RETHINKING DIFFUSION CACHING: FROM HISTORICAL EXTRAPOLATION TO CURRENT-OBSERVATION COMPLETION

Anonymous authors
Paper under double-blind review

ABSTRACT

Training-free diffusion caching reconstructs skipped features from historical evaluations. Its accuracy depends on whether the cached directions can represent the current residual motion. We identify a representation bottleneck: on Wan2.1, the latest input secant captures 98.9% of input displacement energy, while the latest residual secant captures 2.7% of the deployed CFG-weighted residual displacement energy. We characterize the motion that a fixed historical basis cannot represent and use oracle reconstruction to measure this limitation. This motivates MoorCache, which combines historical prediction with *current-observation completion*. History propagates persistent motion, and an affordable current observation supplies complementary directions through a projected displacement update. Same-cost ablations demonstrate the value of this completion, while projection controls test how current information should be combined with history. Across Wan2.1, HunyuanVideo, DiT-XL/2, and FLUX.1-dev, MoorCache improves quality–efficiency trade-offs with observation and reconstruction costs. On HunyuanVideo, it achieves $2.784\times$ measured speedup with 38.506 dB reference PSNR, connecting the representation analysis to practical diffusion acceleration.

1 INTRODUCTION

Diffusion Transformers (DiTs) Peebles & Xie (2023) achieve strong image and video generation quality but repeatedly execute an expensive Transformer during denoising. Training-free caching reduces this cost by skipping selected evaluations and reconstructing their outputs from previously computed features Liu et al. (2025a); Ma et al. (2024); Liu et al. (2025b). Existing methods have largely focused on two questions, *when* should computation be reused, and *how far* should cached representations be extrapolated. Both implicitly assume a third property, that the historical residual direction remains informative at the current timestep. We revisit this assumption and ask a complementary question, *where should the residual move?*

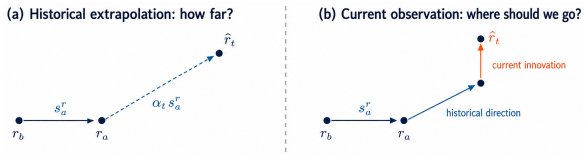

Figure 1: **Rethinking diffusion cache reconstruction.** History-based methods fix *when* to compute and *how far* to extrapolate, our diagnosis adds a complementary question, *where* the residual should move, and completes the missing direction with a current observation.

Our analysis reveals a substantial asymmetry, input motion is highly predictable while residual motion is not. On Wan2.1 Wan et al. (2025), extrapolating along the input's most recent direction explains 98.9% of the input displacement energy, whereas the same historical extrapolation explains only 2.7% of the deployed CFG-weighted residual displacement energy. In a separate unconditional-residual diagnostic, the offline-optimal step size along history leaves median displacement NMSE at 31.13%, while supplying the missing current direction lowers it to 23.62%. A richer short-history span also recovers only a small fraction of the missing directional energy on its eligible support. These observations suggest that the dominant limitation is often not *how far* to move along history, but motion the historical span cannot represent.

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.

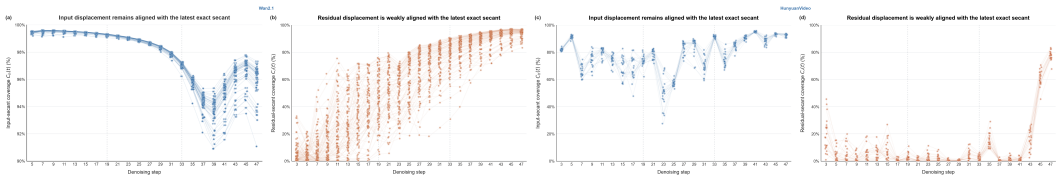

Figure 2: **Input and residual secant coverage across backbones.** (a,b) Wan2.1 Wan et al. (2025) and (c,d) HunyuanVideo Kong et al. (2024) trajectories. In both backbones, input displacement is substantially better aligned with the latest exact secant than residual displacement.

This diagnosis motivates *current-observation completion*. We decompose reconstruction into two roles, history propagates the persistent component of the residual trajectory, while a current observation supplies complementary motion outside the historical span. The key change is therefore informational, since rather than only reweighting historical directions, reconstruction obtains a new direction from information available at the current timestep. We evaluate the framework on four image and video diffusion backbones under multiple acceleration budgets, and all observation and reconstruction costs are included in the reported FLOPs and latency.

We make three contributions to the analysis, reconstruction, and evaluation of diffusion caching:

- **A representation-based diagnosis of diffusion caching.** We characterize the error that coefficient selection cannot remove within a fixed historical span, and use secant-coverage and oracle controls to measure the missing residual motion.
- **Current-observation completion.** We formulate cache reconstruction as historical-span prediction plus current-observation completion, history propagates the persistent motion, while a current observation supplies the missing motion outside the historical span.
- **Validation across model families.** We validate this idea on Wan2.1 Wan et al. (2025), HunyuanVideo Kong et al. (2024), DiT-XL/2 Peebles & Xie (2023), and FLUX.1-dev Labs (2024), the reconstructions consistently improve quality–efficiency trade-offs over existing caches, and same-cost ablations isolate the contribution of current observations.

2 Related Work

Training-free diffusion caching. Existing caching methods reduce denoising cost by deciding when or where to reuse previous computation. Learning-to-Cache allocates caching across layers and timesteps, while TeaCache, SeaCache, MagCache, EasyCache, DiCache, SenCache, and DPCache use input changes, runtime statistics, sensitivity, or trajectory-level criteria to determine reuse Ma et al. (2024); Liu et al. (2025a); Chung et al. (2026); Ma et al. (2025); Zhou et al. (2025); Bu et al. (2026); Haghighi & Alahi (2026); Cui et al. (2026). ToCa and DuCa operate at token or feature granularity Zou et al. (2025; 2026). A complementary line improves reconstruction from historical information: TaylorSeer extrapolates temporal derivatives Liu et al. (2025b); D2Cache, TC-Padé, and HiCache use higher-order or richer trajectory models Liu et al. (2026); He et al. (2026); Feng et al. (2026); L^2P learns combinations of historical features Shen et al. (2026); and SVD-Cache decomposes stable and volatile feature subspaces Chen et al. (2026a). MoorCache is complementary to these approaches: rather than only changing the schedule or enriching the historical predictor, it asks whether the missing motion lies outside the historical span altogether.

Current-signal and CFG-aware acceleration. Several methods exploit information available at the current timestep. Adaptive Guidance selectively removes CFG evaluations, while FasterCache uses same-timestep conditional–unconditional redundancy for reconstruction Castillo et al. (2025); Lyu et al. (2025). RFC transfers the relation between current input and historical output motion into a Taylor-style prediction Son et al. (2026). ResCa obtains sparse on-step information through native-resolution proxy tokens Fang et al. (2026), whereas Bottleneck Sampling changes the resolution schedule by running full low-resolution denoising in intermediate stages Tian et al. (2025). MoorCache differs in using current information as a *new reconstruction direction*: a current residual observation supplements historical prediction, with source and alignment set per unit.

Connections to predictor–corrector and multirate views. Our formulation resembles a predictor–corrector structure in which a historical model propagates a state and a current measurement supplies an innovation Kalman (1960); Luenberger (1964). The low-resolution observation is also related to coarse models in multirate and parareal methods Gear & Wells (1984); Lions et al. (2001), as well as multi-resolution generation Ho et al. (2022); Saharia et al. (2022); Jin et al. (2025).

These connections motivate separating historical prediction from a current innovation; MoorCache realizes this structure with cached residuals and a fixed observation gain.

3 RETHINKING: THE MISSING RESIDUAL DIRECTION

3.1 PROBLEM SETUP

Consider a cacheable Transformer unit with observable input h_t and residual output r_t:

$$\mathcal{F}_t(h_t) = h_t + r_t, \qquad r_t = \mathcal{F}_t(h_t) - h_t, \tag{1}$$

At a skip step, h_t remains available but r_t does not. Let $z \in \{h, r\}$, and let $b < a < t$ be the two latest exact steps. The secant, displacement, projection coefficient, and projection are

$$\begin{aligned} s_a^z &= z_a - z_b, & d_t^z &= z_t - z_a, \\ \alpha_t^z &= \frac{\langle d_t^z, s_a^z \rangle}{\|s_a^z\|_2^2 + \varepsilon}, & \Pi_a^z(d_t^z) &= \alpha_t^z s_a^z. \end{aligned} \tag{2}$$

For $z = h$, the terms are observable; for $z = r$, d_t^r and α_t^r require the skipped residual for diagnosis.

3.2 FINDING 1: INPUT DISPLACEMENT FOLLOWS A STABLE LOCAL SECANT

Feature similarity, $h_t \approx h_a$, does not imply predictable motion. We therefore measure the fraction of displacement energy captured by the one-dimensional subspace in Equation 2:

$$C_z(t) = 1 - \frac{\|d_t^z - \Pi_a^z(d_t^z)\|_2^2}{\|d_t^z\|_2^2 + \varepsilon} \simeq \frac{\|\Pi_a^z(d_t^z)\|_2^2}{\|d_t^z\|_2^2} = \cos^2 \angle(d_t^z, s_a^z). \tag{3}$$

The middle identity is the Pythagorean decomposition of an orthogonal projection; the stabilizer ε only adds a relative slack of order $\varepsilon / \|d_t^z\|_2^2$. On Wan2.1 Wan et al. (2025), the median input coverage is $C_h = 98.9\%$: the observable displacement is almost one-dimensional along the latest exact input secant. The older-secant comparison confirms that this direction is local and must be refreshed. Dataset construction, stagewise values, and locality statistics are reported in Appendix D.

3.3 FINDING 2: RESIDUAL DISPLACEMENT IS WEAKLY ALIGNED WITH TEMPORAL SECANT

For $z = r$, the same operator bounds predictors on the latest residual secant. The deployed denoising update uses the CFG-weighted residual $r_t = r_t^u + w\,(r_t^c - r_t^u)$ with $w = 5$, whose conditional and unconditional components share the same noisy latent and timestep. Measuring coverage on this deployed residual, we find that despite $C_h = 98.9\%$, the median eventwise residual coverage is only

$$\underset{\text{events}}{\text{median}}\ C_r^{\text{final}} = 2.7\%, \tag{4}$$

The event distribution is strongly right-skewed—the 90th percentile is around 40%—so coverage is low at most events but not all. The CFG-weighted displacement amplifies cross-branch gap motion; Appendix D analyzes this effect. Coverage depends on the reconstruction target: intermediate features have median coverage of 67%–70%, indicating a remaining representation error but not the near-total coverage loss of the combined residual. On HunyuanVideo Kong et al. (2024), median deployed-residual coverage is 2.5%. These measurements motivate examining the motion missed by history on each target rather than assuming a common coverage value across cache locations.

