ACCELERATING DIFFUSION MODELS WITH PHYSICAL VELOCITY EVOLUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion and flow-based generative models deliver strong visual synthesis quality but incur substantial inference cost during iterative sampling. Existing caching methods typically rely on high-dimensional feature differences that can entangle content and noise, overlooking the physical-time velocity dynamics underlying the generative ODE. These representation-level differences do not directly characterize the physical evolution of the latent state, which is governed by a velocity field over continuous sampling time. Instead, grounding cache reuse in physical-time velocity dynamics connects approximation errors to the latent updates they perturb, with the actual physical time intervals determining how velocity errors translate into trajectory deviations. Based on this observation, we introduce Trajectory-Aligned Flow Caching (TAFC), a training-free framework that operationalizes physical-time velocity dynamics through three complementary components. Physical-Time velocity Extrapolation predicts upcoming latent updates by extrapolating velocity dynamics over physical sampling intervals. Curvature-Aware Reuse Gating uses velocity curvature to assess whether the observed dynamics support continued local extrapolation. Self-Calibrating Error Feedback measures a posteriori approximation errors at refresh steps and adapts reuse tolerances, calibrating physical trajectory predictions against model-evaluated updates. Together, these components connect physical-time velocity dynamics, prediction reliability, and online error correction in a closed-loop caching framework. Extensive experiments on large-scale image and video generative models demonstrate that TAFC achieves favorable efficiency–fidelity trade-offs over existing training-free caching methods.

1 INTRODUCTION

Turning noise into a detailed image or coherent video requires diffusion and flow-based models to advance a latent state through physical sampling time Ho et al. (2020); Song et al. (2020b); Lipman et al. (2022); Liu et al. (2022b). With large Transformer backbones, repeated model evaluations make this process expensive Peebles & Xie (2023); Esser et al. (2024). Caching reduces redundant computation without retraining.

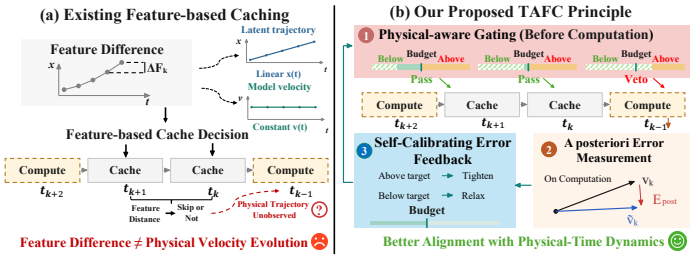


Figure 1: **Motivation and conceptual overview of TAFC. (a)** Feature differences and predictions provide indirect evidence of physical-time velocity dynamics and may not reflect the trajectory perturbation caused by cache reuse. **(b)** TAFC connects reuse decisions to physical-time dynamics through residual extrapolation and reliability assessment, then uses approximation errors observed at model refreshes to calibrate subsequent reuse tolerances.

Existing caching approaches instead assess reuse through internal representations or other surrogate signals. TeaCache Liu et al. (2025a) and SeaCache Chung et al. (2026) derive refresh signals from feature changes, while TaylorSeer Liu et al. (2025b) extrapolates recent feature trends. Other approaches characterize reuse through interval-average velocity,

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.

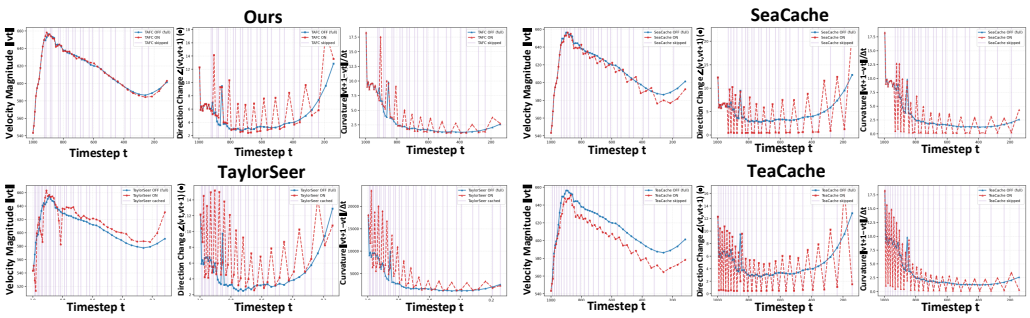


Figure 2: **Velocity dynamics under caching.** The shown magnitude, direction-change, and curvature traces averaged over ten samples compare TAFC with SeaCache, TaylorSeer, and TeaCache. TAFC more closely follows the full-evaluation dynamics in these examples.

output sensitivity, or higher-order residual deltas Gao et al. (2026); Haghighi & Alahi (2026); Liu et al. (2026). Yet these signals do not directly reflect how the latent state evolves over physical sampling time. Thus, representation-level variation can be misaligned with the actual trajectory perturbation caused by cache reuse.

