

Heuristics are Not Enough: Calibrated and Corrected Caching for Video Diffusion Acceleration

Jiyue Jiang^{♡*}, Zhiyu Huo^{♡*}, Yuhao Chen[♡], Yanruisheng Shao[♡], Gong Zhang[♣], Lei Deng^{♣†}

[♡] Department of Computer Science and Engineering, The Chinese University of Hong Kong

[♣] Theory Lab, 2012 Labs, Huawei Technology Co. Ltd

{jiangjy, zhiyuhuo, yuhanchen, 1155231342}@link.cuhk.edu.hk

{nicholas.zhang, deng.lei2}@huawei.com

Abstract

Feature caching has become a mainstream training-free strategy for accelerating video diffusion models by reusing intermediate outputs across denoising steps. However, existing adaptive methods rely on heuristic difference measures misaligned with actual generation quality, employ a single global threshold that ignores heterogeneous phase dynamics, and never validate their budget estimates against true approximation error, allowing systematic bias to propagate uncorrected. We propose **AlignCache**, a closed-loop caching framework that directly aligns scheduling decisions with their true influence on the final generated video through three tightly coupled components: (1) phase-adaptive budget routing that partitions the denoising trajectory into structurally distinct phases, each governed by an independent threshold and a staleness-aware guard; (2) closed-loop threshold calibration that exploits counterfactual error measurement at the classifier-free guided output level to adjust decision boundaries online via an EMA feedback loop; and (3) momentum-augmented draft synthesis combining direction-magnitude gated residual extrapolation with anchor-guided unconditional correction. By transforming scheduling from a static risk accumulator into a feedback-driven, self-correcting controller, AlignCache consistently achieves superior acceleration-quality trade-offs across state-of-the-art video generation models. Our code is available at <https://github.com/jiangjy/AlignCache>.

1 Introduction

Diffusion models [7, 20, 21] have become the dominant paradigm for high-quality generative modeling, with Diffusion Transformers (DiTs) [20] powering state-of-the-art video generation systems such as Open-Sora [34], HunyuanVideo [13], and Wan2.1 [25]. Despite their impressive generation quality, the iterative denoising nature of these models incurs substantial computational costs, dozens of sequential forward passes through billion-parameter networks, making efficient inference a critical bottleneck for deployment and motivating a growing body of training-free acceleration research.

Among various training-free strategies, feature caching [14, 18, 33] has emerged as an effective paradigm by exploiting temporal redundancy across denoising steps. Static caching methods [23, 33] adopt fixed intervals but cannot adapt to timestep dynamics, while adaptive approaches [11, 14, 18, 28, 35] use runtime metrics to decide whether to skip or compute. However, existing adaptive methods exhibit three structural limitations: (i) they rely on heuristic difference measures that do not reflect the impact of caching on the generated video; (ii) they employ a single global threshold blind

*These authors are co-first authors.

†Corresponding author

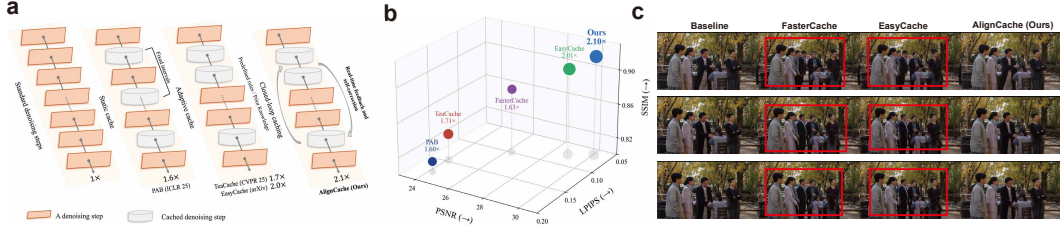


Figure 1: **Overview and motivation of AlignCache.** (a) Existing schedulers rely on predefined rules and prior knowledge to allocate exact and cached steps, while AlignCache leverages real-time feedback and self-correction. (b) AlignCache dominates other methods on PSNR, LPIPS, and speedup. (c) Comparison on “The bund Shanghai, pan left.”: AlignCache preserves fine details where competing methods distort (red boxes). Videos in (c) are generated by HunyuanVideo.

to heterogeneous phase dynamics; and (iii) their budget estimates are never validated against actual approximation error, allowing systematic bias to propagate uncorrected during sampling.

In this work, we propose **AlignCache**, a closed-loop caching framework that directly aligns scheduling decisions with their actual influence on the final generated video. As illustrated in Figure 1, in contrast to prior schedulers that rely on predefined rules and prior knowledge to dispatch exact and cached denoising steps, AlignCache leverages real-time feedback and self-correction to achieve a superior speed-quality trade-off while preserving fine-grained visual details that competing methods tend to distort. AlignCache introduces three tightly coupled components: (1) a *phase-adaptive budget routing* mechanism that partitions the denoising trajectory into structurally distinct phases (early, mid, late), each governed by an independent threshold and a staleness-aware guard; (2) a *closed-loop threshold calibration* module that exploits counterfactual observations at each budget-triggered exact computation to measure the true approximation error at the classifier-free guided output level, and dynamically adjusts the decision boundary via an exponential moving average feedback loop; and (3) a *momentum-augmented draft synthesis* strategy that employs scaled residual extrapolation with anchor-guided unconditional correction, improving skip-mode fidelity and enabling more aggressive caching without sacrificing quality. By transforming the scheduling paradigm from a static, open-loop risk accumulator into a feedback-driven, self-correcting controller, AlignCache achieves superior acceleration-quality trade-offs across diverse video generation models.

Our main contributions are: (1) We identify the misalignment between heuristic caching criteria and generation quality, and propose AlignCache, a closed-loop framework bridging this gap via counterfactual error measurement. (2) We design a phase-adaptive budget routing mechanism with staleness-aware guards and closed-loop threshold calibration, replacing static thresholds with a self-correcting controller. (3) We introduce a momentum-augmented draft synthesis combining residual extrapolation with anchor-guided correction. (4) Experiments on multiple video generation models demonstrate that AlignCache consistently achieves superior acceleration-quality trade-offs.

2 Related Works

2.1 Video Diffusion Models

Diffusion models [7, 20, 21] have established themselves as the dominant paradigm for generative modeling. Early video generation efforts built upon U-Net architectures [22] and latent diffusion frameworks [1, 2, 9] to synthesize temporally coherent content. More recently, the community has shifted towards Diffusion Transformers (DiTs) [20] for their superior scalability in modeling complex spatiotemporal dynamics. State-of-the-art DiT-based systems including Sora [4], OpenSora [34], CogVideoX [30], HunyuanVideo [13], and Wan2.1 [25] have achieved remarkable fidelity and coherence, largely benefiting from Rectified Flow [16] formulations that enable high-quality generation with fewer sampling steps. Despite this progress, the iterative denoising nature of these models still incurs substantial computational costs, making efficient inference a pressing challenge for practical deployment.