3.4 FINDING 3: COEFFICIENT SELECTION IS SECONDARY TO RECONSTRUCTION DIRECTION

Every scalar extrapolator on the latest residual secant has the form $\widehat{r}_t(\alpha) = r_a + \alpha s_a^r$ with displacement NMSE $E_t(\alpha) = \|r_t - \widehat{r}_t(\alpha)\|_2^2 / \max(\|r_t - r_a\|_2^2, \varepsilon)$, whose non-deployable optimum, when ignoring the stabilizer, is $\alpha_t^{r\star} = \langle d_t^r, s_a^r \rangle / \|s_a^r\|_2^2$. Projection orthogonality then yields the exact decomposition

$$\left\|d_t^r - \alpha_t^h s_a^r\right\|_2^2 = \underbrace{\|d_t^r - \alpha_t^{r\star} s_a^r\|_2^2}_{\text{direction / representation error}} + \underbrace{\|s_a^r\|_2^2 \left(\alpha_t^h - \alpha_t^{r\star}\right)^2}_{\text{coefficient error}}. \tag{5}$$

The first term is the irreducible error within $\operatorname{span}(s_a^r)$; coefficient selection changes only the second. We compute displacement NMSE per event, then report $100 \operatorname{median}_t E_t$. On the same 2,024 events, replay gives 100%, input projection 35.771%, the optimal coefficient 35.167%, and the norm ratio

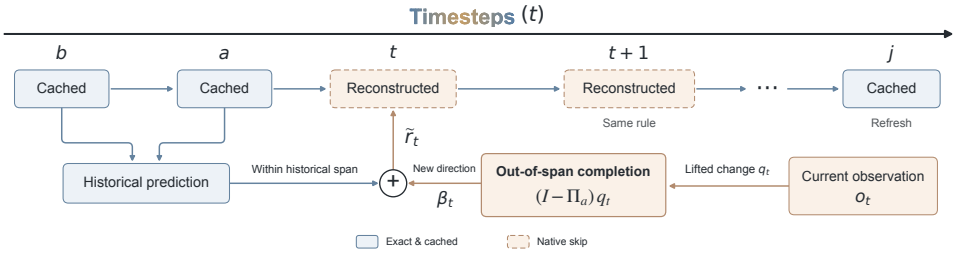

Figure 3: **Current-observation completion.** Cached anchors propagate a historical prediction within the historical span, while a current observation o_t contributes an out-of-span completion $(I - \Pi_a)q_t$; the same rule is refreshed at each exact evaluation.

35.747%. The latter three differ by at most 0.605 percentage points. Relative to replay, the input coefficient recovers 99.07% of the reduction attained by the optimal coefficient, using the ratio of reductions in median displacement NMSE (Appendix D). Most remaining error is therefore outside the historical direction on this support. Coefficient transfer is model-specific: on HunyuanVideo Kong et al. (2024), input and optimal coefficients have Spearman correlation -0.03.

3.5 Representation limits of historical reconstruction

Fix the target, anchor r_a, and historical-transition matrix H_t, with $\mathcal{H}_t = \mathrm{col}(H_t)$. Every predictor $r_a + H_t\alpha_t$ lies in this affine space, regardless of its coefficient rule. For $d_t^r = r_t - r_a$, the minimum error is

$$\min_{\alpha} \|d_t^r - H_t\alpha\|_2^2 = \|e_t^{\perp}\|_2^2, \qquad e_t^{\perp} = (I - P_{\mathcal{H}_t})d_t^r. \tag{6}$$

Target-informed least squares attains this bound; coefficient selection cannot recover the orthogonal component.

Eight-secant oracle. On matched eligible late events, expanding the historical basis from one to eight secants increases median coverage from 77.80% to 78.95%; the median paired gain is 0.70 percentage points. Target-informed least squares measures the motion recoverable from each basis. The limited additional coverage motivates obtaining complementary information from the current timestep. Appendix D specifies the diagnostic target and eligible support.

Scope. This characterization applies to the fixed anchor and specified historical basis, including time-dependent polynomial coefficients within that affine space. Additional history or transformed features define a different representation space.

3.6 From historical extrapolation to current-observation completion

The analysis above motivates a simple principle: *history should predict what is persistent, while a current observation should supply what history cannot express*. Let o_t be a low-cost quantity observed at this step and let $\mathcal{L}_t$ lift its displacement into the target-residual space:

$$q_t = \mathcal{L}_t(o_t - o_a), \qquad \Pi_a(v) = \frac{\langle v, s_a^r \rangle}{\|s_a^r\|_2^2 + \varepsilon} s_a^r, \qquad \widehat{r}_t = r_a + \widetilde{\alpha}_t s_a^r + \beta_t(I - \Pi_a)q_t, \tag{7}$$

which separates reconstruction into a historical-span term and a current-observation innovation term.

The observation expands the available reconstruction directions beyond the cached span, while projection removes the component already represented by history. Section 4 develops the resulting update, its error reduction condition, and the inference procedure.

4 Current-Observation Completion for Diffusion Caching

MoorCache completes historical prediction with current observations. The unified update assigns persistent motion to history and complementary motion to the observation, with projection organizing their contributions. We characterize when this completion reduces reconstruction error.

4.1 HISTORICAL PREDICTION AND ITS MISSING MOTION

Let r_t be the target residual, $(h_b, r_b), (h_a, r_a)$ the available references, and $s_a^r = r_a - r_b$ the latest residual secant. Observable input motion estimates progress along this direction:

$$\alpha_t^h = \frac{\langle h_t - h_a, h_a - h_b \rangle}{\|h_a - h_b\|_2^2 + \varepsilon}, \qquad \widetilde{\alpha}_t^h = \mathrm{clip}(\alpha_t^h, -2, 2). \tag{8}$$

For an available parallel estimate $\widetilde{\alpha}_t$, define the historical-span prediction and its unknown error as

$$\bar{r}_t = r_a + \widetilde{\alpha}_t s_a^r, \qquad e_t = r_t - \bar{r}_t. \tag{9}$$

The parallel estimate can use $\widetilde{\alpha}_t^h$ or current observed displacement. Either choice leaves $\bar{r}_t - r_a$ in the cached span. As Section 3.5 shows, coefficient selection cannot recover motion outside that span; this missing motion motivates correcting the prediction.

4.2 EXTRACTING COMPLEMENTARY CURRENT INFORMATION

Let o_t be an affordable current observation, o_a its reference, and $\mathcal{L}_t$ its alignment into target space. Using displacement removes the reference offset. With Π_a from Equation 7,

$$q_t = \mathcal{L}_t(o_t - o_a), \qquad q_t^{\parallel} = \Pi_a q_t, \qquad q_t^{\perp} = (I - \Pi_a) q_t. \tag{10}$$

Adding $q_t^{\parallel}$ can duplicate motion already represented by the parallel estimate. We therefore use $q_t^{\perp}$ as the complementary correction. This regularized decomposition approaches an orthogonal one as $\varepsilon \to 0$ for a nonzero secant; it does not assume that the observation equals the target.

4.3 UNIFIED RECONSTRUCTION AND ERROR REDUCTION

We combine the parallel estimate and the complementary observation in a single residual update:

$$\widetilde{r}_t = \bar{r}_t + \beta_t q_t^{\perp} = r_a + \widetilde{\alpha}_t s_a^r + \beta_t q_t^{\perp}, \qquad r_t - \widetilde{r}_t = e_t - \beta_t q_t^{\perp}. \tag{11}$$

Here β_t controls the amount of complementary motion admitted into the prediction. All deployed observation corrections use $\beta_t = 1$; $\beta_t = 0$ retains only the parallel estimate. Expanding the squared error gives the exact improvement over that estimate:

$$\|e_t\|_2^2 - \|r_t - \widetilde{r}_t\|_2^2 = 2\beta_t \langle e_t, q_t^{\perp} \rangle - \beta_t^2 \|q_t^{\perp}\|_2^2. \tag{12}$$

Thus a current observation helps when its complementary displacement aligns with the missing residual motion strongly enough to offset the energy it introduces. For $\beta_t > 0$ and $q_t^{\perp} \neq 0$, strict improvement occurs precisely when

$$\langle e_t, q_t^{\perp} \rangle > 0, \qquad 0 < \beta_t < \frac{2\langle e_t, q_t^{\perp} \rangle}{\|q_t^{\perp}\|_2^2}. \tag{13}$$

The condition identifies directional agreement and controlled magnitude as requirements for improvement. This identity characterizes reconstruction error using the unknown e_t; deployment uses the fixed gain $\beta_t = 1$. Section 5.4 tests the resulting prediction with raw and projected observations under fixed and offline-optimal gains.

4.4 INFERENCE PROCEDURE

A fixed, prompt-independent schedule selects exact steps $\mathcal{E}$ and observation-assisted steps $\mathcal{O}$. Reconstruction operates on the original solver grid:

$$r_t^{\mathrm{use}} = \begin{cases} \mathcal{F}_t(h_t) - h_t, & t \in \mathcal{E}, \\ \bar{r}_t + \beta_t q_t^{\perp}, & t \in \mathcal{O}, \\ r_a + \widetilde{\alpha}_t^h s_a^r, & \text{otherwise}, \end{cases} \qquad h_t^{\mathrm{out}} = h_t + r_t^{\mathrm{use}}. \tag{14}$$

Exact evaluations refresh target anchors on the accelerated trajectory. Algorithm 1 and Appendix B specify cache updates and reference management. Reported FLOPs and latency include observation, alignment, and reconstruction costs.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Model configurations. We evaluate Wan2.1-T2V-1.3B Wan et al. (2025) and HunyuanVideo 13B Kong et al. (2024) for video generation, FLUX.1-dev Labs (2024) for text-to-image generation, and DiT-XL/2 Peebles & Xie (2023) for class-conditional generation. Dense references use 50 denoising steps. DiT uses DDIM-50 with CFG 1.5. MoorCache reconstructs skipped evaluations on the original solver grid with fixed, prompt-independent schedules.

Evaluation data. FLUX generates 1024×1024 images for 200 DrawBench prompts. Video evaluation uses 946 VBench prompts with five random seeds per prompt, producing 4,730 videos per configuration at 480p and 65 frames. DiT generates 50,000 images across 1,000 ImageNet classes at 256×256. Accelerated outputs share seeds and initial noise with their dense references.

Comparisons and metrics. We compare reduced-step sampling, replay, and 14 caching baselines covering historical prediction, token/feature reuse, and adaptive cache selection. Tables 1–4 report the backbone-specific methods and operating parameters at moderate and stronger acceleration; Appendix A lists all baselines. PSNR, SSIM, and LPIPS measure reference fidelity. VBench assesses video quality; ImageReward and CLIP score assess image preference and text alignment; FID, sFID, and IS evaluate class-conditional generation.

Efficiency. GBU experiments use NVIDIA H100 GPUs. We report measured latency and total FLOPs, including observation, alignment, and reconstruction costs, with speedups relative to each dense reference. Figure 6 compares measured operating points at matched speeds; connecting lines are visual guides. Appendix A provides additional model, baseline, and timing settings.

5.2 VIDEO GENERATION

On Wan2.1, MoorCache achieves $2.486\times$ measured speedup with 34.022 dB PSNR and 80.00 VBench, and $5.747\times$ with 23.181 dB PSNR and 77.55 VBench (Table 1). It leads VBench and reference fidelity among accelerated methods in each group. At the moderate operating point, SeaCache reaches 32.643 dB at $2.04\times$, while MoorCache improves fidelity and reduces latency to 39.09 s. Appendix E reports per-prompt fidelity; Table 10 gives VBench dimensions.