We bridge this gap through the velocity field of the generative ODE. At a fixed state and time, an Euler update converts velocity error into latent displacement error through multiplication by the physical time interval. Equal prediction errors therefore have different consequences over unequal intervals. Modeling physical-time velocity dynamics makes this dependence explicit: a cached prediction should track the latent state's evolution over the actual sampling schedule. Figure 1 motivates this connection between physical dynamics and reuse reliability.

We develop *Trajectory-Aligned Flow Caching (TAFC)* to turn this physical-time perspective into training-free cache decisions. Its three components address prediction along physical sampling time, reliability of the extrapolated dynamics, and correction using observed errors.*Physical-Time Velocity Extrapolation* starts from two evaluated velocities and extends their trend over the physical interval to the current step. This respects nonuniform time spacing rather than treating all iterations as equally separated. The extrapolated velocity directly determines how the cached computation affects the physical evolution of the latent state.A physical-time prediction remains useful only while its local dynamics remain reliable. *Curvature-Aware Reuse Gating* examines velocity magnitude and directional variation through a curvature proxy. Together with the extrapolated correction, this signal tests whether the available velocity history supports extending the physical-time trend or calls for a fresh evaluation.

At a refresh, the model supplies an observed velocity against which the physical-time extrapolation can be checked. *Self-Calibrating Error Feedback* uses this a posteriori approximation error to adjust reuse tolerances relative to a warm-up-calibrated target. Above budget tighten subsequent reuse; Below budget allow more permissive tolerances. Physical trajectory predictions are thereby calibrated through model-evaluated velocities, without evaluating skipped steps.

TAFC thus couples physical-time dynamics with reliability assessment and online correction in a closed loop. Evaluations on multiple visual generative models show that this design preserves reconstruction fidelity and generation quality while reducing computation. Comparisons with existing training-free caching methods support its efficiency–fidelity benefits, while Figure 2 illustrates closer tracking of the reference physical-time velocity dynamics.

Our contributions are threefold:

- We ground cache reuse in physical-time velocity dynamics, relating approximation errors to trajectory deviations through actual physical sampling intervals.
- We introduce TAFC, integrating Physical-Time Velocity Extrapolation, Curvature-Aware Reuse Gating, and Self-Calibrating Error Feedback into a training-free closed-loop framework.
- We demonstrate favorable efficiency–fidelity trade-offs against existing training-free caching methods on large-scale visual generative backbones, supported by component ablations and refresh-pattern analysis.

2 Related Work

GENERATIVE MODEL ACCELERATION. Acceleration methods reduce either the number of sampling steps or the cost of each step. Numerical approaches such as DDIM Song et al. (2020a), PNDM Liu et al. (2022a), DEIS Zhang & Chen (2022), DPM-Solver Lu et al. (2022), DPM-Solver++ Lu et al. (2025a), and UniPC Zhao et al. (2023) improve sampling under limited evaluation budgets; EDM Karras et al. (2022) studies sampler design, while GENIE Dockhorn et al. (2022) exploits higher-order derivatives. Related efforts have also explored parallel solvers, efficient trajectory optimization, diffusion-based restoration, and event-driven visual modeling Lu et al. (2025b); Lu et al.; Zhu et al. (2025); Hou et al. (2026); Zhou et al. (2026); Chen et al. (2026). These methods address numerical integration, but aggressive step reduction can still trade fidelity for speed. Distillation and consistency-based methods, including progressive distillation Salimans & Ho (2022), improved consistency training Song & Dhariwal (2024), Latent Consistency Models Luo et al. (2023), and Adversarial Diffusion Distillation Sauer et al. (2024), enable few-step generation through training. Structural methods such as Token Merging Bolya et al. (2022) instead reduce per-evaluation computation. TAFC focuses on training-free caching: it retains the sampling grid and selectively substitutes predicted updates for neural evaluations, making reuse decisions aligned with the physical-time evolution of the generative ODE.

