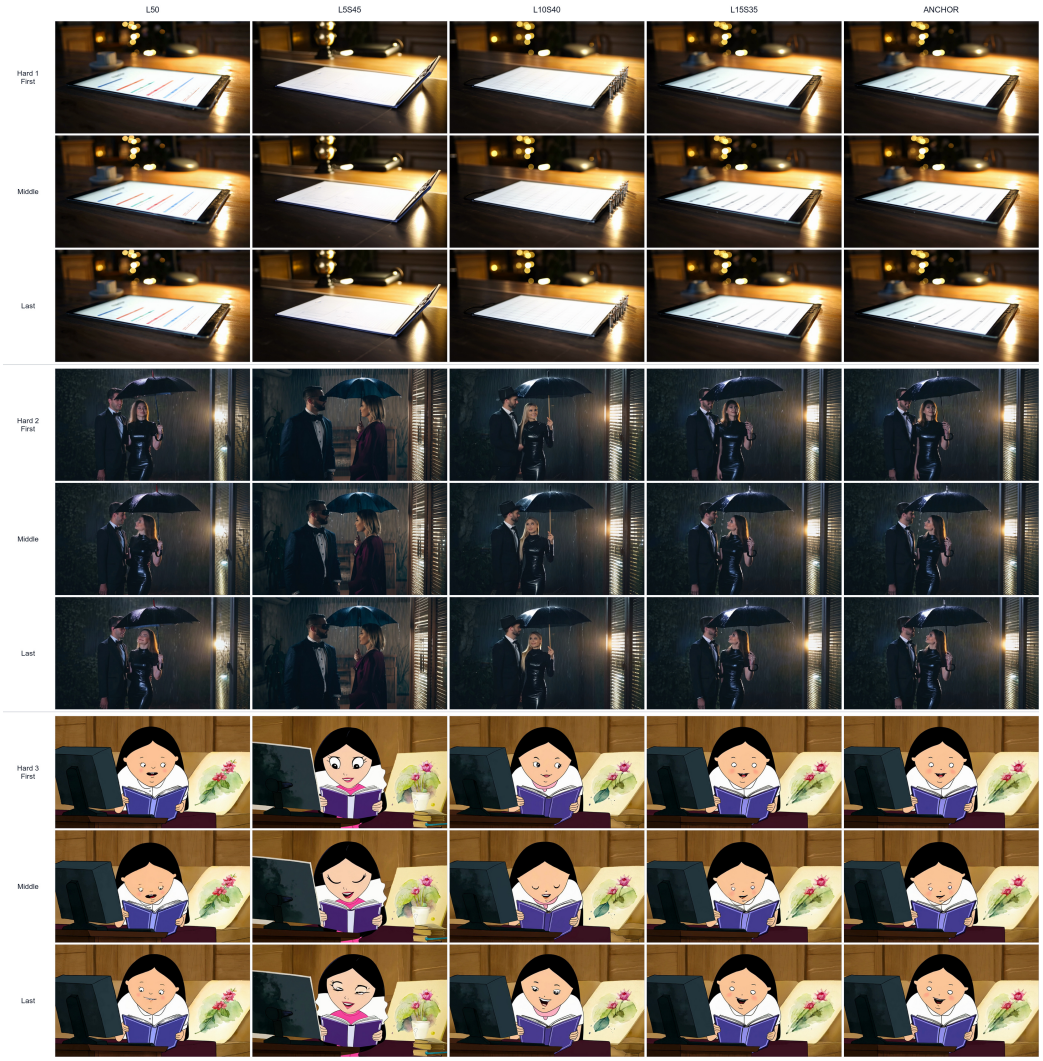


Figure 8: **ANCHOR on high-demand cases.** Each case occupies three rows (first, middle, and last frames). From top to bottom, the prompts are "science museum," "A couple in formal evening wear going home get caught in a heavy downpour with umbrellas, in cyberpunk style," and "Gwen Stacy reading a book, watercolor painting." ANCHOR selects $L^{15}S^{35}$ for all three cases.

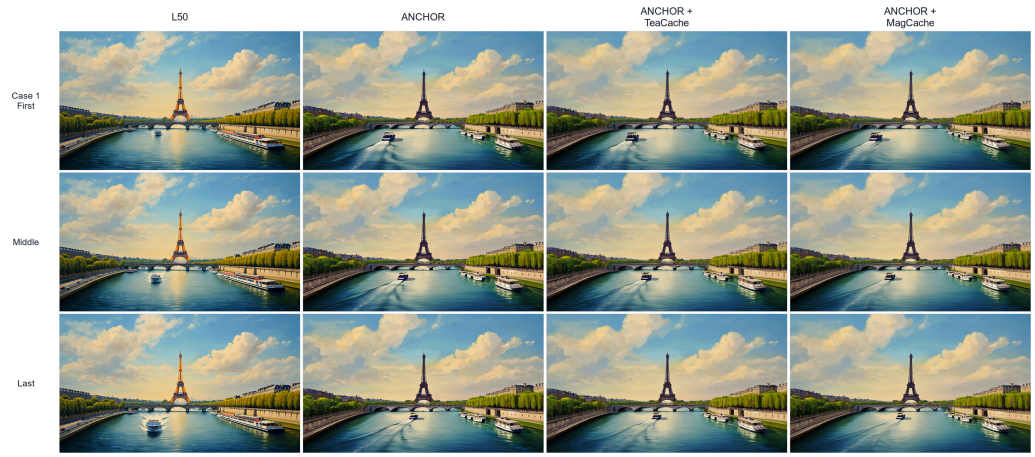


Figure 9: **Composition with caching.** Prompt: "A boat sailing leisurely along the Seine River with the Eiffel Tower in background, oil painting." Rows show the first, middle, and last frames.