On HunyuanVideo, MoorCache reaches $2.784\times$ measured speedup with 38.506 dB PSNR and 81.99 VBench, and $4.972\times$ with 28.013 dB PSNR and 80.73 VBench (Table 2). It leads the reported quality metrics among the accelerated methods in both groups. Figure 6 provides the quality–latency curves, while Figure 4 shows aligned frames from the dense and accelerated videos. VBench measures video quality across multiple dimensions, while reference-fidelity metrics quantify agreement with dense outputs. Together they distinguish perceived quality from reference fidelity.

5.3 IMAGE GENERATION

Table 3 evaluates DiT-XL/2 under the ImageNet protocol in Section 5, reporting both FLOPs and measured latency.

On DiT-XL/2 Peebles & Xie (2023), MoorCache leads FID, sFID, and IS in both groups containing its operating points. Its 28-call route reaches $3.57\times$ nominal speedup with 7.600 FID, while the 22-call route reaches $4.545\times$ with 7.695 FID and 249.423 IS. Measured gains are smaller because per-block observation and reconstruction introduce additional overhead.

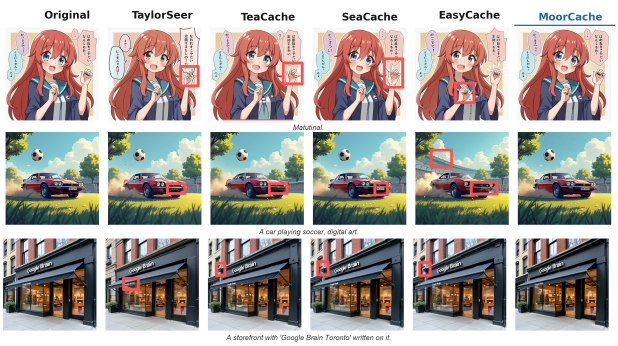

Figure 5: **FLUX.1-dev qualitative comparison.**

For FLUX, all rows use the same implementation, prompts, seeds, and dense reference. At $2.83\times$ measured speedup, MoorCache leads the reported quality metrics among accelerated methods, with 32.124 dB PSNR, 0.9615 SSIM, and 0.0766 LPIPS. In the stronger-acceleration group, MoorCache reaches $4.236\times$ measured speedup and leads ImageReward, CLIP score, SSIM, and LPIPS. Table 4 reports the much more details.

Table 1: **Text-to-video generation on Wan2.1-T2V-1.3B.**

Method	Acceleration				VBench ↑ Score (%)	Quality Metrics		
	Latency (s) ↓	Measured ↑	FLOPs (T) ↓	Theoretical ↑		PSNR ↑	SSIM ↑	LPIPS ↓
Dense, 50 steps Wan et al. (2025)	97.18	1.00×	27991.0	1.00×	80.30	–	–	–
SenCache Haghighi & Alahi (2026)	60.15	1.62×	17472.5	1.602×	79.92	31.225	0.9444	0.0323
MagCache Ma et al. (2025)	53.09	1.83×	15115.1	1.852×	79.86	31.116	0.9430	0.0339
TaylorSeer Liu et al. (2025b) ($N = 2, O = 1$)	50.00	1.94×	13995.5	2.000×	79.64	16.654	0.6395	0.3590
TeaCache Liu et al. (2025a) ($\ell = 0.08$)	48.64	2.00×	13995.5	2.000×	79.53	25.261	0.8690	0.0924
DiCache Bu et al. (2026) ($\tau = 0.08$)	49.30	1.99×	13950.5	2.006×	79.64	29.380	0.9296	0.0414
Reduced steps Wan et al. (2025) (25 NFE)	48.11	2.02×	13995.5	2.000×	79.49	16.703	0.6410	0.3568
SeaCache Chung et al. (2026) ($\ell = 0.20$)	47.59	2.04×	13422.6	2.085×	79.85	32.643	0.9502	0.0310
D2Cache Liu et al. (2026) ($\tau = 0.10$)	46.99	2.07×	13435.7	2.083×	78.39	24.694	0.8584	0.0966
EasyCache Zhou et al. (2025) ($\tau = 0.027$)	45.33	2.14×	13203.9	2.120×	79.84	31.253	0.9411	0.0321
MoorCache (40 calls, ours)	39.09	2.486×	11196.4	2.500×	**80.00**	**34.022**	**0.9616**	**0.0188**
TC-Padé He et al. (2026) ($N = 1.4, I = 8, O = 3$)	25.00	3.89×	6717.8	4.167×	76.65	21.215	0.7396	0.2696
TaylorSeer Liu et al. (2025b) ($N = 5, O = 1$)	21.11	4.60×	5598.2	5.000×	75.37	14.809	0.5527	0.4980
Reduced steps Wan et al. (2025) (10 NFE)	19.21	5.06×	5598.2	5.000×	76.26	14.769	0.5553	0.5013
TeaCache Liu et al. (2025a) ($\ell = 0.40$)	18.05	5.38×	5038.4	5.556×	75.35	16.814	0.6360	0.3921
MoorCache (17 calls, ours)	16.91	5.747×	4758.5	5.882×	**77.55**	**23.181**	**0.8168**	**0.1302**

Table 2: **Text-to-video generation on HunyuanVideo 13B.**

Method	Acceleration				VBench ↑ Score (%)	Quality Metrics		
	Latency (s) ↓	Measured ↑	FLOPs (T) ↓	Theoretical ↑		PSNR ↑	SSIM ↑	LPIPS ↓
Dense, 50 steps Kong et al. (2024)	94.19	1.00×	29773.0	1.00×	80.98	–	–	–
Reduced steps Kong et al. (2024) (25 NFE)	47.14	1.998×	14886.5	2.000×	81.28	26.358	0.8132	0.2045
Replay (∼26 native calls)	49.02	1.921×	15482.0	1.923×	81.04	29.211	0.8548	0.1412
TaylorSeer Liu et al. (2025b) ($N = 2$)	50.30	1.873×	14890.1	2.000×	81.59	27.123	0.8224	0.1807
ToCa Zou et al. (2025) ($N = 2$)	50.89	1.851×	15543.8	1.915×	81.72	25.944	0.7963	0.2082
DuCa Zou et al. (2026) ($N = 2$)	51.06	1.845×	15216.9	1.957×	81.04	25.972	0.7972	0.2072
TeaCache Liu et al. (2025a) ($\ell = 0.12$)	47.16	1.998×	14868.6	2.002×	80.65	28.922	0.8542	0.1427
MagCache Ma et al. (2025)	47.10	2.000×	14886.5	2.000×	81.28	37.929	0.9605	0.1347
EasyCache Zhou et al. (2025)	45.89	2.053×	14466.7	2.058×	81.06	30.354	0.9526	0.1553
SeaCache Chung et al. (2026) ($\ell = 0.20$)	47.53	1.982×	14774.9	2.015×	80.94	37.372	0.9477	0.1503
DiCache Bu et al. (2026) ($\tau = 0.10$)	39.53	2.383×	12335.0	2.414×	80.69	35.548	0.9417	0.1472
MoorCache (17 calls, ours)	33.825	2.784×	10587.4	2.812×	**81.99**	**38.506**	**0.9787**	**0.1278**
DPCache Cui et al. (2026) ($K = 13, O = 2$)	25.45	3.701×	7741.0	3.846×	79.41	22.974	0.7324	0.3269
SeaCache Chung et al. (2026) ($\ell = 0.40$)	27.10	3.476×	8336.4	3.571×	79.05	26.121	0.8089	0.2191
TaylorSeer Liu et al. (2025b) ($N = 5$)	23.17	4.072×	5960.4	4.995×	79.35	22.306	0.7236	0.3462
ToCa Zou et al. (2025) ($N = 5$)	25.20	3.739×	7006.2	4.250×	78.26	22.065	0.7243	0.3644
DuCa Zou et al. (2026) ($N = 5$)	22.14	4.255×	6483.2	4.592×	78.85	22.106	0.7282	0.3670
TeaCache Liu et al. (2025a) ($\ell = 0.40$)	20.30	4.649×	6326.8	4.706×	78.99	23.199	0.7542	0.3337
MoorCache (9 calls, ours)	18.94	4.972×	5937.2	5.014×	**80.73**	**28.013**	**0.8664**	**0.1197**

Latency and fidelity trade-offs. Relative to SeaCache, MoorCache-16 raises PSNR from 30.936 to 32.124 dB while reducing latency from 6.136 to 3.97 s, a 35.3% reduction. MoorCache-10 offers a higher-throughput operating point at 2.653 s, saving a further 33.2% of the 16-call latency with 22.104 dB PSNR, 0.8350 SSIM, and 0.1260 LPIPS. These operating points expose the practical choice between stronger dense-reference fidelity and a shorter generation time.

Quality at stronger acceleration. At the faster FLUX operating point, MoorCache-10 combines higher SSIM and lower LPIPS with a shorter latency than DiCache, whose PSNR is higher (22.997 versus 22.104 dB). The structural and perceptual metrics complement the pixelwise comparison in Table 4. The same-cost controls below test which part of MoorCache produces the fidelity gains.

Qualitative comparisons. All methods share prompts, seeds, and initial noise. The three selected prompts probe text rendering, object composition, and color fidelity. The galleries illustrate how accelerated reconstruction preserves these structures relative to the dense Original; Figure 7 provides the remaining prompt comparisons. These aligned examples connect the aggregate fidelity scores to visible content, while Appendix E reports complementary per-prompt distributions for Wan2.1.

Table 5: **Same-cost out-of-span ablation.**

Model	No innovation	MoorCache (✓)	ΔPSNR ↑
Wan2.1	31.342	**34.022**	+2.68
HunyuanVideo	36.132	**38.506**	+2.37
FLUX.1-dev	22.145	**32.124**	+9.98

Table 6: **Projection ablation on Wan2.1.**

Method (displacement NMSE)	$\beta = 1$	Oracle β
Historical span only	31.78%	–
Raw injection q_t	76.28%	23.83%
Projected injection $(I - \Pi_a)q_t$	**24.53%**	**20.20%**

Table 3: **Class-conditional image generation with DiT-XL/2 on ImageNet.**

Method	Acceleration				Generation Quality		
	Latency(s)	Speed	FLOPs(T)	Speed	FID↓	sFID↓	IS↑
DDIM-50 Song et al. (2021)	0.483	1.00×	23.74	1.00×	7.777	34.880	257.324
DDIM-25 Song et al. (2021)	0.256	1.89×	11.87	2.00×	8.464	34.839	245.336
EasyCache Zhou et al. (2025) ($\tau = 0.18$)	0.276	1.75×	12.73	1.87×	8.379	35.212	246.561
SeaCache Chung et al. (2026) ($\ell = 0.16$)	0.271	1.78×	12.40	1.91×	9.042	35.474	235.589
TeaCache Liu et al. (2025a) ($\ell = 0.12$)	0.254	1.90×	11.92	1.99×	8.416	35.157	245.568
ToCa Zou et al. (2025) ($N = 3$)	0.271	1.78×	10.23	2.32×	8.426	35.009	248.941
DuCa Zou et al. (2026) ($N = 3$)	0.252	1.92×	9.58	2.48×	8.424	35.000	248.599
DDIM-20 Song et al. (2021)	0.205	2.36×	9.49	2.50×	9.052	34.925	237.260
FORA Selvaraju et al. (2024) ($N = 3$)	0.213	2.27×	8.58	2.77×	8.875	35.766	245.797
TaylorSeer Liu et al. (2025b) ($N = 3, O = 3$)	0.306	1.58×	8.56	2.77×	7.884	35.130	256.530
DuCa Zou et al. (2026) ($N = 4$)	0.206	2.34×	7.61	3.11×	9.036	34.991	240.099
ToCa Zou et al. (2025) ($N = 5$)	0.210	2.30×	6.77	3.51×	12.061	37.280	202.782
TaylorSeer Liu et al. (2025b) ($N = 4, O = 4$)	0.297	1.63×	6.66	3.56×	8.005	35.443	254.870
MoorCache (28 calls, ours)	0.191	2.53×	6.65	3.57×	**7.600**	**33.826**	**257.679**
TC-Padé He et al. (2026) ($N = 1.4, I = 8, O = 3$)	0.163	2.97×	5.70	4.17×	16.538	34.051	233.640
FORA Selvaraju et al. (2024) ($N = 5$)	0.148	3.26×	5.24	4.53×	11.261	38.074	210.344
TaylorSeer Liu et al. (2025b) ($N = 5, O = 3$)	0.247	1.96×	5.24	4.53×	8.152	35.395	211.047
MoorCache (22 calls, ours)	0.137	3.52×	5.22	4.545×	**7.695**	**33.227**	**249.423**