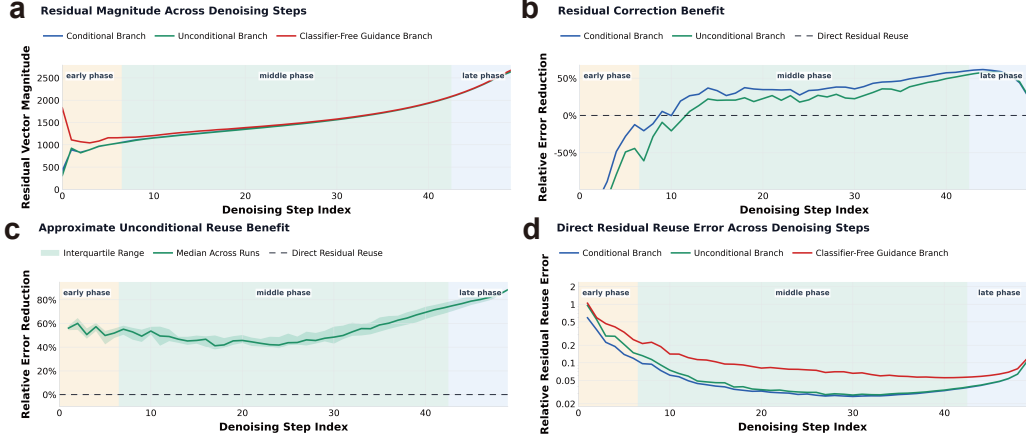


Figure 2: **Empirical analysis of residual dynamics across denoising steps on Wan2.1-1.3B.** (a) Residual magnitude exhibits distinct early, middle, and late phases. (b) Residual correction yields large error reduction over direct reuse in mid-to-late phases. (c) Approximate unconditional reuse provides consistent benefit across the trajectory. (d) Direct residual reuse error grows along the trajectory and differs across branches.

2.2 Feature Caching for Diffusion Acceleration

To accelerate diffusion inference without retraining, various training-free strategies have been explored, including ODE solvers [12, 17, 24], token merging [3, 32, 36], and sparsity-based approaches [26, 29, 31]. Among these, feature caching has emerged as an effective paradigm for video generation by exploiting temporal redundancy across denoising steps. Static caching methods [5, 23, 33] use fixed intervals and cannot adapt to timestep importance, while adaptive caching approaches [6, 11, 14, 18, 28] introduce runtime metrics for dynamic decisions yet rely on heuristic difference measures that do not reflect the impact of caching on generation quality, leading to either conservative caching with limited speedup or aggressive caching with noticeable quality degradation. Recent works [15, 27] address this misalignment through frequency decomposition or higher-order feature forecasting, yet the fundamental gap between caching criteria and perceptual quality remains under-explored. In contrast, AlignCache directly aligns the caching decision criterion with the influence on the generated video via closed-loop counterfactual error measurement, enabling more precise redundancy identification and superior acceleration-quality trade-offs.

3 Preliminaries

Empirical Analysis of Residual Dynamics. To motivate AlignCache, we analyze how output residuals evolve along the denoising trajectory of Wan2.1-1.3B (Figure 2). Three observations emerge: (i) residual magnitude exhibits markedly heterogeneous dynamics across the early, middle, and late phases, making a single global threshold structurally insufficient; (ii) properly designed correction reduces relative error by 30-50% over direct reuse, with the largest gap precisely where direct reuse becomes most error-prone; and (iii) direct reuse error varies systematically across conditional, unconditional, and CFG-guided branches, indicating that scheduling decisions made on a single branch cannot reliably predict the distortion injected into the final output. These observations directly motivate our phase-adaptive routing (§4.1), momentum-augmented draft synthesis (§4.3), and closed-loop calibration at the CFG-guided level (§4.2), respectively.

Cache-Based Acceleration for Guided Diffusion. In classifier-free guidance (CFG) [8], each denoising step evaluates the network ϵ_ϕ twice, once on the text prompt $\mathbf{c}$ and once on the unconditional input $\mathbf{c}_\emptyset$, forming the guided prediction $\tilde{\epsilon}_t = \epsilon_\phi(\mathbf{x}_t, t, \mathbf{c}_\emptyset) + \gamma(\epsilon_\phi(\mathbf{x}_t, t, \mathbf{c}) - \epsilon_\phi(\mathbf{x}_t, t, \mathbf{c}_\emptyset))$ with guidance scale γ . We index timesteps by t , with conditional output $\mathbf{c}_t$ and unconditional output $\mathbf{u}_t$. Cache-based methods [19, 28, 35] exploit the temporal smoothness of $\{\mathbf{c}_t, \mathbf{u}_t\}$ by caching the output residual $\mathbf{r}_t^{(b)} = \mathbf{y}_t^{(b)} - \mathbf{x}_t^{(b)}$ for each branch $b \in \{\text{cond}, \text{uncond}\}$ at the most recent *exact anchor* $a(t)$ and reusing it via $\hat{\mathbf{y}}_t^{(b)} = \mathbf{x}_t^{(b)} + \mathbf{r}_{a(t)}^{(b)}$. A *scheduler* decides at every timestep whether to execute a

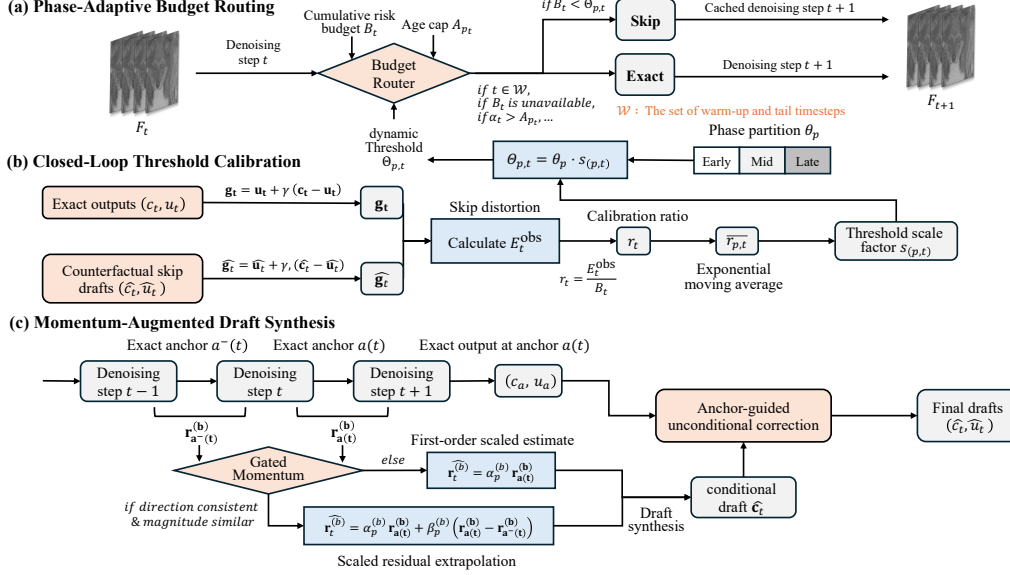


Figure 3: **Overview of AlignCache.** (a) **Pipeline of phase-adaptive budget routing.** The system decides whether to skip or perform exact computation based on the cumulative risk budget B_t , age cap $A_{p,t}$, and dynamic threshold $\theta_{p,t}$. F_t is the input at timestep t , α_t the anchor age, and $\mathcal{W}$ denotes warm-up and tail timesteps unconditionally computed exactly. (b) **Workflow of closed-loop threshold calibration.** The process computes exact outputs and counterfactual skip drafts, then derives the skip distortion. The calibration ratio is updated from observed distortion and combined with the dynamic threshold. The resulting scale factor and phase partition θ_p determine $\theta_{p,t}$ in part (a). (c) **Process of momentum-augmented draft synthesis.** The current residual is predicted from the two most recent exact anchors $a(t)$ and $a^-(t)$, applying momentum correction when direction and magnitude align, or a first-order estimate otherwise. The final drafts $(\hat{c}_t, \hat{u}_t)$ feed into part (b).

full forward pass (*exact*) or apply the cached approximation (*skip*); the quality-efficiency trade-off is governed entirely by this binary routing policy.