CACHING-BASED ACCELERATION. Caching exploits redundancy between denoising steps at different levels. DeepCache Ma et al. (2024b) reuses high-level U-Net features, and Block Caching Wimbauer et al. (2024) selects reusable network blocks. For transformers, Learning-to-Cache Ma et al. (2024a), FORA Selvaraju et al. (2024), SmoothCache Liu et al. (2025c), ToCa Zou et al. (2024), and Δ-DiT Chen et al. (2024) explore layer, attention/MLP, or token-level reuse. PAB Zhao et al. and FasterCache Lyu et al. (2025) additionally exploit video-specific redundancy. TeaCache Liu et al. (2025a) and SeaCache Chung et al. (2026) use lightweight feature-derived signals to guide refresh decisions, whereas TaylorSeer Liu et al. (2025b) forecasts features using temporal finite differences. These approaches motivate two distinct design questions: how to approximate the next computation, and how to assess the risk of using that approximation. TAFC uses Physical-Time velosity Extrapolation for the former and combines Curvature-Aware Reuse Gating with Self-Calibrating Error Feedback for the latter. This closed-loop design connects approximation, reuse reliability, and error correction directly to the physical-time trajectory followed.

3 Method

3.1 Residual Caching Formulation

Our goal is to preserve the physical evolution of the generative trajectory, which is governed by the model velocity over sampling time. However, the quantity directly manipulated by caching is the computation reused inside the Transformer. Let $\mathbf{h}(t)$ denote the hidden representation at physical sampling time t, and let $\mathcal{F}_\theta$ denote the mapping implemented by the cached Transformer layers. We define the corresponding residual as

$$\mathbf{r}(t) = \mathcal{F}_\theta(\mathbf{h}(t), t) - \mathbf{h}(t). \tag{1}$$

Hence, a full evaluation gives $\mathcal{F}_\theta(\mathbf{h}(t), t) = \mathbf{h}(t) + \mathbf{r}(t)$. TAFC therefore models this reusable residual as the operational quantity for preserving the velocity-driven physical trajectory.

Conventional residual caching reuses the most recently evaluated residual. If t_b denotes the latest sampling time at which the Transformer blocks are fully evaluated, then a cached step uses

$$\widehat{\mathbf{r}}(t) = \mathbf{r}(t_b), \qquad t > t_b. \tag{2}$$

This rule implicitly treats the residual as locally constant over sampling time. Feature-based criteria such as those used by TeaCache and SeaCache determine when reuse is permitted, but the cached approximation itself remains the latest evaluated residual.

TAFC instead models how $\mathbf{r}(t)$ evolves along physical sampling time. The two most recent model-evaluated residuals estimate its local temporal variation, which is then extrapolated to subsequent cached steps. Thus, rather than directly modeling the latent trajectory, TAFC aligns the reusable Transformer computation with its physical-time evolution by replacing the zero-order residual approximation with a first-order temporal model.

Algorithm 1 Trajectory-Aligned Flow Caching

Require: Sampling schedule $\{t_k\}_{k=0}^{N}$, Transformer mapping $\mathcal{F}_\theta$, and refresh indicator $\mathbb{I}_{\text{ref}}(t_k) \in \{0, 1\}$.