Table 4: **Text-to-image generation with FLUX.1-dev on DrawBench.**

Method	Acceleration				Image Reward DrawBench↑	CLIP Score↑	PSNR↑	SSIM↑	LPIPS↓
	Latency(s)	Measured	FLOPs(T)	Theoretical					
FLUX.1[dev] Labs (2024)	11.237	1.00×	3719.50	1.00×	0.915	32.191	–	–	–
[dev]:60% steps Labs (2024)	6.741	1.67×	2231.70	1.67×	0.899	32.276	18.558	0.7653	0.2445
[dev]:40% steps Labs (2024)	4.538	2.48×	1487.80	2.50×	0.887	32.187	16.167	0.6886	0.3490
Δ-DiT Chen et al. (2026b) ($N = 2$)	7.691	1.461×	2480.01	1.50×	0.900	32.329	16.791	0.7148	0.3167
EasyCache Zhou et al. (2025) ($\tau = 0.20$)	6.579	1.708×	2212.36	1.681×	0.807	31.903	19.891	0.7942	0.2032
TeaCache Liu et al. (2025a) ($\ell = 0.30$)	6.105	1.840×	2008.53	1.852×	0.848	31.916	21.312	0.8208	0.1697
SeaCache Chung et al. (2026) ($\ell = 0.20$)	6.136	1.831×	1992.16	1.867×	0.907	32.193	30.936	0.9429	0.1462
Δ-DiT Chen et al. (2026b) ($N = 3$)	5.758	1.952×	1686.76	2.21×	0.897	32.370	15.018	0.6457	0.4111
TaylorSeer Liu et al. (2025b) ($N = 3, O = 2$)	5.046	2.227×	1320.07	2.82×	0.812	31.975	19.130	0.7625	0.2289
FORA Selvaraju et al. (2024) ($N = 3$)	4.136	2.717×	1320.07	2.82×	0.880	32.366	15.327	0.6531	0.3920
MoorCache (16 native calls, ours)	3.97	2.83×	1241.2	3.00×	**0.917**	**33.973**	**32.124**	**0.9615**	**0.0766**
ToCa Zou et al. (2025) ($N = 5$)	7.483	1.502×	1126.76	3.30×	0.869	32.032	17.299	0.6920	0.3249
DiCache Bu et al. (2026) ($\tau = 0.40$)	3.269	3.437×	1057.75	3.516×	0.874	32.255	**22.997**	0.8146	0.1850
DPCache Cui et al. (2026) ($K = 13, O = 2$)	2.977	3.775×	967.07	3.846×	0.866	32.109	21.204	0.8007	0.1843
TaylorSeer Liu et al. (2025b) ($N = 5, O = 2$)	3.775	2.977×	893.54	4.16×	0.870	32.177	16.344	0.6771	0.3460
FORA Selvaraju et al. (2024) ($N = 5$)	2.625	4.280×	893.54	4.16×	0.868	32.367	13.905	0.5768	0.4887
TC-Padé He et al. (2026) ($N = 1.4, I = 8, O = 3$)	2.846	3.949×	892.68	4.167×	0.839	32.225	20.152	0.7618	0.2584
MoorCache(10 native calls, ours)	2.653	4.236×	803.8	4.63×	**0.892**	**33.101**	22.104	**0.8350**	**0.1260**

5.4 Ablation Studies

Benefit of current-observation completion. At the same computational cost, restoring the out-of-span component improves PSNR by 2.680, 2.374, and 9.979 dB on Wan2.1, HunyuanVideo, and FLUX, with corresponding gains in SSIM and LPIPS (Table 5). The $\beta_t = 0$ controls use the main-table video routes and the 16-call FLUX route in Appendix D.1. Each control preserves the schedule, anchors, parallel estimate, and observation computation, then removes the orthogonal innovation. On Wan2.1, the parallel estimate retains its current-observation coefficient. The comparison therefore isolates the complementary contribution under the deployed parallel choice.

Combining observations with historical prediction. On Wan2.1, we fix the parallel prediction and compare raw and projected proxy injection (Table 6). With $\beta = 1$, the historical predictor has 31.78% median displacement NMSE; raw injection increases it to 76.28%, whereas projected injection reduces it to 24.53%. With an offline-optimal gain, projected injection gives 20.20% compared with 23.83% for raw injection. The same-cost ablation measures the complementary update. Appendix F reports the controlled comparisons.

5.5 Efficiency Analysis

On HunyuanVideo, the 17-call route achieves 2.784× measured speedup and 2.812× FLOPs reduction, including observation and reconstruction costs. Figure 6 compares reference fidelity at matched measured speedups, complementing

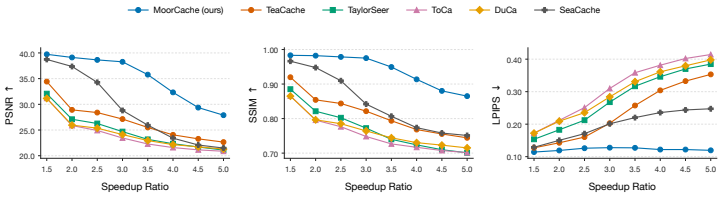

Figure 6: **Matched-speed curves on HunyuanVideo 13B.**

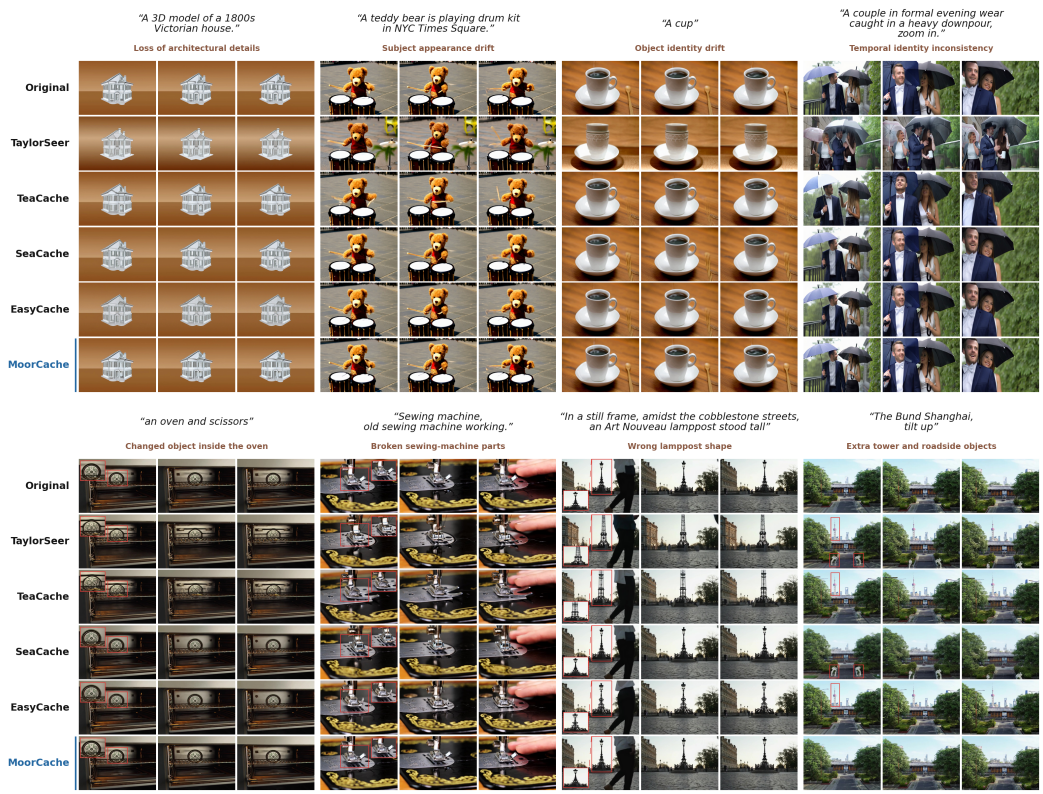

Figure 4: **Aligned video comparisons on Wan2.1 (top) and HunyuanVideo (bottom).**

Table 2. The curves trace the quality–latency trade-off across acceleration budgets, while the end-to-end timings demonstrate that current observations preserve substantial runtime savings.

On HunyuanVideo, the close agreement between measured and FLOPs-based speedups indicates that reduced arithmetic translates into wall-clock savings. On DiT, observation and reconstruction overhead creates a larger gap. Measured latency therefore captures the practical cost of each method beyond its reduction in arithmetic.

Within HunyuanVideo, moving from 17 to 9 calls reduces latency from 33.825 to 18.94 s, a further 44.0% reduction. VBench changes from 81.99 to 80.73, while PSNR decreases from 38.506 to 28.013 dB. The faster route therefore maintains aggregate video quality close to the dense reference, whose VBench is 80.98, while allowing larger deviations from its particular outputs. The 17-call route provides stronger reference fidelity; the 9-call route favors throughput. Figure 6 places these choices in the broader measured quality–latency trade-off.

The efficiency results and same-cost controls together characterize the use of the observation budget. Retaining observation computation while removing its complementary component lowers fidelity on all three evaluated backbones (Table 5). The quality gains therefore reflect how the observation is incorporated at a fixed cost. This connection between reconstruction quality and measured latency supports selecting an operating point according to the required fidelity and available runtime.

6 Conclusion

MoorCache connects a representation-based diagnosis of diffusion caching with current-observation reconstruction. Historical prediction propagates motion within the cached span, while current observations provide complementary information. Oracle diagnostics measure the residual motion that short-term history leaves unexplained. Same-cost ablations demonstrate the contribution of current-observation completion across image and video generation. With the historical prediction fixed, controlled diagnostics show the benefit of projection under both fixed and oracle gains. Across four backbones, the resulting framework improves quality–efficiency trade-offs with observation and reconstruction costs included. HunyuanVideo reaches $2.784\times$ measured speedup.

AI USE STATEMENT

In this work, we have not used generative AI tools for any of the tasks with required disclosure, including generating synthetic data sets, developing theoretical models or conceptual frameworks, formulating mathematical claims, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, or interpreting results; those tasks are not applicable to this work. Additionally, we used generative AI tools for the following tasks with recommended disclosure: creating and editing software code, and drafting parts of the manuscript. We have reviewed all AI-assisted work: AI-assisted code was reviewed and tested by the authors, and all AI-assisted text was reviewed and revised by the authors. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI. See the ICLR 2027 AI Policy for Authors for more details.