Open-Loop Budget Scheduling and Its Limitations. Existing schedulers [35] frame routing as a cumulative risk test: a scalar budget B_t accumulates a heuristic risk increment per timestep, and a skip is taken whenever $B_t < \tau$ for a fixed threshold τ (full update rule in Appendix A.2). This open-loop design has three structural limitations: (i) a global τ cannot accommodate heterogeneous dynamics across early, mid, and late phases; (ii) the absence of a staleness constraint permits unbounded skip runs when inputs change slowly; and (iii) the budget is never validated against approximation error, so systematic bias propagates uncorrected. AlignCache addresses these limitations through phase-adaptive routing, staleness-aware guarding, and closed-loop threshold calibration (§4).

4 Methodology: AlignCache

We consider CFG sampling over T denoising timesteps. At each timestep t , the model evaluates conditional and unconditional forward passes, with conditional input $\mathbf{x}_t$, exact conditional output $\mathbf{y}_t$, and CFG-guided output $\mathbf{g}_t$. AlignCache decides at each t whether to execute a full forward pass (*exact*) or reuse cached computations (*skip*). As illustrated in Figure 3, our framework comprises three coupled components: (i) a *phase-adaptive budget routing* mechanism that assigns phase-specific thresholds and staleness constraints to segments of the denoising trajectory (§4.1); (ii) a *closed-loop threshold calibration* module that leverages counterfactual observations from exact computations to adjust decision boundaries (§4.2); and (iii) a *momentum-augmented draft synthesis* strategy that improves skip-mode fidelity, enabling more aggressive caching with quality guarantees (§4.3). Unlike prior open-loop schedulers comparing a cumulative budget against a fixed threshold τ , AlignCache replaces this static decision boundary with a dynamic, phase-aware, and self-correcting threshold $\Theta_{p,t}$, transforming scheduling from a risk accumulator into a feedback-driven closed-loop controller.

4.1 Phase-Adaptive Budget Routing

We partition the denoising trajectory into three structurally distinct phases, *early*, *mid*, and *late*, based on transition ratios (ρ_e, ρ_l) over the normalized progress $\rho_t = t / \max(T-1, 1)$, with each phase p governed by an independent base threshold θ_p that permits conservative policies in structurally sensitive phases and aggressive ones in stable phases. We maintain a cumulative risk budget $B_t = B_{t-1} + \Delta B_t$ that accrues a normalized input perturbation amplified by an adaptive motion gain $g_{a(t)}$ since the last exact computation, and resets upon each exact step (full update rules including the motion gain in Appendix A.3). To prevent unbounded skip runs when inputs change slowly, we further introduce a staleness guard: with anchor age $\alpha_t = t - t_{\text{last}}$ and per-phase cap A_p , exact computation is forced whenever $\alpha_t > A_{p_t}$. The unified routing rule is:

$$\text{Route}(t) = \begin{cases} \text{exact}, & \text{if } t \in \mathcal{W} \text{ or } B_t \text{ unavailable or } \alpha_t > A_{p_t}, \\ \text{skip}, & \text{if } B_t < \Theta_{p,t}, \\ \text{exact}, & \text{otherwise,} \end{cases} \quad (1)$$

where $\mathcal{W}$ is the set of warm-up and tail timesteps that are unconditionally computed exactly, and the *dynamic threshold* is $\Theta_{p,t} = \theta_p \cdot s_{p,t}$. The base θ_p encodes phase-specific conservativeness, while the online scale factor $s_{p,t}$ is maintained by the closed-loop calibration module described next, enabling the scheduler to self-adjust according to the evolving fidelity of its own predictions.

4.2 Closed-Loop Threshold Calibration

A limitation of open-loop schedulers is that the budget B_t may systematically deviate from the true approximation error, causing excessive skipping or unnecessary exact computations. We close this gap by exploiting each budget-triggered exact computation as a calibration opportunity.

Counterfactual Error Measurement. When the scheduler triggers an exact step because $B_t \geq \Theta_{p_t,t}$, we first construct counterfactual skip drafts $(\hat{\mathbf{c}}_t, \hat{\mathbf{u}}_t)$ via the draft synthesis module (§4.3) *before* the true forward pass. After obtaining the exact outputs $(\mathbf{c}_t, \mathbf{u}_t)$, we form their guided compositions $\mathbf{g}_t = \mathbf{u}_t + \gamma(\mathbf{c}_t - \mathbf{u}_t)$ and $\hat{\mathbf{g}}_t = \hat{\mathbf{u}}_t + \gamma(\hat{\mathbf{c}}_t - \hat{\mathbf{u}}_t)$, and measure the observed error at the guided level via relative MSE:

$$E_t^{\text{obs}} = \frac{\|\hat{\mathbf{g}}_t - \mathbf{g}_t\|_2^2}{\|\mathbf{g}_t\|_2^2}. \quad (2)$$

This directly quantifies the distortion the skip approximation would have injected into the final sample, which is more informative than per-branch comparison since $\mathbf{g}_t$ is the quantity that ultimately determines generation quality.

Online Scale Update with Selective Feedback. We define the calibration ratio $r_t = E_t^{\text{obs}} / B_t$, where $r_t > 1$ indicates that the budget underestimates the true error and $r_t < 1$ signals overestimation. We maintain a clipped per-phase exponential moving average $\bar{r}_{p,t}$ and compute the threshold scale factor $s_{p,t}$ via an inverse-clipping rule (full update equations in Appendix A.4). The inverse relationship ensures a self-correcting feedback loop: when $\bar{r}_{p,t} > 1$, $s_{p,t}$ decreases, tightening $\Theta_{p,t}$ and triggering more frequent exact computations; when $\bar{r}_{p,t} < 1$, the threshold relaxes for additional skips. To avoid contaminating the calibration signal with structural scheduling events, we restrict feedback updates to exact computations triggered *solely* by the budget criterion ($B_t \geq \Theta_{p,t}$), excluding warm-up, bootstrap, and staleness-triggered events.

4.3 Momentum-Augmented Draft Synthesis

The fidelity of skip-mode outputs affects both generation quality and the reliability of the closed-loop calibration signal. We design a draft synthesis module that goes beyond naive residual reuse via direction-magnitude gated extrapolation and anchor-guided correction.

Scaled Residual Extrapolation with Gated Momentum. For each branch $b \in \{\text{cond}, \text{uncond}\}$, given the residuals at the two most recent exact anchors $a(t)$ and $a^-(t)$, we predict the current residual via a first-order scaled estimate augmented with a gated second-order momentum correction:

$$\hat{\mathbf{r}}_t^{(b)} = \alpha_p^{(b)} \mathbf{r}_{a(t)}^{(b)} + \beta_p^{(b)} \left(\mathbf{r}_{a(t)}^{(b)} - \mathbf{r}_{a^-(t)}^{(b)} \right), \quad (3)$$

Table 1: Comparison between other acceleration strategies and our AlignCache. We report inference latency, speedup, visual retention metrics (PSNR, SSIM, LPIPS, and GMSD), and VBench Total score on representative video generation models. The baseline is reported with the original T denoising steps. “+” denotes applying the corresponding acceleration strategy on top of the baseline.