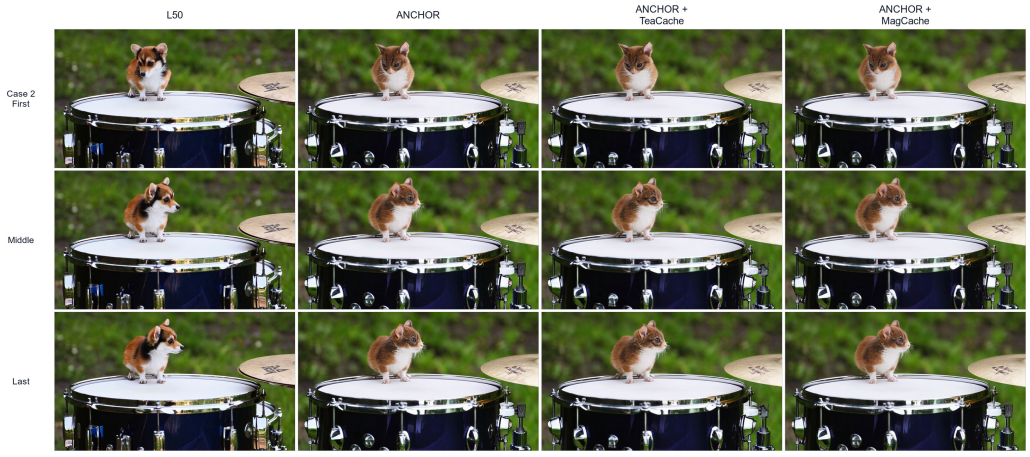


Figure 10: **Composition with caching.** Prompt: "A corgi is playing drum kit." Rows show the first, middle, and last frames.



Figure 11: **Composition with caching.** Prompt: "A couple in formal evening wear going home get caught in a heavy downpour with umbrellas, in cyberpunk style." Rows show the first, middle, and last frames.

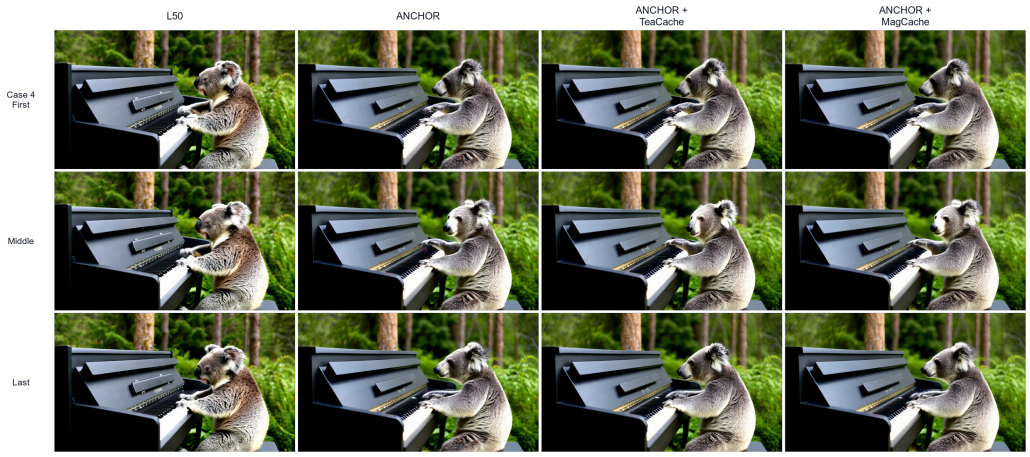


Figure 12: **Composition with caching.** Prompt: "A koala bear playing piano in the forest." Rows show the first, middle, and last frames.



Figure 13: **Composition with caching.** Prompt: "A shark swimming in clear Caribbean ocean." Rows show the first, middle, and last frames.

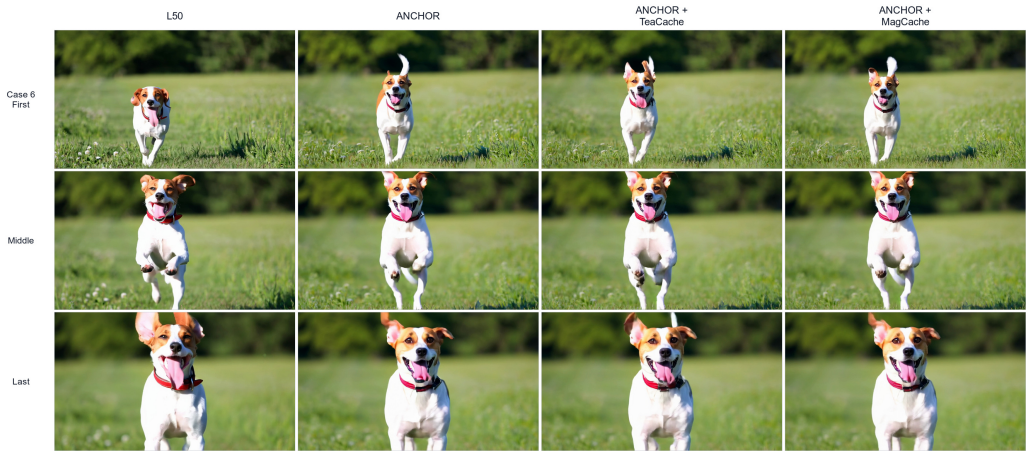


Figure 14: **Composition with caching.** Prompt: "A dog running happily." Rows show the first, middle, and last frames.



Figure 15: **Composition with caching.** Prompt: "A bicycle gliding through a snowy field." Rows show the first, middle, and last frames.