REFERENCES

Jiazi Bu, Pengyang Ling, Yujie Zhou, Yibin Wang, Yuhang Zang, Dahua Lin, and Jiaqi Wang. Dicache: Let diffusion model determine its own cache. In *International Conference on Learning Representations*, volume 2026, pp. 73778–73803, 2026.

Angela Castillo, Jonas Kohler, Juan C. Pérez, Juan Pablo Pérez, Albert Pumarola, Bernard Ghanem, Pablo Arbeláez, and Ali Thabet. Adaptive guidance: Training-free acceleration of conditional diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 1962–1970, 2025. doi: 10.1609/aaai.v39i2.32192.

Guantao Chen, Shikang Zheng, Yuqi Lin, and Linfeng Zhang. Forecast the principal, stabilize the residual: Subspace-aware feature caching for diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23632–23641, 2026a.

Pengtao Chen, Mingzhu Shen, Peng Ye, Jianjian Cao, Chongjun Tu, Christos-Savvas Bouganis, Yiren Zhao, and Tao Chen. Δ-DiT: Accelerating diffusion transformers without training via denoising property alignment. *International Journal of Computer Vision*, 134(6), 2026b. doi: 10.1007/s11263-026-02871-7. arXiv:2406.01125.

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

Bowen Cui, Yuanbin Wang, Huajiang Xu, Biaolong Chen, Aixi Zhang, Hao Jiang, Zhengzheng Jin, Xu Liu, and Pipei Huang. Denoising as path planning: Training-free acceleration of diffusion models with DPCache. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 43632–43642, 2026.

Haipeng Fang, Yu Li, Fan Tang, Yixing Lu, Juan Cao, and Sheng Tang. Resca: Residual caching for diffusion transformers acceleration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32957–32966, 2026.

Liang Feng, Shikang Zheng, Jiacheng Liu, Yuqi Lin, Qinming Zhou, Peiliang Cai, Xinyu Wang, Junjie Chen, Chang Zou, Yue Ma, and Linfeng Zhang. HiCache: A plug-in scaled-hermite upgrade for taylor-style cache-then-forecast diffusion acceleration. In *International Conference on Learning Representations*, 2026.

Charles William Gear and David R Wells. Multirate linear multistep methods. *BIT Numerical Mathematics*, 24(4):484–502, 1984.

Yasaman Haghighi and Alexandre Alahi. Sencache: Accelerating diffusion model inference via sensitivity-aware caching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14295–14304, 2026.

Shaoxuan He, Benlei Cui, Bukun Huang, Zhizeng Ye, Yunyun Sun, Longtao Huang, Hui Xue, Yang Yang, Haiwen Hong, Jingqun Tang, and Zhou Zhao. TC-Padé: Trajectory-consistent Padé approximation for diffusion acceleration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35768–35778, 2026.

Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. In *International Conference on Learning Representations*, 2025.

Rudolf Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

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.

Black Forest Labs. FLUX.1 [dev]: Open-weight text-to-image model. `https://huggingface.co/black-forest-labs/FLUX.1-dev`, 2024.

Jacques-Louis Lions, Yvon Maday, and Gabriel Turinici. Résolution d'edp par un schéma en temps "pararéel". *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 332(7):661–668, 2001.

Enhuai Liu, Yunke Wang, Changming Sun, and Chang Xu. D2cache: Second-order delta caching for higher video diffusion acceleration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 43589–43599, 2026.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025a.

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

David G Luenberger. Observing the state of a linear system. *IEEE Transactions on Military Electronics*, 8(2):74–80, 1964.

Zhengyao Lyu, Chenyang Si, Junhao Song, Zhenyu Yang, Yu Qiao, Ziwei Liu, and Kwan-Yee K. Wong. Fastercache: Training-free video diffusion model acceleration with high quality. In *International Conference on Learning Representations*, pp. 33132–33156, 2025.

Xinyin Ma, Gongfan Fang, Michael Bi Mi, and Xinchao Wang. Learning-to-cache: Accelerating diffusion transformer via layer caching. In *Advances in Neural Information Processing Systems*, 2024.

Zehong Ma, Longhui Wei, Feng Wang, Shiliang Zhang, and Qi Tian. Magcache: Fast video generation with magnitude-aware cache. In *Advances in Neural Information Processing Systems*, 2025. arXiv:2506.09045.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182, 2023. doi: 10.1109/ICCV51070.2023.00387.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

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

Zhirong Shen, Rui Huang, Jiacheng Liu, Chang Zou, Peiliang Cai, Shikang Zheng, Zhengyi Shi, Liang Feng, and Linfeng Zhang. Beyond fixed formulas: Data-driven linear predictor for efficient diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30792–30801, 2026.

Byunggwan Son, Jeimin Jeon, Jeongwoo Choi, and Bumsub Ham. Relational feature caching for accelerating diffusion transformers. In *International Conference on Learning Representations*, 2026.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Ye Tian, Xin Xia, Yuxi Ren, Shanchuan Lin, Xing Wang, Xuefeng Xiao, Yunhai Tong, Ling Yang, and Bin Cui. Training-free diffusion acceleration with bottleneck sampling. *arXiv preprint arXiv:2503.18940*, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Xin Zhou, Dingkang 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.

Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. In *International Conference on Learning Representations*, 2025.

Chang Zou, Shikang Zheng, Evelyn Zhang, Runlin Guo, Haohang Xu, Zhengyi Shi, Conghui He, Xuming Hu, and Linfeng Zhang. Rethinking token-wise feature caching: Accelerating diffusion transformers with dual feature caching. *IEEE Transactions on Image Processing*, 35:6211–6220, 2026. doi: 10.1109/TIP.2026.3698363.

A EXPERIMENTAL PROTOCOL

Model configurations. We evaluate Wan2.1-T2V-1.3B Wan et al. (2025) and HunyuanVideo 13B Kong et al. (2024) for text-to-video generation, FLUX.1-dev Labs (2024) for text-to-image generation, and DiT-XL/2 Peebles & Xie (2023) for class-conditional image generation. The dense references use 50 denoising steps. DiT-XL/2 uses DDIM-50 at 256×256 with CFG 1.5. MoorCache reconstructs skipped evaluations on the original solver grid using fixed, prompt-independent schedules; reduced-step baselines use the step counts listed in the tables.

Datasets and generation settings. For text-to-image generation, we evaluate 200 DrawBench prompts and generate FLUX images at 1024×1024. For video generation, we evaluate 946 VBench prompts with five random seeds per prompt, yielding 4,730 videos for each model and configuration. Each video contains 65 frames at 480p. For class-conditional generation, we generate 50,000 images across 1,000 ImageNet classes at 256×256.

Baselines and operating points. Across Tables 1–4, we evaluate reduced-step sampling, replay, and 14 caching baselines: TaylorSeer Liu et al. (2025b), TeaCache Liu et al. (2025a), EasyCache Zhou et al. (2025), SeaCache Chung et al. (2026), SenCache Haghighi & Alahi (2026), MagCache Ma et al. (2025), DiCache Bu et al. (2026), D2Cache Liu et al. (2026), DPCache Cui et al. (2026), ToCa Zou et al. (2025), DuCa Zou et al. (2026), FORA Selvaraju et al. (2024), Δ-DiT Chen et al. (2026b), and TC-Padé He et al. (2026). Each table specifies the methods evaluated on that backbone. Tables report each method's operating parameters and measured cost. For FLUX, EasyCache, TeaCache, and SeaCache thresholds are selected on a separate two-prompt screen. TC-Padé, EasyCache, SeaCache, and TeaCache are rerun under the stated DiT protocol. MoorCache's fixed schedules are shared across prompts and seeds; Appendix C specifies the observation configurations and call accounting.

Quality evaluation. Accelerated samples share conditioning, random seeds, and initial noise with their dense references. PSNR, SSIM, and LPIPS measure fidelity to these paired references. VBench evaluates video generation quality, ImageReward and CLIP score evaluate FLUX outputs, and FID, sFID, and Inception Score evaluate class-conditional ImageNet generation. Tables report both generation-quality and reference-fidelity metrics where applicable. The trajectory diagnostics in Section 3.5 use the diagnostic supports and event-level aggregation specified in Appendix D.

Efficiency evaluation. Experiments on the GBU cluster use NVIDIA H100 GPUs. We report measured latency and total FLOPs for each operating point, including current-observation computation, alignment, and residual reconstruction. Speedups are relative to the corresponding dense model. Native-call counts specify the schedule, while FLOPs and latency quantify its computational and runtime costs. Tables group moderate and stronger acceleration; Figure 6 additionally compares reference fidelity at matched speeds. Every plotted point is a measured operating point; connecting lines serve as visual guides. Results are point estimates unless an uncertainty analysis is reported.

Timing scope for Wan2.1 and FLUX. Wan2.1 and FLUX use BF16 Transformer inference and generate one sample at a time. Their latency measurements bracket the denoising loop with CUDA synchronization and a wall-clock timer. Text conditioning is prepared before timing; VAE decoding and output serialization follow the timed region. The recorded latency therefore measures denoising, including reconstruction work executed within that loop. Each prompt–seed pair contributes one timed generation in these evaluation runs.

B OBSERVATION IMPLEMENTATION AND INFERENCE

B.1 COST, PROVENANCE, AND SCOPE

Current-direction transfer adds residual projection and combination operations; temporal fallback adds two tensor differences, one inner product, one squared norm, and two linear combinations, all $O(d)$. Reductions use FP32 accumulation. The proxy is a fixed implementation choice rather than a separately optimized module, and its full computation, residual lifting, and all reconstruction operations are included in the reported FLOPs and measured latency.

All anchors lie on the accelerated trajectory, so the method improves online reconstruction rather than recovering an unavailable dense trajectory.

B.2 OBSERVATION STATE AND RECONSTRUCTION

For a cacheable target residual r_t, every reconstruction step invokes an observation adapter that returns

$$\mathfrak{O}_t = (o_t, o_a, \mathcal{L}_t, \widetilde{\alpha}_t, \beta_t), \qquad q_t = \mathcal{L}_t(o_t - o_a), \tag{15}$$

where $o_t - o_a$ is a current displacement, $\mathcal{L}_t$ maps it to the shape of r_t, and $(\widetilde{\alpha}_t, \beta_t)$ weight its components parallel and orthogonal to the latest target-residual secant. With the projector Π_a from Equation 7, the observation step is

$$\widetilde{r}_t = r_a + \widetilde{\alpha}_t s_a^r + \beta_t (I - \Pi_a) q_t. \tag{16}$$

This contract separates the common geometry from the architecture-specific source of current information. The decomposition is non-redundant: the span term represents motion aligned with the historical residual secant, while the innovation term represents only motion outside that span. Orthogonalization therefore separates representational roles rather than suppressing useful current information. The adapter also owns its reference-anchor update rule. The minimal current-input adapter is always available: it sets $\widetilde{\alpha}_t = \widetilde{\alpha}_t^h$ using Equation 8 and $\beta_t = 0$, so Equation 16 becomes temporal input-projection. HunyuanVideo and FLUX retain this input-derived parallel coefficient when adding the proxy correction. Wan2.1 and DiT-XL/2 instead use the observation-derived parallel coefficient when a conditional residual is evaluated, as specified below.