Methods	Efficiency		Visual Retention				VBench
	Latency (s) ↓	Speedup ↑	PSNR ↑	SSIM ↑	LPIPS ↓	GMSD ↓	Total (%) ↑
<i>Wan2.1-1.3B [25] (81 frames, 832×480)</i>							
Wan2.1-1.3B ($T=50$)	131.9	1.00×	—	—	—	—	81.64
+ PAB [33]	74.2	1.78×	21.42	0.780	0.162	0.164	81.00
+ TeaCache [14]	86.2	1.53×	27.57	0.903	0.055	0.100	81.23
+ FasterCache [18]	75.9	1.74×	26.35	0.891	0.063	0.109	81.41
+ EasyCache [35]	61.1	2.19×	28.38	0.903	0.051	0.089	81.04
+ AlignCache (Ours)	57.5	2.29×	29.80	0.925	0.038	0.080	81.28
<i>Open-Sora-1.2 [34] (102 frames, 848×480)</i>							
Open-Sora ($T=30$)	70.2	1.00×	—	—	—	—	80.07
+ PAB [33]	48.3	1.45×	18.76	0.716	0.319	0.188	75.62
+ TeaCache [14]	47.2	1.49×	22.10	0.802	0.179	0.156	79.61
+ FasterCache [18]	45.5	1.54×	25.29	0.861	0.115	0.124	80.08
+ EasyCache [35]	41.1	1.75×	25.87	0.876	0.112	0.114	79.58
+ AlignCache (Ours)	38.1	1.86×	26.06	0.878	0.106	0.114	79.59
<i>HunyuanVideo [13] (65 frames, 864×480)</i>							
HunyuanVideo ($T=50$)	208.7	1.00×	—	—	—	—	82.55
+ PAB [33]	133.4	1.56×	32.36	0.939	0.050	0.061	82.17
+ TeaCache [14]	98.7	2.11×	22.52	0.788	0.206	0.150	82.42
+ FasterCache [18]	130.1	1.60×	27.13	0.874	0.119	0.109	82.64
+ EasyCache [35]	101.9	2.09×	32.33	0.923	0.063	0.059	82.08
+ AlignCache (Ours)	98.1	2.15×	34.23	0.945	0.046	0.049	82.16

where $\alpha_p^{(b)}$ is a phase-dependent scaling coefficient and $\beta_p^{(b)}$ is an adaptive momentum weight modulated by a direction-magnitude gating function (full form in Appendix A.5). The direction gate suppresses extrapolation when consecutive anchor residuals diverge in orientation, while the magnitude gate attenuates the correction when residual norms change rapidly, gracefully decaying $\beta_p^{(b)}$ toward zero during transient dynamics. Compared with static residual reuse ($\hat{\mathbf{y}}_t^{(b)} = \mathbf{x}_t^{(b)} + \mathbf{r}_{a(t)}^{(b)}$), this extrapolation captures the evolving trend of residuals and reduces drift during extended skip runs.

Anchor-Guided Unconditional Correction. We further refine the unconditional draft by exploiting the structural correlation between conditional and unconditional branches at the most recent anchor. Let $(\mathbf{c}_a, \mathbf{u}_a)$ denote the exact outputs at anchor $a(t)$ and $\hat{\mathbf{c}}_t$ the current conditional draft. The final unconditional draft blends the extrapolation baseline $\hat{\mathbf{u}}_t^{\text{base}}$ with an anchor-transferred estimate $\tilde{\mathbf{u}}_t = \mathbf{u}_a + \alpha_p(\hat{\mathbf{c}}_t - \mathbf{c}_a)$ via a blending weight ω (Appendix A.5). This leverages the cond-uncond coupling inherent in CFG to produce more faithful unconditional drafts, yielding more accurate guided approximations during both skip execution and counterfactual error measurement.

5 Experiments and Results

5.1 Setup

Baselines. We compare AlignCache against state-of-the-art training-free cache-based accelerators, including PAB [33], TeaCache [14], FasterCache [18], and EasyCache [35], using their official implementations whenever available, and faithfully re-implementing them on backbones not directly supported by the original code.

Models, Metrics, and Datasets. We evaluate on three DiT-based video generators: Wan2.1-1.3B [25] (81 frames, 832×480, $T=50$), Open-Sora-1.2 [34] (102 frames, 848×480, $T=30$), and HunyuanVideo [13] (65 frames, 864×480, $T=50$). For each model, we measure end-to-end latency and

speedup, reference-fidelity metrics against the unaccelerated output (PSNR, SSIM, LPIPS, GMSD), and the VBench [10] Total score on a subset of official prompts with 5 or 10 samples per prompt.

Implementation Details. Hyperparameters are tuned by sweeping 5 prompts $\times$ 2 seeds. For Wan2.1-1.3B, the base threshold is 0.065 with 12 warm-up exact CFG pairs and phase scales (1.15, 1.00, 0.80) for early/mid/late. For Open-Sora-1.2 and HunyuanVideo, the base thresholds are 0.11 and 0.024, with 8 and 10 warm-up pairs, respectively. Phase boundaries are (15%, 86%) for Wan2.1-1.3B/Open-Sora-1.2 and (18%, 72%) for HunyuanVideo. Main results are obtained on NVIDIA RTX PRO 6000-96GB; ablations on Wan2.1-1.3B use 5 prompts $\times$ 2 seeds (42, 954) on a single NVIDIA H100-80GB.

5.2 Main Results

Table 1 compares AlignCache with prior cache-based accelerators across the three backbones. AlignCache simultaneously attains the *highest speedup* and the *best fidelity* on every model, dominating the Pareto frontier without exception.

Pareto dominance across backbones. On Wan2.1-1.3B, AlignCache reaches $2.29\times$ speedup with 29.80dB PSNR and 0.038 LPIPS, improving over the strongest baseline EasyCache by +1.42dB PSNR and -25% LPIPS while being faster ($2.29\times$ vs. $2.19\times$). On HunyuanVideo, the margin widens to +1.90dB PSNR and -22% LPIPS at a higher $2.15\times$ speedup. On Open-Sora-1.2, AlignCache delivers the largest speedup ($1.86\times$) among all methods while still leading on PSNR, SSIM, and LPIPS. The consistency of this gain across three architectures with different sampling lengths and aspect ratios indicates that the improvement stems from a model-agnostic property of the scheduler rather than backbone-specific tuning.

Source of the joint speed-quality improvement.

A natural concern is whether AlignCache simply tightens the threshold to trade speed for quality. The data argue otherwise: on Wan2.1-1.3B, AlignCache simultaneously *reduces* latency relative to EasyCache (57.5 s vs. 61.1 s) and improves PSNR by over a full dB. Such co-improvement is enabled by two complementary mechanisms acting in concert. The momentum-augmented draft synthesis produces higher-fidelity skip outputs by capturing residual curvature beyond static reuse, directly reducing approximation error along extended skip runs and accounting for a substantial portion of the PSNR gain. The closed-loop calibration further redirects the scheduler toward exact computation when the calibration ratio $\tilde{r}_{p,t}$ exceeds 1, ensuring that residual budget is spent on the most informative timesteps. Inspecting the per-step routing logs, we observe that AlignCache skips aggressively in the middle phase where residuals are smooth and reverts to exact computation in the late phase when feedback signals indicate growing distortion, behavior that an open-loop scheduler with a single global τ structurally cannot reproduce.

VBench scores reflect a coarser axis than visual fidelity. VBench Total is comparable across methods because reference-free aggregate scores conflate true generation quality with prompt-induced variation; the more discriminative signal lies in PSNR/LPIPS against the unaccelerated baseline, where AlignCache consistently leads by a clear margin. AlignCache attains 0.8128/0.7959/0.8216 on the three backbones, on par with the best baseline on each, indicating that the closed-loop calibration recovers the original generation distribution rather than over-smoothing it into a more “VBench-friendly” but distorted output.

Table 2: Ablations on **scheduler-related** axes on Wan2.1-1.3B. Representative configurations shown; full grid in App. A.6.