Fully evaluate $\mathcal{F}_\theta$ at two initial sampling steps t_a t_b to obtain $\mathbf{r}(t_a)$ $\mathbf{r}(t_b)$.
for each subsequent sampling timestep t_k **do**
  Obtain the current hidden representation $\mathbf{h}(t_k)$.
  $\mathbf{d}_b \leftarrow \big(\mathbf{r}(t_b) - \mathbf{r}(t_a)\big)/|t_b - t_a|$
  $\widehat{\mathbf{r}}(t_k) \leftarrow \mathbf{r}(t_b) + \mathbf{d}_b|t_k - t_b|$
  if $\mathbb{I}_{\text{ref}}(t_k) = 1$ **then**
    $\mathbf{h}^{\text{out}}(t_k) \leftarrow \mathcal{F}_\theta\big(\mathbf{h}(t_k), t_k\big)$
    $\mathbf{r}(t_k) \leftarrow \mathbf{h}^{\text{out}}(t_k) - \mathbf{h}(t_k)$
    $(t_a, \mathbf{r}(t_a)) \leftarrow (t_b, \mathbf{r}(t_b))$
    $(t_b, \mathbf{r}(t_b)) \leftarrow (t_k, \mathbf{r}(t_k))$
  else
    $\mathbf{h}^{\text{out}}(t_k) \leftarrow \mathbf{h}(t_k) + \widehat{\mathbf{r}}(t_k)$
  end if
  Continue the remaining model computation from $\mathbf{h}^{\text{out}}(t_k)$.
end for

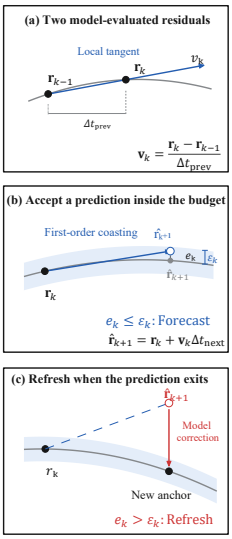


Figure 3: Trajectory-Aligned residual forecasting. Two model-evaluated residuals define a local temporal trend for predicting subsequent cached steps. Cached predictions do not modify the anchors; a refresh replaces the prediction with a newly evaluated residual and advances the anchor pair.

3.2 Trajectory-Aligned First-Order Residual Forecasting

Equation 2 treats the latest evaluated residual as constant until the next refresh. TAFC instead models how this residual evolves along physical sampling time. Let t_a and t_b denote the two most recent timesteps at which the cached transformer are fully evaluated, with $\mathbf{r}(t_a)$ and $\mathbf{r}(t_b)$ denoting the corresponding residuals. We estimate their local temporal variation by

$$\mathbf{d}_b = \frac{\mathbf{r}(t_b) - \mathbf{r}(t_a)}{|t_b - t_a|}, \tag{3}$$

and forecast the residual at a subsequent cached timestep t as

$$\widehat{\mathbf{r}}(t) = \mathbf{r}(t_b) + \mathbf{d}_b|t - t_b|. \tag{4}$$

The use of physical-time distances is important because diffusion and flow-matching schedules are generally nonuniform and may evolve in decreasing time. During consecutive cached steps, the two anchors remain fixed, while $|t - t_b|$ increases with the coasting horizon. A refresh produces a new model-evaluated residual and shifts the two anchors forward. Therefore, TAFC replaces the constant residual model with a local linear model estimated only from model-evaluated residuals.

Figure 3 summarizes this Trajectory-Aligned forecasting mechanism. The refresh decision shown in the algorithm is determined by the reliability criteria introduced in Section 3.3.

The benefit of the additional temporal information can be characterized locally without making assumptions on the full generative ODE. Since the implementation estimates the residual slope from two evaluated anchors, the approximation error contains both the local extrapolation error and the error of the finite-difference slope.

Proposition 3.1 (Local Accuracy of Residual Forecasting)**.** *Let s denote elapsed physical time along the sampling direction, and let $\mathbf{r}(s)$ be the residual defined in Eq. equation 1. Consider two evaluated anchors $s_a < s_b$ and a target $s > s_b$ satisfying $s_b - s_a \le \Delta$ and $s - s_b \le \Delta$.*

Assume that $\mathbf{r}$ is twice continuously differentiable on $[s_a, s]$, with $\|\mathbf{r}'(u)\| \le L$ and $\|\mathbf{r}''(u)\| \le M$ for all $u \in [s_a, s]$. Define the Trajectory-Aligned forecast by

$$\widehat{\mathbf{r}}_1(s) = \mathbf{r}(s_b) + \frac{\mathbf{r}(s_b) - \mathbf{r}(s_a)}{s_b - s_a}(s - s_b), \qquad \widehat{\mathbf{r}}_0(s) = \mathbf{r}(s_b). \tag{5}$$

Then we have

$$\|\mathbf{r}(s) - \widehat{\mathbf{r}}_1(s)\| \le M\Delta^2, \qquad \|\mathbf{r}(s) - \widehat{\mathbf{r}}_0(s)\| \le L\Delta. \tag{6}$$