Algorithm 1 summarizes the resulting model-independent inference rule under a fixed schedule.

Algorithm 1 MoorCache inference with a current-observation adapter

Require: exact set $\mathcal{E}$, observation adapter $\mathfrak{O}$, observable inputs h_t
Ensure: target residuals $\widetilde{r}_t$

```
S ← ∅ {adapter state, including reference anchors}
for t = 0 to T − 1 do
    if t ∈ E then
        execute the target computation to obtain r_t
        update S with the exact target evaluation
        r̃_t ← r_t
    else
        obtain O_t from the current-observation adapter
        r̃_t ← Equation 16
        apply the adapter-specific reference update to S
    end if
    apply the model's standard guidance and scheduler update using r̃_t
end for
```

B.3 WAN2.1 AND DIT-XL/2

For explicit CFG, take the unconditional residual as the target and the current conditional residual as the observation. Let $p < t$ be the latest paired-exact step and define the held gap

$$g_t = r_t^u - r_t^c, \qquad g_p = r_p^u - r_p^c. \tag{17}$$

The adapter uses

$$o_t = r_t^c, \qquad o_p = r_p^c, \qquad \mathcal{L}_t = I, \qquad q_t = r_t^c - r_p^c, \qquad \beta_t = 1. \tag{18}$$

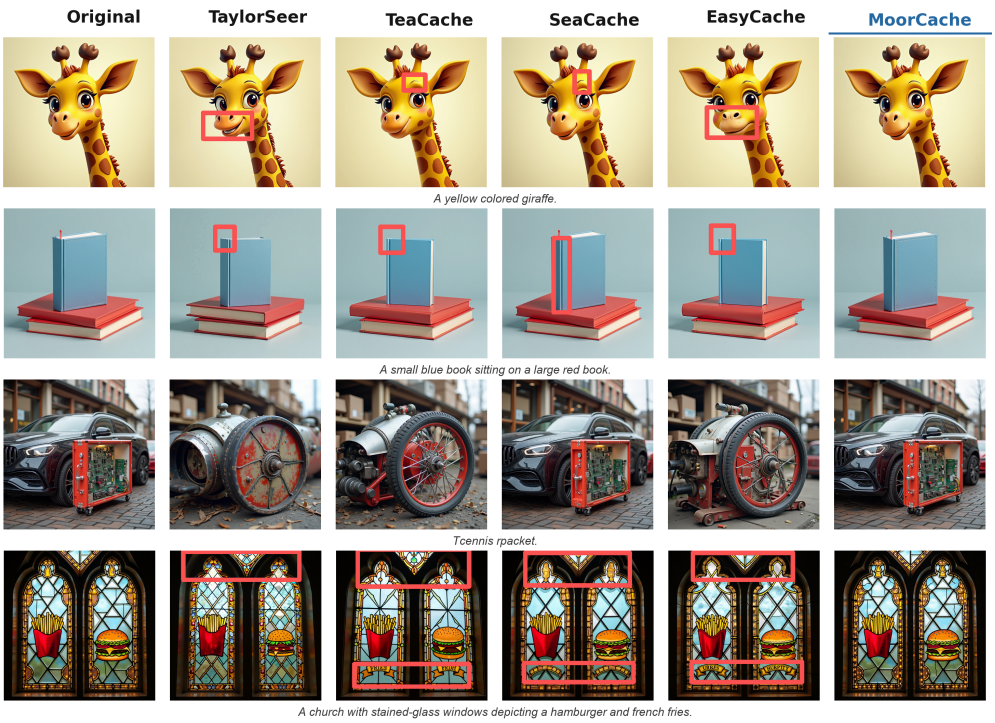

Figure 7: **FLUX.1-dev qualitative comparison (remaining prompts).** Same protocol as Figure 5; each row shows one prompt (labeled beneath the row), and red boxes mark localized deviations from the dense Original.

For any cached unconditional secant s_p^u, choose the parallel coefficient

$$\widetilde{\alpha}_t = \frac{\langle q_t, s_p^u \rangle}{\|s_p^u\|_2^2 + \varepsilon}. \tag{19}$$

The resulting target update uses the cached gap:

$$\widetilde{r}_t^u = r_t^c + g_p, \qquad r_t^u - \widetilde{r}_t^u = g_t - g_p. \tag{20}$$

The cached state is

$$\mathcal{A}_t^c = \{(h_b^c, r_b^c), (h_a^c, r_a^c)\}, \qquad g_p = r_p^u - r_p^c, \tag{21}$$

where conditional-only evaluations update $\mathcal{A}_t^c$, paired evaluations additionally update g_p, and current-input-only steps reconstruct r_t^c from $\mathcal{A}_t^c$ before adding g_p. In the unified target space, this is equivalent to using references $\bar{r}_a^u = r_a^c + g_p$ and $\bar{r}_b^u = r_b^c + g_p$: their secant is exactly $r_a^c - r_b^c$, so the generic temporal adapter reproduces the original conditional-reconstruction-plus-gap rule. Wan2.1 Wan et al. (2025) and DiT-XL/2 Peebles & Xie (2023) instantiate this adapter. Appendix F compares the deployed coefficient with an input-derived parallel coefficient while keeping the orthogonal term fixed.

B.4 HunyuanVideo and FLUX

For HunyuanVideo and FLUX, let ρ_t denote a lower-resolution current residual observation produced by the same stream, and let ρ_a be its value at the native anchor. The adapter is

$$o_t = \rho_t, \qquad o_a = \rho_a, \qquad q_t = \mathcal{L}_t(\rho_t - \rho_a), \qquad \widetilde{r}_t = r_a + \widetilde{\alpha}_t^h s_a^r + \beta_t (I - \Pi_a) q_t. \tag{22}$$

Here $\mathcal{L}_t$ performs the required spatial or token alignment, and β_t is frozen during evaluation. HunyuanVideo Kong et al. (2024) and FLUX use a lower-resolution pass from the same stream as the proxy observation. HunyuanVideo Kong et al. (2024) lifts the pass's final residual to the native token grid. Because FLUX caches double- and single-block residuals rather than only the final

Transformer residual, the same pass records the residual at each corresponding cache unit and lifts it to that unit's native token grid. The difference is therefore only the target granularity, not the observation mechanism. Proxy-only events do not replace native residual anchors, and all proxy computation is included in the reported FLOPs and measured latency.

We set $\beta_t = 1$ for these observation corrections. For HunyuanVideo Kong et al. (2024), the native latent is 60×80 tokens (480×640 pixels) and the proxy pass runs at 16×22 tokens (128×176 pixels); the latent downsampling and the residual lifting $\mathcal{L}_t$ both use trilinear interpolation with *align_corners*= False. For FLUX, the native latent is 1024×1024 pixels and the proxy pass runs at 256×256 pixels, i.e., $1/4$ per spatial dimension and $1/16$ of the area; the latent, the final output, and each cache unit's token-grid lifting use bilinear interpolation with *align_corners*= False.

C Backbone instantiations and fixed schedules

Common abstraction. For every backbone, MoorCache operates on a cacheable unit with observable input h_t, residual r_t, and a prompt-independent set of exact timesteps. Let $\mathcal{E}$ denote those timesteps. The temporal fallback is

$$\widetilde{r}_t = \begin{cases} r_t, & t \in \mathcal{E}, \\ r_a + \widetilde{\alpha}_t (r_a - r_b), & t \notin \mathcal{E}, \end{cases} \qquad \widetilde{\alpha}_t = \mathrm{clip}\left(\frac{\langle h_t - h_a, h_a - h_b \rangle}{\|h_a - h_b\|_2^2 + \varepsilon}, -2, 2 \right), \tag{23}$$

where $b < a < t$ are the two latest exact timesteps. Skip steps never update the native anchors. The observation rules in Equation 18 and Equation 22 augment this same one-dimensional reconstruction. HunyuanVideo Kong et al. (2024) and FLUX execute their proxy observation at every reconstructed timestep in the main-table routes.

Table 7: **MoorCache implementation by backbone.** Observation sources, cache locations, and alignment are specified for each model. Coefficient and anchor-update rules are detailed in Appendix B.

Backbone	Current observation	Cacheable unit	Alignment / parallel coefficient
Wan2.1 Wan et al. (2025)	Conditional residual	Full-Transformer residual	Identity / observation projection
DiT-XL/2 Peebles & Xie (2023)	Conditional residual	Per-block attention/MLP residuals	Identity / observation projection
HunyuanVideo Kong et al. (2024)	Low-resolution residual	Full-Transformer residual	Spatial lift / input projection
FLUX.1-dev Labs (2024)	Low-resolution block residuals	Double-/single-block residuals	Per-unit spatial lift / input projection

Schedule notation and call accounting. For Wan2.1 and DiT-XL/2, native calls count conditional and unconditional evaluations separately; for HunyuanVideo and FLUX, they count native-resolution exact evaluations. These native-call labels describe the schedule only: the main-table FLOPs and measured latency also include every proxy-observation and reconstruction operation. For each reported budget, exact evaluations are distributed at a fixed, approximately uniform interval over the 50 denoising timesteps, with endpoint adjustment only to meet the stated integer call count. Table 8 reports every budget used in the main tables; all remaining evaluations use Equation 23 augmented by Equation 7, or the observation-based reconstruction in Equation 11.

Table 8: **Fixed MoorCache interval schedules used in Tables 1–4.** Each schedule is determined only by its call budget and is shared by every prompt and seed.

Backbone / calls	Schedule	Native exact-call ratio
Wan2.1 Wan et al. (2025) / 40	Uniform fixed interval	$100/40 = 2.500\times$
Wan2.1 Wan et al. (2025) / 17	Uniform fixed interval	$100/17 = 5.882\times$
HunyuanVideo Kong et al. (2024) / 17	Uniform fixed interval	$50/17 = 2.941\times$
HunyuanVideo Kong et al. (2024) / 9	Uniform fixed interval	$50/9 = 5.556\times$
DiT-XL/2 Peebles & Xie (2023) / 28	Uniform fixed interval	$100/28 = 3.571\times$
DiT-XL/2 Peebles & Xie (2023) / 22	Uniform fixed interval	$100/22 = 4.545\times$
FLUX.1-dev Labs (2024) / 16	Uniform fixed interval	$50/16 = 3.125\times$
FLUX.1-dev Labs (2024) / 10	Uniform fixed interval	$50/10 = 5.000\times$

Fixed interval construction. MoorCache does not train, search, or calibrate a router. For every backbone and target call budget, we place exact evaluations at a simple, approximately uniform interval over denoising time and keep that schedule fixed for the entire benchmark. The schedule is independent of the evaluation prompt, seed, intermediate output, and quality score. Consequently, the reported gains measure the reconstruction rule under a transparent call budget rather than the benefit of a separately tuned routing policy.

D ADDITIONAL DISCOVERY DIAGNOSTICS

Measurement protocol. For Wan2.1 Wan et al. (2025), we record input and residual tensors on the diagnostic trajectories specified below. The coefficient diagnostic uses two seeds per prompt; the main video benchmark uses the five-seed protocol in Section 5. Because the two CFG branches receive the same module input, we retain one copy per event, and the retained events come from twice that many trajectories. Unless stated otherwise, reported values are event-level medians. We use $\varepsilon = 10^{-12}$ in projection denominators and 10^{-6} in normalized coverage denominators. Confidence intervals use cluster-bootstrap resamples over prompts; trajectory-clustered intervals are included as a sensitivity check.