Variant	Lat.(s)↓	Spd.↑	PSNR↑	SSIM↑	LPIPS↓
(a) Warm-up steps W					
$W=5$	42.73	$2.29\times$	23.68	0.829	0.123
$W=7$	43.12	$2.26\times$	24.40	0.850	0.110
$W=9$	43.72	$2.22\times$	24.51	0.862	0.104
$W=12$	46.82	$2.06\times$	25.97	0.882	0.086
$W=14$	49.03	$1.97\times$	26.34	0.887	0.078
(b) Error threshold θ_p					
0.0325	53.77	$1.81\times$	27.20	0.904	0.070
0.0400	50.42	$1.91\times$	25.48	0.870	0.090
0.0425	47.21	$2.06\times$	25.58	0.880	0.088
0.0600	41.75	$2.31\times$	24.05	0.843	0.114
0.0700	39.35	$2.45\times$	23.65	0.832	0.123
(c) Phase ratio (ρ_e, ρ_l)					
(0.125, 0.88)	44.29	$2.19\times$	25.11	0.873	0.095
(0.150, 0.85)	45.12	$2.14\times$	24.61	0.859	0.102
(0.150, 0.80)	45.15	$2.14\times$	24.61	0.859	0.102
(0.175, 0.88)	44.28	$2.19\times$	25.05	0.873	0.096
(0.200, 0.86)	45.14	$2.14\times$	24.65	0.859	0.100

5.3 Ablation Studies

To understand the contribution of each design choice in AlignCache, we conduct controlled ablations along five orthogonal axes on *Wan2.1-1.3B* [25], organized into two conceptual layers: *scheduler-related* axes (warm-up steps, error threshold, phase ratio) in Table 2 and *predictor-related* axes (motion gain, residual extrapolation order) in Table 3. All variants use 5 prompts $\times$ 2 random seeds against an unaccelerated baseline whose mean latency is 96.54 s on a single NVIDIA H100-80GB, and we report end-to-end latency, sampling time speedup, and reference-fidelity (PSNR/SSIM/LPIPS) measured against the baseline output.

Effect of warm-up steps. Subgroup (a) of Table 2 shows that increasing the warm-up budget from $W=5$ to $W=14$ raises PSNR by +2.66 dB and reduces LPIPS by 37%, while the speedup decreases monotonically from $2.29\times$ to $1.97\times$. This trade-off arises because warm-up serves as a statistical prerequisite for closed-loop calibration: with too few warm-up steps, the EMA in Eq. (9) (Appendix A.4) lacks sufficient samples to stabilize, and the resulting noisy feedback degrades routing decisions, which is why $W=5$ achieves a similar speedup to $W=9$ but 0.83 dB lower PSNR. Beyond a sufficient warm-up budget, additional exact steps forfeit skip opportunities without commensurate quality gain, indicating that warm-up duration governs a calibration-stability regime that grows with W .

Effect of error base threshold. Subgroup (b) shows that each 0.0025 increment in θ_p buys roughly $+0.05\times$ to $+0.10\times$ extra speedup with a graceful degradation in fidelity across the swept range. The fidelity drop accelerates at higher thresholds when skip runs grow long enough that the residual predictor alone cannot fully compensate, exposing a tight coupling between threshold and predictor: raising θ_p without strengthening the predictor amplifies extrapolation error super-linearly along skip runs, so the achievable quality-speed trade-off is shaped jointly by these two layers rather than by the threshold alone.

Effect of phase ratio. Subgroup (c) indicates that the phase ratio is the *least sensitive* axis: PSNR varies by less than 0.55 dB and SSIM by less than 0.015 across the full 11-point grid. This insensitivity is itself evidence that the closed-loop design is doing what it claims: the calibrated $\Theta_{p,t}$ absorbs small misspecifications in phase boundaries through online adjustment of $s_{p,t}$, shifting the dominant performance bottleneck to threshold and motion-gain settings rather than partitioning. The flatness of the grid further confirms that the inductive split between transient and converged dynamics is qualitative rather than knife-edge, so phase boundaries can be treated as a structural prior rather than a tightly tuned hyperparameter.

Effect of motion gain. Subgroup (a) of Table 3 reveals that motion gain is the strongest *quality-recovery* knob: increasing $\beta_0^{(p)}$ from 0.50 to 0.85 lifts PSNR by +11.1 dB and reduces LPIPS by 83%, while $\beta_0^{(p)}=0.90$ collapses the speedup to $1.00\times$ as the gain becomes so conservative that the controller never skips. The order-of-magnitude swing in PSNR with a small change in $\beta_0^{(p)}$ indicates that skip-mode fidelity is governed by the curvature of the residual trajectory rather than its raw magnitude, and the second-order momentum term, when correctly gated, recovers most of the lost signal that anchor reuse cannot reconstruct. The threshold scale s provides finer control: holding $\beta_0^{(p)}=0.80$ and varying s from 0.75 to 1.25 trades +5.1 dB PSNR for $-0.20\times$ speedup, giving practitioners a smooth quality-efficiency dial that can be adjusted along the trade-off curve.

Effect of residual extrapolation. Subgroup (b) shows that disabling residual extrapolation entirely costs 0.10 dB in PSNR and 16% in LPIPS for only $+0.09\times$ extra speedup, and the first-order-only variant likewise underperforms the full predictor. The second-order momentum correction in Eq. (3) therefore plays a dual role: it captures the curvature of the residual trajectory during extended skip runs to improve skip-mode fidelity, and it sharpens the calibration signal E_t^{obs} , since a more accurate counterfactual draft yields a less biased estimate of skip distortion. Removing it therefore degrades

Table 3: Ablations on **predictor-related** axes on Wan2.1-1.3B. Subgroups (a)-(b) vary motion-gain parameters and residual extrapolation order, respectively.

Variant	Lat.(s)↓	Spd.↑	PSNR↑	SSIM↑	LPIPS↓
(a) Motion gain ($\beta_0^{(p)}, s$)					
(0.50, 1.00)	49.76	$1.94\times$	25.58	0.873	0.089
(0.75, 1.00)	62.23	$1.56\times$	30.21	0.937	0.042
(0.80, 0.75)	63.73	$1.51\times$	29.23	0.930	0.045
(0.80, 0.95)	67.25	$1.44\times$	32.70	0.954	0.030
(0.80, 1.00)	68.23	$1.41\times$	32.22	0.954	0.029
(0.80, 1.05)	69.38	$1.39\times$	33.69	0.959	0.026
(0.80, 1.15)	71.15	$1.36\times$	34.35	0.963	0.022
(0.80, 1.25)	73.37	$1.31\times$	33.49	0.961	0.024
(0.85, 1.00)	79.70	$1.21\times$	36.69	0.973	0.015
(0.90, 1.00)	96.42	$1.00\times$	—	1.000	0.000
(b) Residual extrapolation order					
None	43.45	$2.23\times$	24.51	0.843	0.118
1st-order only	44.13	$2.19\times$	24.04	0.841	0.117
1st+2nd	45.12	$2.14\times$	24.61	0.859	0.102

both generation quality and downstream calibration accuracy, compounding the loss across both layers of the framework.

Cross-platform efficiency. The $2.29\times$ speedup on Wan2.1-1.3B transfers consistently across NVIDIA Hopper and Blackwell architectures, reaching 42.4 s end-to-end latency on H100-80GB and 57.5 s on RTX PRO 6000-96GB, both at the same $2.29\times$ speedup ratio over the unaccelerated baseline. The identical speedup factor across two distinct architectures demonstrates that the acceleration is computation-bound rather than memory-bound and that the scheduling overhead is negligible.

5.4 Case Studies

Figure 4 presents qualitative comparisons on representative VBench prompts spanning diverse motion patterns. On “A shark is swimming in the ocean, pan right.” (Wan2.1-1.3B), FasterCache and EasyCache visibly distort the shark’s silhouette and fin geometry under their respective caching policies, while AlignCache preserves the object boundary and motion direction at a comparable or higher speedup without temporal flicker. On “A person is shining shoes.” (Open-Sora-1.2), competing methods introduce blurring artifacts and texture inconsistencies on the shoe surface and hand region (red boxes), whereas AlignCache faithfully retains fine leather textures and high-frequency edges throughout the entire sequence. The qualitative gap concentrates on semantically salient regions, object boundaries, faces, and high-frequency textures, precisely where direct residual reuse incurs the largest approximation error in the late denoising phase. Through E_t^{obs} , the closed-loop calibrator detects these high-error events and adaptively forces exact computations, preventing the open-loop failure mode of accumulating distortion past the perceptual threshold and explaining why the LPIPS gap exceeds the PSNR gap on average across prompts.