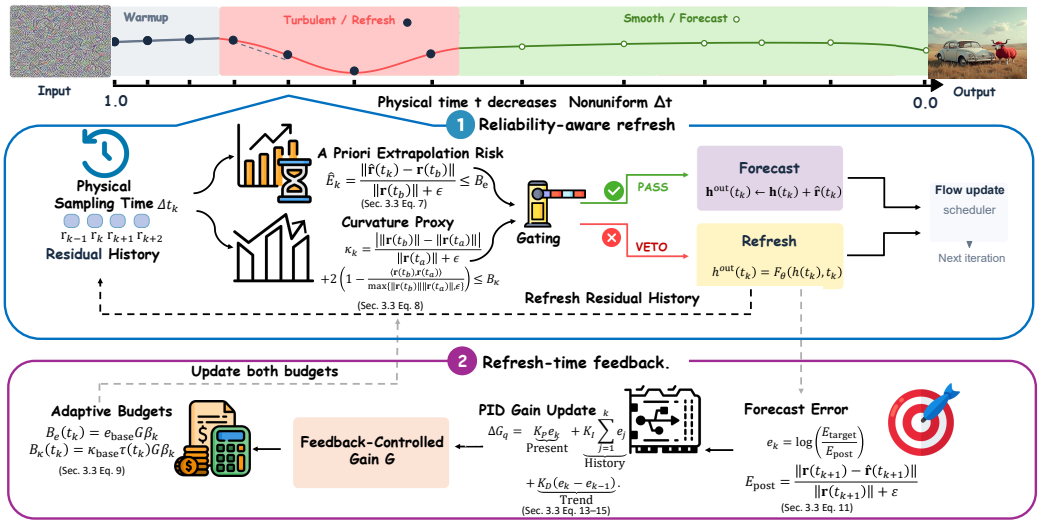


Figure 4: Practical TAFC under the trajectory-aligned formulation. Two evaluated residuals define a first-order forecast, while extrapolation risk and residual curvature are compared with adaptive budgets to determine forecast or refresh. At refresh steps, the observed error E_{post} updates the residual pair and drives PID feedback; the resulting gain G recalibrates both budgets for subsequent decisions.

Proposition 3.1 shows that constant residual reuse incurs an $\mathcal{O}(\Delta)$ local error because it ignores the leading temporal variation of the residual, whereas TAFC incorporates this variation through two evaluated anchors and achieves an $\mathcal{O}(\Delta^2)$ bound under the stated local smoothness conditions. This local improvement translates into a smaller cache-induced trajectory perturbation: Appendix A.1 shows that the deviation from the fully evaluated trajectory is bounded by the accumulated residual approximation errors. Since the first-order model is only locally valid, TAFC couples forecasting with the reliability mechanism introduced next to determine when a fresh evaluation is required.

3.3 Practical Reliability Control

Given the Trajectory-Aligned forecast in Eq. equation 4, TAFC determines whether the local residual model remains reliable enough for reuse or should be refreshed by a new evaluation.

Reliability-aware refresh. We first define an a priori extrapolation risk

$$\widehat{E}_k = \frac{\|\hat{\mathbf{r}}(t_k) - \mathbf{r}(t_b)\|}{\|\mathbf{r}(t_b)\| + \epsilon}, \tag{7}$$

where t_b is the latest evaluated anchor and $\epsilon > 0$ stabilizes the normalization. This quantity measures the relative magnitude of the extrapolated correction over the current coasting horizon.

We further characterize the variation between the two evaluated anchors t_a and t_b by

$$\kappa_k = \frac{\|\|\mathbf{r}(t_b)\| - \|\mathbf{r}(t_a)\|\|}{\|\mathbf{r}(t_a)\| + \epsilon} + 2\left(1 - \frac{\langle \mathbf{r}(t_b), \mathbf{r}(t_a)\rangle}{\max\{\|\mathbf{r}(t_b)\|\|\mathbf{r}(t_a)\|, \epsilon\}}\right), \tag{8}$$

which combines relative magnitude and directional changes and is used only as a local reliability signal.