Diagnostic targets and support. The combined-residual analysis evaluates the CFG-weighted denoising residual. The coefficient diagnostic pools 2,024 branch-level events from 22 prompts, two seeds, 23 skip steps, and two streams. The eight-secant comparison restricts both bases to events with all eight historical transitions available. The projection control in Appendix F evaluates 506 unconditional-residual events from 22 prompts with the historical prediction fixed. Each analysis uses its own target displacement for normalization and reports eventwise statistics aggregated over its stated support.

Secant locality and stagewise coverage. The latest input secant explains a median 98.9% of input-displacement energy. Per-step medians range from 93.8% to 99.9%, and the early, middle, and late-stage medians are 99.5%, 98.9%, and 95.8%, respectively. This predictability is local: replacing the latest secant by the second-most-recent one lowers overall coverage to 90.8% and late-stage coverage to 71.8%. The latest secant wins at every measured event, with a median advantage of 7.9 percentage points.

Per-branch residual coverage. When the coverage operator is applied to a single CFG branch in isolation (the unconditional branch unless noted), the residual follows a different pattern from the input: its median coverage is 64.8%, with a prompt-clustered 95% interval of $[58.1, 70.0]\%$ and a trajectory-clustered interval of $[61.0, 67.9]\%$. Stagewise medians rise from 13.8% early to 66.8% in the middle and 92.1% late. Input coverage exceeds residual coverage at 98.6% of events, showing that predictable progress in the observable input does not make the historical residual direction reliable, especially early in denoising. These per-branch values serve as the baseline for the deployed-combined-residual measurement reported next; the main-text finding uses the combined residual.

Combined-residual (final) coverage. Because the deployed denoising update uses the CFG-weighted residual $r_t = r_t^u + w\,(r_t^c - r_t^u)$ with $w = 5$, we also measure the secant coverage of this combined residual, recording per-step cond/uncond residual tensors along dense trajectories under the same protocol. The conditional and unconditional displacements are highly aligned (median mutual coverage 74.8%, i.e., the conditional displacement explains 74.8% of the unconditional displacement energy), consistent with both branches sharing the same noisy latent and timestep. The combined displacement $d_t = d_t^u + w\,(d_t^c - d_t^u)$ is therefore dominated, at $w = 5$, by the amplified cross-branch gap motion; the gap motion itself has very low median secant coverage (1.3%–9.7%), so the combined displacement is nearly orthogonal to its historical secant. The median combined-residual coverage is 2.72% (per-prompt median 1.96%, per-prompt range $[0.63\%, 6.61\%]$; prompt-clustered 95% interval $[1.82\%, 3.77\%]$). A complementary diagnostic recombines saved conditional and unconditional outputs along a fixed trajectory, rebuilding the displacement and historical secant for each weight. Median coverage is 65.6%, 1.63%, and 0.75% at $w = 1, 5, 7.5$, respectively, illustrating how CFG weighting changes the geometry of residual motion along the recorded trajectory. The event distribution is strongly right-skewed (90th percentile $\approx 40\%$), so the historical direction is uninformative at most events but not all; the direction bottleneck thus holds for the deployed combined residual, not merely for either CFG branch in isolation.

Intermediate-feature coverage. Feature-extrapolation methods such as TaylorSeer Liu et al. (2025b) cache and extrapolate intermediate block features rather than the final residual. To check whether the direction bottleneck extends to their extrapolation targets, we repeat the coverage measurement on block-prefix features at depths 1, 2, 4, and 8 (the input to the corresponding block prefix) over the same full diagnostic protocol. Median coverages are $69.8\%/69.9\%/70.0\%/69.9\%$ (cond) and $67.2\%/67.1\%/67.3\%/69.9\%$ (uncond) across depths 1/2/4/8, essentially identical to the single-branch final-residual value on the same protocol (68.9% cond) rather than to the CFG-weighted combined coverage (2.72%). These feature-level measurements show incomplete historical coverage, but the deficit is smaller than for the CFG-weighted final residual. The fixed-span bound in §3.5 applies separately to each target and does not equate their coverage values.

Oracle multi-secant historical span. The one-secant result could in principle reflect an insufficiently rich history rather than the need for a current observation. We therefore allow the temporal baseline to access the latest eight exact residual transitions as an idealized offline predictor. Let

$$y_t = r_t - r_a, \qquad V_K = [r_a - r_b,\ r_b - r_c, \ldots], \qquad C_K = \frac{\|P_{\mathrm{span}(V_K)} y_t\|_2^2}{\|y_t\|_2^2}.$$

Here C_K uses offline least-squares coefficients and is thus a representation upper bound, not a deployable predictor. We compare only the events from prompt–seed groups for which all eight transitions are available. On this common support, median coverage increases from $C_1 = 77.80\%$ to only $C_8 = 78.95\%$; the median paired gain is 0.70 percentage points. Relative to the energy missed by the latest secant, the idealized eight-secant span recovers

$$R_8 = \underset{e}{\mathrm{median}}\ \frac{C_8(e) - C_1(e)}{1 - C_1(e)} = 3.49\%$$

at the median, and 8.67% at the 90th percentile. Because medians and ratios do not commute, the ratio of the quoted medians, $(78.95 - 77.80)/(100 - 77.80) = 5.18\%$, is larger than the median eventwise recovery; both figures leave the conclusion unchanged. The conditional value $C_1 = 77.80\%$ is higher than the all-event value 64.8% above because eight-transition availability excludes early events; all comparisons here use the same eligible-event set. Since the two- and three-secant spaces are nested within this eight-secant space, this stronger idealized control also upper-bounds their attainable representation gains event by event. Thus the limitation is not merely the use of one latest secant: a substantially larger short-history span still leaves nearly all of its missing directional energy unexplained. This result does not rule out arbitrarily long or nonlinear history models; it isolates the limitation of short linear historical spans studied here.

Coefficient diagnostics. The input-derived coefficient α_t^h closely tracks the optimal residual coefficient $\alpha_t^{r\star}$ along each trajectory: the median per-trajectory Spearman correlation is 0.980, with prompt- and trajectory-clustered 95% intervals of $[0.973, 0.984]$ and $[0.975, 0.983]$, respectively, and a minimum of 0.751. This correlation is partly driven by their shared denoising-time trend: it is 0.35 after detrending, 0.12 for first differences, and 0.426 across samples at fixed steps. Nevertheless, coefficient choice has little effect on reconstruction error. On 2,024 unique events (22 prompts, two seeds, 23 skip steps, and two streams), median displacement NMSE is 100% for replay, 35.771% for input projection, 35.167% for the optimal residual coefficient, 35.747% for the RFC norm ratio, and 37.302% for time interpolation. With $m_j = \mathrm{median}_t\, E_t^{(j)}$, the recovered improvement is $\eta = (m_{\mathrm{replay}} - m_{\mathrm{input}})/(m_{\mathrm{replay}} - m_{\mathrm{oracle}}) = 99.07\%$. This is a ratio of median-error reductions, not the median of eventwise recovery ratios.

Error normalization and aggregation. The coefficient and conditional-direction diagnostics use $E_t = \|r_t - \widehat{r}_t\|_2^2 / \max(\|r_t - r_a\|_2^2, \varepsilon)$, normalized by target displacement energy. Replay has $\widehat{r}_t = r_a$, hence $E_t = 1$ away from the numerical floor. The coefficient records store residual-relative L2 errors $\ell_t = \|r_t - \widehat{r}_t\|_2 / \|r_t\|_2$. We convert each event using $E_t = (\ell_t / \ell_t^{\mathrm{replay}})^2$ before taking medians; neither squaring an aggregate percentage nor dividing aggregate medians performs this conversion. Coverage in Equation 3 measures represented displacement energy and remains a separate quantity. A difference of median errors and a median paired error reduction are also reported distinctly.

Cross-backbone replication. We repeat the analysis on HunyuanVideo Kong et al. (2024) using the same complete evaluation protocol, 65 frames, and 50 denoising steps. Median input coverage remains high at 81%, though below Wan2.1 Wan et al. (2025)'s 99%, whereas the deployed

residual's median coverage drops to 2.5%, comparable to the 2.7% combined-residual value on Wan2.1 Wan et al. (2025) (both measurements concern deployed residuals rather than isolated CFG branches). On the 506 recorded events, median displacement NMSE is 100% for replay, 111.020% for time interpolation, 109.849% for the RFC norm ratio, 107.775% for input projection, and 97.715% for the optimal coefficient. Under this common normalization, the observable coefficient choices do not improve on replay; only the offline-optimal coefficient reduces median error. The correlation between the input and optimal coefficients is also weak there (Spearman -0.03, versus 0.98 on Wan2.1 Wan et al. (2025)). End to end, moving from replay to residual-secant extrapolation improves PSNR by only 0.95 dB on HunyuanVideo Kong et al. (2024), versus 2.2 dB on Wan2.1 Wan et al. (2025). These results reinforce that the historical residual direction, not merely its coefficient, is the unstable component.

D.1 Matched-route end-to-end direction ablation

The decisive end-to-end control keeps the exact schedule, anchors, current-observation computation, and all other hyperparameters fixed. It removes only the out-of-span innovation by setting $\beta_t = 0$. Thus both rows have identical cost, and their quality difference isolates the contribution of $(I - \Pi_a)q_t$ in Equation 11.

Table 9: **Same-route out-of-span ablation.** "No out-of-span correction" sets $\beta_t = 0$; the schedule, anchors, current-observation computation, and all other settings are unchanged.

Backbone / route	Reconstruction	PSNR ↑	SSIM ↑	LPIPS ↓
Wan2.1 Wan et al. (2025) / 40 calls	No out-of-span correction ($\beta_t = 0$)	31.342	0.9211	0.1103
	MoorCache full reconstruction	34.022	0.9616	0.0188
HunyuanVideo Kong et al. (2024) / 17 native calls	No out-of-span correction ($\beta_t = 0$)	36.132	0.9151	0.2139
	MoorCache full reconstruction	38.506	0.9787	0.1278
FLUX.1-dev Labs (2024) / 16 native calls	No out-of-span correction ($\beta_t = 0$)	22.145	0.8246	0.1886
	MoorCache full reconstruction	32.124	0.9615	0.0766

The out-of-span innovation improves all three reference-fidelity metrics on every backbone. Because both rows retain the same current-observation computation, these gains isolate how the observation is used rather than purchasing additional compute.

Conditional-observation coverage on Wan2.1. We first test whether the current conditional displacement contains information unavailable to the old unconditional secant. Let $S_a^u = \mathrm{span}(s_a^u)$, and let $C_t(S) = \|P_S d_t^u\|_2^2 / \|d_t^u\|_2^2$ measure the eventwise coverage of the target unconditional displacement by a subspace S. For each event, define the fraction of historically missing energy recovered after adding the conditional displacement as

$$R_{\mathrm{CFG}}(t) = \frac{C_t(S_a^u + \mathrm{span}(d_t^c)) - C_t(S_a^u)}{1 - C_t(S_a^u)}. \tag{24}$$

The median of $R_{\mathrm{CFG}}(t)$ is 40.69%, with a prompt-clustered 95% interval of $[38.28, 43.47]\%$, and prompt-level median recovery is positive throughout the evaluation set. Separately, the event-level median historical and augmented coverages are 68.87% and 80.73%, respectively. These unconditional-residual values differ from the earlier all-residual coverage because they use the unconditional stream on the paired-event support. This is an idealized representation test: it establishes that conditional motion supplies a complementary direction, rather than merely another coefficient on the historical secant.