Figure 4: **Qualitative comparison** on “A shark is swimming in the ocean, pan right.” (left, Wan2.1-1.3B) and “A person is shining shoes.” (right, Open-Sora-1.2) under matched or higher speedup settings. Red boxes highlight regions where competing methods introduce noticeable distortions, while AlignCache faithfully preserves object geometry and fine details at a comparable or higher speedup. More test method case studies can be found in the Appendix A.7.

6 Conclusion and Future Works

Conclusion. We presented **AlignCache**, a closed-loop caching framework that addresses three structural limitations of existing adaptive schedulers: heuristic criteria misaligned with generation quality, single global thresholds insensitive to phase dynamics, and uncorrected systematic bias in budget estimation. By coupling phase-adaptive budget routing, counterfactual error measurement at the CFG-guided output level, and momentum-augmented draft synthesis, AlignCache transforms the scheduling paradigm from a static risk accumulator into a feedback-driven, self-correcting controller. Experiments on Wan2.1-1.3B, Open-Sora-1.2, and HunyuanVideo show that AlignCache consistently achieves the best speed-quality Pareto frontier, simultaneously delivering higher speedup and better fidelity than prior cache-based accelerators.

Future Works. Several directions remain open. First, the closed-loop principle behind AlignCache is not specific to caching, the same counterfactual error measurement could calibrate other training-free accelerators such as token pruning or sparse attention. Second, current calibration treats each prompt independently; learning a prompt-conditioned prior over $\bar{r}_{p,t}$ could further reduce warm-up overhead and unlock higher speedups on short trajectories. Third, combining AlignCache with orthogonal

acceleration techniques such as quantization, distillation, or compiler-level kernel fusion could yield compounded speedups while the closed-loop controller absorbs the additional approximation error introduced by these layers and preserves stable quality across heterogeneous deployment hardware. Finally, extending AlignCache to long-form video generation and multi-modal diffusion pipelines, where redundancy patterns differ substantially from text-to-video, is a natural next step toward a unified, self-correcting acceleration framework for generative models at scale.

References

- [1] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.
- [2] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22563–22575, 2023.
- [3] Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4599–4603, 2023.
- [4] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Leo Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.
- [5] 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*, 2024.
- [6] Huanpeng Chu, Wei Wu, Guanyu Feng, and Yutao Zhang. Omnicache: A trajectory-oriented global perspective on training-free cache reuse for diffusion transformer models. pages 16302–16312, 2025.
- [7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [8] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- [9] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.
- [10] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024.
- [11] 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. pages 15240–15252, 2025.
- [12] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.
- [13] 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.
- [14] 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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7353–7363, 2025.

- [15] Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Junjie Chen, and Linfeng Zhang. From reusing to forecasting: Accelerating diffusion models with taylorseers. pages 15853–15863, 2025.
- [16] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.
- [17] 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. *Advances in neural information processing systems*, 35:5775–5787, 2022.
- [18] 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. *arXiv preprint arXiv:2410.19355*, 2024.
- [19] 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*, pages 15762–15772, 2024.
- [20] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.
- [21] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [23] 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.
- [24] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [25] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Fei Wu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.
- [26] Hongjie Wang, Difan Liu, Yan Kang, Yijun Li, Zhe Lin, Niraj K Jha, and Yuchen Liu. Attention-driven training-free efficiency enhancement of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16080–16089, 2024.
- [27] Jiangshan Wang, Kang Zhao, Jiayi Guo, Jiayu Wang, Hang Guo, Chenyang Zhu, Xiu Li, and Xiangyu Yue. Precise-cache: Precise feature caching for efficient and high-fidelity video generation. *arXiv preprint arXiv:2603.00976*, 2026.
- [28] Felix Wimbauer, Bichen Wu, Edgar Schoenfeld, Xiaoliang Dai, Ji Hou, Zijian He, Artsiom Sanakoyeu, Peizhao Zhang, Sam Tsai, Jonas Kohler, et al. Cache me if you can: Accelerating diffusion models through block caching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6211–6220, 2024.
- [29] Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, et al. Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity. *arXiv preprint arXiv:2502.01776*, 2025.
- [30] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.
- [31] Hancheng Ye, Jiakang Yuan, Renqiu Xia, Xiangchao Yan, Tao Chen, Junchi Yan, Botian Shi, and Bo Zhang. Training-free adaptive diffusion with bounded difference approximation strategy. volume 37, pages 306–332, 2024.

- [32] Evelyn Zhang, Jiayi Tang, Xuefei Ning, and Linfeng Zhang. Training-free and hardware-friendly acceleration for diffusion models via similarity-based token pruning. 39(9):9878–9886, 2025.
- [33] Xuanlei Zhao, Xiaolong Jin, Kai Wang, and Yang You. Real-time video generation with pyramid attention broadcast. *arXiv preprint arXiv:2408.12588*, 2024.
- [34] 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.
- [35] Xin Zhou, Dingkan Liang, Kaijin Chen, Tianrui Feng, Xiwu Chen, Hongkai Lin, Yikang Ding, Feiyang Tan, Hengshuang Zhao, and Xiang Bai. Less is enough: Training-free video diffusion acceleration via runtime-adaptive caching. *arXiv preprint arXiv:2507.02860*, 2025.
- [36] Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. 2024.

A Technical Appendices and Supplementary Material

A.1 Implementation Details

Due to the modular design of AlignCache, the method has several hyperparameters. To achieve a better quality-speed trade-off, we tune these hyperparameters by sweeping over 5 prompts with 2 seeds per prompt. The final hyperparameters are as follows. For Wan2.1-1.3B, we set the base cache threshold to 0.065 with 12 warm-up exact CFG pairs, and use phase-adaptive threshold scales of 1.15, 1.00, and 0.80 for the early, middle, and late denoising phases, respectively, corresponding to phase thresholds of 0.07475, 0.065, and 0.052. For Open-Sora-1.2, we use a base threshold of 0.11, 8 warm-up pairs, and phase thresholds of 0.1045, 0.1265, and 0.11. For HunyuanVideo, we use the setting with a base threshold of 0.024, 10 warm-up pairs, and phase thresholds of 0.0252, 0.0288, and 0.0324. Unless otherwise specified, the phase boundaries are set to 15% and 86% of the denoising trajectory for Wan2.1-1.3B/Open-Sora-1.2, and 18% and 72% for HunyuanVideo. For the principal quantitative evaluations reported on a subset of the VBench default prompts, we generate 5 or 10 video samples, depending on the official recommendations for different evaluation dimensions, with different seeds for each prompt on NVIDIA RTX PRO 6000-96GB GPUs. To expedite ablation studies, we generate 2 videos per prompt with seeds 42 and 954, using Wan2.1-1.3B as the baseline.

A.2 Detailed Open-Loop Budget Scheduling

In existing open-loop schedulers [35] referenced in Section 3, the cumulative risk budget B_t is maintained by accumulating, at each timestep, a heuristic risk increment:

$$\Delta B_t = g_{a(t)} \cdot \frac{\text{mean}(|\mathbf{x}_t - \mathbf{x}_{t-1}|)}{\text{mean}(|\mathbf{y}_{a(t)}|)}, \quad (4)$$

where $g_{a(t)}$ is a gain factor estimated from the two most recent anchors. The timestep is skipped whenever $B_t < \tau$ for a fixed global threshold τ , and B_t is reset upon each exact computation. The three structural limitations of this design (single global threshold, lack of staleness constraint, lack of error validation) are discussed in the main text.