The corresponding adaptive budgets are

$$B_e(t_k) = e_{\text{base}} G \beta_k, \qquad B_\kappa(t_k) = \kappa_{\text{base}} \tau(t_k) G \beta_k, \tag{9}$$

where $G > 0$ is the feedback-controlled gain, while β_k and $\tau(t_k)$ provide prescribed time-dependent modulation. The refresh indicator is

$$\mathbb{I}_{\text{ref}}(t_k) = \mathbf{1}\left\{\mathcal{C}_{\text{calib}} \vee \mathcal{C}_{\text{cut}} \vee \widehat{E}_k \geq B_e(t_k) \vee \kappa_k \geq B_\kappa(t_k) \vee c_k \geq M\right\}, \tag{10}$$

where $\mathcal{C}_{\text{calib}}$ and $\mathcal{C}_{\text{cut}}$ denote the prescribed calibration and cutoff regions, c_k is the consecutive-cache counter, and M is its maximum value.

Table 1: Comprehensive quantitative comparison of different acceleration methods on **FLUX.1-dev** at different acceleration levels. We report Latency and TFLOPs for efficiency, PSNR, SSIM, and LPIPS for reconstruction fidelity, along with CLIP Score, ImageReward, and HPSv3 for semantic and perceptual quality. HPSv3 uses 12,000 samples. The best results within each acceleration regime are highlighted in **bold**.

	Method	Efficiency		Reconstruction Quality			Semantic & Aesthetic Quality		
		Latency (s) ↓	TFLOPs ↓	PSNR ↑	SSIM ↑	LPIPS ↓	CLIP Score ↑	ImageReward ↑	HPSv3 ↑
	Baseline (50 steps)	20.90	1668	–	–	–	0.2742	0.9578	10.6060
$\sim 2.00\times$	Vanilla (25 steps)	10.50	834	15.55	0.668	0.409	0.2734	0.9465	10.0800
	TeaCache ($\delta = 0.3$)	10.34	862	20.76	0.810	0.211	0.2729	0.9508	10.5814
	TaylorSeer ($S = 3$)	9.96	748	22.78	0.828	0.163	0.2737	0.9542	10.2312
	SeaCache ($\delta = 0.3$)	9.82	782	26.29	0.893	0.106	0.2741	0.9567	10.5126
	TAFC ($\delta = 0.04$)	**9.37**	**703**	**29.87**	**0.941**	**0.051**	**0.2743**	**0.9600**	**10.6623**
$\sim 3.00\times$	Vanilla (15 steps)	6.40	501	17.84	0.740	0.305	0.2718	0.9216	9.7400
	Δ-DiT	7.16	604	17.40	0.710	0.336	0.2717	0.9286	9.8700
	ToCa	7.08	596	18.40	0.700	0.324	0.2724	0.9417	9.9600
	TeaCache ($\delta = 0.6$)	6.83	571	17.21	0.714	0.348	0.2729	0.9723	10.2568
	TaylorSeer ($S = 5$)	6.57	492	19.97	0.762	0.236	0.2731	0.9481	10.3669
	SeaCache ($\delta = 0.6$)	6.43	518	21.33	0.798	0.226	0.2738	0.9563	10.3573
	TAFC ($\delta = 0.15$)	**5.94**	**451**	**21.35**	**0.820**	**0.196**	**0.2756**	**0.9753**	**10.8605**
$\sim 4.00\times$	TeaCache ($\delta = 0.8$)	5.43	448	16.47	0.704	0.388	0.2763	0.9513	9.9860
	TaylorSeer ($S = 7$)	5.18	362	15.01	0.654	0.476	0.2766	0.9378	10.6458
	SeaCache ($\delta = 0.8$)	5.07	391	18.49	0.788	0.266	0.2752	0.9012	10.1945
	TAFC ($\delta = 0.3$)	**4.61**	**327**	**19.41**	**0.792**	**0.240**	**0.2769**	**0.9589**	**10.9129**

Refresh-time feedback. The actual forecast error is unavailable at cached steps and becomes observable only when a refresh evaluates the current residual. We then measure

$$E_{\text{post}} = \frac{\|\mathbf{r}(t_k) - \widehat{\mathbf{r}}(t_k)\|}{\|\mathbf{r}(t_k)\| + \epsilon}. \tag{11}$$