We then evaluate the deployable reconstruction in Equation 20 against the offline-optimal scalar coefficient on the old unconditional secant. Despite granting the temporal baseline its best possible coefficient, current-direction reconstruction lowers median displacement NMSE from 31.13% to 23.62%. The median paired reduction is 5.03 percentage points, with a prompt-clustered 95% interval of $[3.74, 6.48]$, and it wins in the overwhelming majority of events and in median throughout the prompt set. The benefit is largest early: stagewise displacement NMSE changes from 76.34% to 39.55%, from 28.71% to 23.41%, and from 6.90% to 4.14% in the early, middle, and late stages, respectively.

E PER-PROMPT FIDELITY DISTRIBUTIONS

Figure 8 complements the aggregate Wan2.1 Wan et al. (2025) results in Table 1 with paired distributions over the complete evaluation prompt set.

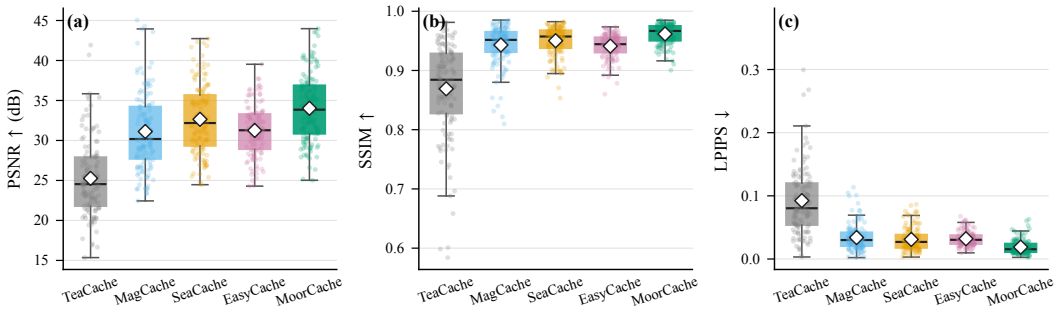

Figure 8: **Per-prompt reference fidelity on Wan2.1.** Each point is one evaluation prompt; boxes summarize distributions and diamonds mark means.

F CONTROLLED ANALYSES OF RECONSTRUCTION COMPONENTS

This appendix isolates the contributions of the parallel coefficient, reconstruction rule, and observation projection under controlled evaluation settings.

F.1 Parallel-coefficient ablation on Wan2.1. The deployed Wan2.1 reconstruction uses the observation-derived parallel coefficient (Equation 19). An alternative keeps the input-projected parallel term and adds only the out-of-span component,

$$\widetilde{r}_t^{u,\text{strict}} = r_p^u + \widetilde{\alpha}_t^h s_p^u + (I - \Pi_p)(r_t^c - r_p^c).$$

We evaluate both coefficient choices on the Wan2.1 40-call route (Table 11). The input-derived coefficient gives 27.836 dB PSNR, 0.8922 SSIM, and 0.0848 LPIPS at $2.390\times$ measured speedup, versus 34.022 dB, 0.9616, and 0.0188 at $2.486\times$ for the deployed observation-derived coefficient. Both use the same out-of-span term $(I-\Pi_p)(r_t^c - r_p^c)$; the 6.186 dB difference therefore isolates the parallel-term choice. This result supports using the observed parallel displacement on this target. The cached-gap update in Appendix B implements this observation-derived choice. The control in F.3 instead tests adding an observation to a fixed historical prediction.

Table 11: **F.1: Parallel-coefficient ablation on the Wan2.1 40-call route.** Both rows share the same route and full evaluation protocol. The out-of-span term is identical in both variants; the comparison isolates the parallel-term choice.

Method	Calls	PSNR ↑	SSIM ↑	LPIPS ↓	Measured speedup ↑
Strict out-of-span	40	27.836	0.8922	0.0848	$2.390\times$
Deployed held-gap (MoorCache)	40	**34.022**	**0.9616**	**0.0188**	$2.486\times$

F.2 Existing reconstructor plus held gap. Under the same 40-call Wan2.1 Wan et al. (2025) route, MoorCache exceeds TaylorSeer with held-gap reconstruction by 2.887 dB PSNR, 0.0205 SSIM, and 0.0104 lower LPIPS (Table 12). TaylorSeer (N=2, O=1) reconstructs the conditional stream and uses the same held unconditional gap. The shared route and compute budget isolate the reconstruction choice, with measured speedups of $2.486\times$ for MoorCache and $2.406\times$ for the control.

Table 10: **VBench metrics in Wan2.1 1.3B.**

Models	Subject Consistency	Background Consistency	Temporal Flickering	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Imaging Quality	Object Class
SenCache Haghighi & Alahi (2026)	96.21%	98.47%	99.45%	98.40%	62.50%	63.07%	63.74%	75.00%
MagCache Ma et al. (2025) (e=0.06, k=2)	96.16%	98.48%	99.43%	98.41%	62.50%	63.04%	63.67%	75.78%
TaylorSeer Liu et al. (2025b) (N=2, O=1)	96.73%	98.96%	99.48%	97.99%	62.50%	64.43%	60.93%	62.50%
TeaCache Liu et al. (2025a) (ℓ=0.08)	96.07%	98.72%	99.43%	98.36%	62.50%	62.58%	61.90%	75.78%
DiCache Bu et al. (2026) (τ=0.08)	96.02%	98.53%	99.47%	98.38%	62.50%	63.15%	63.94%	75.00%
Reduced steps (25 NFE)	96.64%	98.88%	99.44%	98.09%	62.50%	63.46%	60.48%	63.28%
SeaCache Chung et al. (2026) (ℓ=0.20)	96.33%	98.46%	99.35%	**98.46%**	62.50%	62.86%	63.70%	75.00%
D2Cache Liu et al. (2026) (τ=0.10)	96.23%	98.34%	99.46%	98.35%	50.00%	62.87%	62.97%	75.00%
EasyCache Zhou et al. (2025) (τ=0.027)	96.18%	98.30%	99.40%	98.40%	62.50%	62.66%	63.67%	75.00%
MoorCache (40 calls, ours)	**96.73%**	**98.96%**	**99.48%**	97.28%	**62.50%**	**64.43%**	**64.29%**	**75.78%**

Models	Multiple Objects	Human Action	Color	Spatial Relationship	Scene	Temporal Style	Appearance Style	Overall Consistency
SenCache Haghighi & Alahi (2026)	52.34%	62.50%	**85.71%**	86.75%	12.50%	22.43%	18.62%	25.94%
MagCache Ma et al. (2025) (e=0.06, k=2)	52.34%	62.50%	82.14%	**87.50%**	12.50%	22.46%	18.60%	25.97%
TaylorSeer Liu et al. (2025b) (N=2, O=1)	71.88%	**75.00%**	66.67%	75.76%	12.50%	22.94%	20.01%	26.00%
TeaCache Liu et al. (2025a) (ℓ=0.08)	50.78%	62.50%	79.46%	83.85%	17.19%	22.42%	18.52%	25.95%
DiCache Bu et al. (2026) (τ=0.08)	50.00%	50.00%	**85.71%**	**87.50%**	12.50%	22.33%	18.90%	26.03%
Reduced steps (25 NFE)	**73.44%**	**75.00%**	64.58%	78.76%	12.50%	22.65%	19.71%	25.73%
SeaCache Chung et al. (2026) (ℓ=0.20)	53.12%	62.50%	82.14%	**87.50%**	12.50%	22.47%	18.51%	25.99%
D2Cache Liu et al. (2026) (τ=0.10)	53.91%	62.50%	74.11%	69.12%	12.50%	22.44%	18.88%	25.67%
EasyCache Zhou et al. (2025) (τ=0.027)	52.34%	62.50%	**85.71%**	86.77%	12.50%	22.53%	18.74%	26.08%
MoorCache (40 calls, ours)	50.00%	62.50%	79.46%	78.91%	**25.00%**	**22.94%**	**20.01%**	**26.31%**

Table 12: **F.2: TaylorSeer plus held gap on the Wan2.1 40-call route.** Both rows share the same route, compute budget, and the full evaluation protocol of Table 1.

Method	Measured speedup ↑	PSNR ↑	SSIM ↑	LPIPS ↓
TaylorSeer + held gap	2.406×	31.135	0.94113	0.02921
MoorCache	2.486×	**34.022**	**0.9616**	**0.0188**

F.3 Projection ablation. This control isolates the contribution of orthogonal projection to current-observation completion. On the Wan2.1 Wan et al. (2025) diagnostic route, we compare raw injection $\beta_t q_t$ against projected injection $\beta_t(I - \Pi_a)q_t$, under both the deployed frozen gain ($\beta_t = 1$) and an offline-optimal per-event gain. Table 13 reports median displacement NMSE over 506 events from 22 prompts, normalized by $\|r_t^u - r_a^u\|_2^2$.

Table 13: **F.3: Projection ablation on Wan2.1.** Median displacement NMSE under frozen (β=1) and offline-optimal gains; the historical span only reconstruction has no injection term, so no offline-optimal gain is defined for it.

Reconstruction	$\beta = 1$	Oracle β
Historical span only	31.78%	–
Raw injection $\beta_t q_t$	76.28%	23.83%
Projected injection $\beta_t(I - \Pi_a)q_t$	**24.53%**	**20.20%**

Projection is decisive in both gain settings: with $\beta_t = 1$ it reduces median displacement NMSE from 31.78% (historical) to 24.53%, whereas raw injection *more than doubles* the error to 76.28%; with an offline-optimal gain, projected injection reduces median displacement NMSE from 23.83% to 20.20% (a 3.63-point difference of medians; median paired reduction 3.06 points, cluster-bootstrapped 95% CI $[2.1, 4.1]$). The projected variant wins at every prompt in both settings, and at 96.84% of events with the frozen gain and 86.76% with the offline-optimal gain. The raw-injection error is consistent with overlap between observation and history: the proxy displacement is 66.3% parallel to the historical secant, so without orthogonalization the historical direction is double-counted, amplifying rather than correcting the extrapolation error. With the historical prediction fixed, projected injection achieves lower displacement NMSE than raw injection under both fixed and oracle gains on this diagnostic support.

G COMPARISON WITH RELATED CACHING METHODS

Table 14 compares historical modeling and the use of current-step observations across related caching methods.

Table 14: **Comparison of caching-method capabilities.** A checkmark denotes a supported capability; a cross denotes its absence.

Method	Historical trajectory modeling	Current-step information	Current residual / feature observation	Explicit complementary direction	General beyond CFG redundancy	Single-stream low-res observation
TaylorSeer Liu et al. (2025b)	✓	×	×	×	✓	×
SeaCache Chung et al. (2026)	✓	✓	×	×	✓	×
RFC Son et al. (2026)	✓	✓	×	×	✓	×
FasterCache Lyu et al. (2025)	✓	✓	✓	×	×	×
MoorCache	✓	✓	✓	✓	✓	✓