A.3 Detailed Phase-Adaptive Budget Routing

This section provides the detailed update rules underlying the phase-adaptive routing module summarized in Section 4.1.

Phase Partition. Let $\rho_t = t / \max(T-1, 1)$ denote the normalized progress at timestep t . The phase assignment is:

$$p_t = \begin{cases} \text{early}, & \text{if } \rho_t < \rho_e, \\ \text{mid}, & \text{if } \rho_e \leq \rho_t < \rho_l, \\ \text{late}, & \text{if } \rho_t \geq \rho_l, \end{cases} \quad (5)$$

where ρ_e and ρ_l are the early-to-mid and mid-to-late transition ratios, respectively.

Cumulative Risk Budget Update. At timestep t , the instantaneous input variation is $\delta_t = \text{mean}(|\mathbf{x}_t - \mathbf{x}_{t-1}|)$, and the budget increment is:

$$\Delta B_t = g_{a(t)} \cdot \frac{\delta_t}{\text{mean}(|\mathbf{y}_{a(t)}|)}, \quad (6)$$

where $a(t)$ is the index of the most recent exact anchor, $\mathbf{y}_{a(t)}$ is its cached output, and $g_{a(t)}$ is a motion gain factor reflecting local input-output sensitivity. The budget accumulates as $B_t = B_{t-1} + \Delta B_t$ and is reset to zero after each exact computation. Intuitively, Eq. (6) normalizes the input perturbation by the output magnitude and amplifies it by the learned gain, yielding a unit-free risk estimate.

Motion Gain Update. The motion gain $g_{a(t)}$ is updated after each exact computation using the ratio of output variation to input variation between the two most recent exact anchors:

$$g_{a(t)} = \frac{\text{mean}(|\mathbf{y}_{a(t)} - \mathbf{y}_{a^-(t)}|)}{\text{mean}(|\mathbf{x}_{a(t)} - \mathbf{x}_{a^-(t)}|)}, \quad (7)$$

where $a^-(t)$ denotes the exact anchor immediately preceding $a(t)$. This ratio adapts the budget sensitivity to the local dynamics of the diffusion trajectory.

Dynamic Threshold. The dynamic threshold used in the routing rule (Eq. (1)) is:

$$\Theta_{p,t} = \theta_p \cdot s_{p,t}, \quad (8)$$

where θ_p is the base phase threshold and $s_{p,t}$ is the online scale factor maintained by the closed-loop calibration module (Appendix A.4).

A.4 Detailed Closed-Loop Threshold Calibration

This section provides the detailed update equations underlying the closed-loop calibration module summarized in Section 4.2. Building on the calibration ratio $r_t = E_t^{\text{obs}}/B_t$ defined in the main text, we maintain a per-phase clipped exponential moving average:

$$\bar{r}_{p,t} = \lambda \bar{r}_{p,t-1} + (1 - \lambda) \text{clip}(r_t, r_{\min}, r_{\max}), \quad (9)$$

where λ is the EMA decay factor controlling the smoothness of the estimate, and $[r_{\min}, r_{\max}]$ are clipping bounds preventing outlier observations from destabilizing the controller. The threshold scale factor is then:

$$s_{p,t} = \text{clip}\left(\frac{1}{\bar{r}_{p,t}}, s_{\min}^{(p)}, s_{\max}^{(p)}\right), \quad (10)$$

where $s_{\min}^{(p)}$ and $s_{\max}^{(p)}$ are per-phase bounds. The inverse relationship in Eq. (10) is what makes the loop self-correcting: when the budget consistently underestimates error ($\bar{r}_{p,t} > 1$), the scale factor decreases, tightening $\Theta_{p,t}$ and triggering more frequent exact computations; conversely, when the budget is overly conservative ($\bar{r}_{p,t} < 1$), the threshold relaxes to permit additional skips.

A.5 Detailed Momentum-Augmented Draft Synthesis

This section provides the detailed equations underlying the gated momentum and anchor-guided correction summarized in Section 4.3.

Direction-Magnitude Gated Momentum. The adaptive momentum weight $\beta_p^{(b)}$ in Eq. (3) is modulated by a gating function that jointly evaluates directional coherence and magnitude consistency:

$$\beta_p^{(b)} = \beta_0^{(p)} \cdot \underbrace{\max\left(0, \text{cossim}(\mathbf{r}_{a(t)}^{(b)}, \mathbf{r}_{a^-(t)}^{(b)})\right)}_{\text{direction gate}} \cdot \underbrace{\min\left(\frac{\|\mathbf{r}_{a(t)}^{(b)}\|}{\|\mathbf{r}_{a^-(t)}^{(b)}\|}, \frac{\|\mathbf{r}_{a^-(t)}^{(b)}\|}{\|\mathbf{r}_{a(t)}^{(b)}\|}\right)}_{\text{magnitude gate}}, \quad (11)$$

where $\beta_0^{(p)}$ is a phase-dependent base momentum. The direction gate suppresses extrapolation when consecutive anchor residuals diverge in orientation (negative cosine similarity is clamped to zero), while the magnitude gate attenuates the correction when residual norms change rapidly, preventing over-extrapolation during transient dynamics. When both residuals are well-aligned and similar in magnitude, $\beta_p^{(b)}$ approaches $\beta_0^{(p)}$, granting the full momentum correction; otherwise the gate gracefully decays toward zero, falling back to the first-order scaled estimate.

Anchor-Guided Unconditional Blending. The final unconditional draft (Section 4.3) blends the extrapolation baseline with the anchor-transferred estimate:

$$\hat{\mathbf{u}}_t = (1 - \omega) \hat{\mathbf{u}}_t^{\text{base}} + \omega \tilde{\mathbf{u}}_t, \quad (12)$$

where $\hat{\mathbf{u}}_t^{\text{base}}$ is the extrapolation output from Eq. (3), $\tilde{\mathbf{u}}_t = \mathbf{u}_a + \alpha_p(\hat{\mathbf{c}}_t - \mathbf{c}_a)$ is the anchor-transferred unconditional estimate using a phase-dependent transfer coefficient α_p , and ω is a blending weight. This correction leverages the conditional–unconditional coupling inherent in CFG to produce more faithful unconditional drafts, which in turn yields more accurate guided approximations during both skip execution and counterfactual error measurement.

A.6 Full Ablation Grids for Scheduler-Related Axes

We provide the full grids underlying Table 2 of the main paper.

Table 4: Full grid for warm-up steps W .

W	Lat.(s)↓	Spd.↑	PSNR↑	SSIM↑	LPIPS↓
5	42.73	2.29×	23.68	0.829	0.123
6	44.17	2.19×	23.51	0.810	0.129
7	43.12	2.26×	24.40	0.850	0.110
8	44.42	2.18×	24.05	0.840	0.112
9	43.72	2.22×	24.51	0.862	0.104
12	46.82	2.06×	25.97	0.882	0.086
14	49.03	1.97×	26.34	0.887	0.078

Table 5: Full grid for error base threshold θ_p .

θ_p	Lat.(s)↓	Spd.↑	PSNR↑	SSIM↑	LPIPS↓
0.0325	53.77	1.81×	27.20	0.904	0.070
0.0350	54.06	1.79×	26.58	0.892	0.076
0.0375	50.29	1.94×	26.04	0.888	0.082
0.0400	50.42	1.91×	25.48	0.870	0.090
0.0425	47.21	2.06×	25.58	0.880	0.088
0.0475	45.06	2.16×	24.93	0.868	0.098
0.0600	41.75	2.31×	24.05	0.843	0.114
0.0700	39.35	2.45×	23.65	0.832	0.123

A.7 Case Studies

We provide additional qualitative comparisons covering all baselines (PAB, TeaCache, FasterCache, EasyCache, and AlignCache) on representative VBench prompts across two backbones, complementing the main-paper Figure 4.

Figure 5 compares all methods on “A person is kissing.” generated by Wan2.1-1.3B. Facial regions and lip contours are highly sensitive to caching-induced errors. PAB and TeaCache produce visible facial distortions, FasterCache exhibits mild ghosting, and EasyCache smooths out subtle expression cues at 2.38×. AlignCache, at the same 2.38× speedup, faithfully preserves facial geometry, lip contour, and skin texture (red boxes).

Figure 6 reports a comparison on “A bird building a nest from twigs and leaves.” generated by Open-Sora-1.2, a prompt dominated by high-frequency natural textures. Competing methods produce

Table 6: Full grid for phase ratio (ρ_e, ρ_l) . The full 11-point grid confirms low sensitivity: PSNR varies by < 0.55 dB across all configurations.

(ρ_e, ρ_l)	Lat.(s)↓	Spd.↑	PSNR↑	SSIM↑	LPIPS↓
(0.100, 0.86)	45.17	2.14×	24.70	0.860	0.100
(0.125, 0.84)	44.48	2.18×	25.13	0.874	0.095
(0.125, 0.88)	44.29	2.19×	25.11	0.873	0.095
(0.150, 0.80)	45.15	2.14×	24.61	0.859	0.102
(0.150, 0.82)	44.46	2.18×	25.00	0.872	0.097
(0.150, 0.85)	45.12	2.14×	24.61	0.859	0.102
(0.150, 0.88)	44.28	2.19×	24.99	0.871	0.097
(0.150, 0.90)	45.17	2.14×	24.59	0.859	0.101
(0.175, 0.84)	44.47	2.18×	25.06	0.873	0.096
(0.175, 0.88)	44.28	2.19×	25.05	0.873	0.096
(0.200, 0.86)	45.14	2.14×	24.65	0.859	0.100

blurred twig boundaries, texture flattening, or color drift on the nest rim, while AlignCache, at the highest $1.64\times$ speedup, recovers the woven twig structure and fine feather details (red boxes). This is consistent with our analysis in Section 5.4: high-frequency textures are where direct residual reuse incurs the largest error, and E_t^{obs} enables AlignCache to detect and correct such regions in real time.

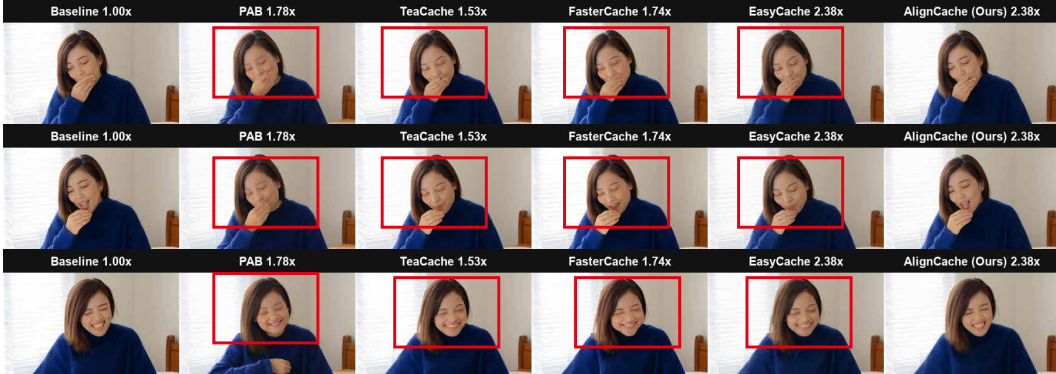


Figure 5: **Qualitative comparison** on “A person is kissing.” generated by Wan2.1-1.3B. Red boxes highlight facial regions where PAB, TeaCache, FasterCache, and EasyCache introduce distortions in lip contour, skin tone, or identity-bearing details, while AlignCache preserves facial geometry and expression cues at the same $2.38\times$ speedup as EasyCache.

A.8 Broader Impacts

AlignCache reduces the inference cost of video diffusion models, lowering the energy footprint of large-scale deployment and improving accessibility for researchers with limited GPU resources. As a training-free acceleration layer, it inherits the dual-use risks of the underlying generators, faster inference could lower the barrier for misuse such as deepfakes or disinformation, and any biases of the base model are preserved. We therefore recommend deploying AlignCache alongside existing upstream safeguards, including content moderation, provenance watermarking, and the release policies of the original video generators.



Figure 6: **Qualitative comparison** on “A bird building a nest from twigs and leaves.” generated by Open-Sora-1.2. Red boxes mark high-frequency texture regions (nest twigs, foliage, feathers) where competing methods produce blurring, structural collapse, or color drift, whereas AlignCache retains fine woven structure and feather details at the highest $1.64\times$ speedup among compared methods.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [Yes]

Justification: The abstract and introduction clearly state the problem setting, the proposed AlignCache framework, and the claimed speed–quality trade-offs. These claims are supported by experiments on three video diffusion models and are scoped to training-free cache-based acceleration for video diffusion models.

Guidelines:

- The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: Section 6 discusses remaining open directions and limitations, including prompt-independent calibration, warm-up overhead, and extension to long-form video generation and multimodal diffusion pipelines.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate “Limitations” section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not present formal theoretical results, theorems, or proofs. The equations are used to define the proposed scheduling, calibration, and draft synthesis procedures.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper describes the algorithmic components in Section 4 and reports the evaluated models, metrics, sampling settings, hyperparameters, prompt/sample counts, seeds, and GPU platforms in Section 5.1 and Appendix A.1. The paper also provides an anonymized code URL including scripts and instructions.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.

- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The paper provides an anonymized code URL and uses public video diffusion backbones, public baselines where available. The released repository provides the implementation and reproduction instructions for the reported results.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://neurips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://neurips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: Section 5.1 and Appendix A.1 specify the baselines, video generation models, resolutions, denoising steps, metrics, prompt/sample counts, seeds, hyperparameters, phase boundaries, and GPU platforms. Since the method is training-free, optimizer and training-split details are not applicable.

Guidelines:

- The answer [\[N/A\]](#) means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[N/A\]](#)

Justification: Following the standard evaluation protocol of training-free diffusion acceleration works (e.g., TeaCache, FasterCache, EasyCache), we report aggregated metrics over multiple prompts and seeds rather than per-run error bars. Since our method is deterministic given a fixed seed and the relative ordering of methods is consistent across all evaluated prompts and backbones, formal significance testing is not the primary mode of comparison in this line of work.

Guidelines:

- The answer [\[N/A\]](#) means that the paper does not include experiments.
- The authors should answer [\[Yes\]](#) if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: The paper reports GPU types and per-method latency/speedup, including RTX PRO 6000-96GB, H100-80GB, and Blackwell/Hopper comparisons.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research proposes a training-free inference acceleration method evaluated on existing video generation models and benchmarks. We are not aware of any aspect of the work that deviates from the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: A broader-impact discussion is provided at the Appendix A.8, covering efficiency/accessibility benefits, dual-use risks inherited from base generators, and mitigation via upstream safeguards.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper releases an acceleration method for existing video diffusion systems, while safeguards for the underlying generators are outside the scope of this work.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper cites the existing video generators' versions, baselines, and evaluation benchmark used in the experiments.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper introduces and releases the AlignCache code as a new asset through an anonymized repository. Documentation and reproduction instructions are provided with the repository, and the method itself is described in Sections 4 and A.1.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing, user studies, or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve human subjects, so IRB or equivalent approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as an important, original, or non-standard component. Any possible use of LLMs for writing, editing, or formatting would not affect the scientific methodology, rigor, or originality of the work.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.