Warm-up refreshes form the calibration set $\mathcal{E}_{\text{calib}}$, from which we set

$$E_{\text{target}} = \max\left\{\text{median}\big(\mathcal{E}_{\text{calib}}\big), \epsilon\right\}. \tag{12}$$

After calibration, let q index refresh-time controller observations. With $S_0 = e_0 = 0$, we update

$$e_q = \log \frac{E_{\text{target}} + \epsilon}{E_{\text{post}} + \epsilon}, \tag{13}$$

$$S_q = S_{q-1} + e_q, \qquad \Delta G_q = K_P e_q + K_I S_q + K_D(e_q - e_{q-1}), \tag{14}$$

$$\log G \leftarrow \log G + \Delta G_q. \tag{15}$$

Thus, errors above the calibrated target tighten subsequent reuse through a smaller gain, while errors below the target allow more permissive budgets. The complete practical procedure, including anchor and counter updates, is given in Appendix A.2.

4 Experiments

4.1 Experimental Setup

Models and data. We evaluate FLUX.1-dev Labs (2024) at 1024×1024 resolution and HunyuanVideo Kong et al. (2024) and Wan2.1-T2V-1.3B Wan et al. (2025) at 480p with 65 frames. All models use a 50-step unaccelerated reference; experiments run on NVIDIA H200/A100 GPUs. Image reconstruction is evaluated on DrawBench Saharia et al. (2022), and video generation on 944 VBench prompts Huang et al. (2024). An additional HPSv3 evaluation covers 12,000 prompts across 12 semantic categories. **Metrics and baselines.** We report latency and TFLOPs for efficiency; PSNR, SSIM Wang et al. (2004), and LPIPS Zhang et al. (2018) for reconstruction against the 50-step output; and CLIP Score Radford et al. (2021), ImageReward Xu et al. (2023), and HPSv3 Ma et al. (2025) for image quality. Video quality uses VBench's aggregate, Quality, and Semantic scores. Baselines comprise step reduction, TeaCache Liu et al. (2025a), TaylorSeer Liu et al. (2025b), SeaCache Chung et al. (2026), ToCa Zou et al. (2024), and Δ-DiT Chen et al. (2024), as applicable in each table. Acceleration regimes are approximate; measured latency and TFLOPs specify the actual operating points.

Table 2: Quantitative comparison of different acceleration methods on **HunyuanVideo**. We report Latency and TFLOPs for efficiency, PSNR, SSIM, and LPIPS for reconstruction fidelity, along with VBench, Quality, and Semantic scores for video generation quality. The best results within each acceleration regime are highlighted in **bold**.

	Method	Efficiency		Reconstruction Quality			VBench Evaluation		
		Latency (s) ↓	TFLOPs ↓	PSNR ↑	SSIM ↑	LPIPS ↓	VBench Score ↑	Quality Score ↑	Semantic Score ↑
	Original (50 steps)	160.4	44457	–	–	–	81.66%	83.93%	72.58%
∼ 2.00×	Vanilla (25 steps)	80.2	22228	19.97	0.731	0.263	80.72%	83.04%	70.91%
	TeaCache ($\delta = 0.12$)	82.9	22694	23.27	0.808	0.201	81.18%	83.59%	71.57%
	TaylorSeer ($S = 2$)	81.7	22163	24.08	0.817	0.155	80.43%	82.74%	71.17%
	SeaCache ($\delta = 0.19$)	79.6	21374	27.60	0.863	0.136	81.64%	83.87%	72.73%
	TAFC ($\delta = 0.02$)	**78.8**	**20791**	**29.24**	**0.895**	**0.103**	**81.70%**	**83.93%**	**72.75%**
∼ 3.00×	Vanilla (15 steps)	53.5	13337	17.49	0.662	0.371	79.84%	82.36%	69.76%
	TeaCache ($\delta = 0.2$)	55.7	15742	20.57	0.752	0.278	80.90%	83.41%	70.86%
	TaylorSeer ($S = 3$)	54.8	15196	20.36	0.730	0.245	80.61%	82.96%	70.94%
	SeaCache ($\delta = 0.35$)	53.1	14623	20.07	0.750	0.291	80.77%	83.13%	71.30%
	TAFC ($\delta = 0.15$)	**52.2**	**13981**	**24.69**	**0.836**	**0.167**	**81.38%**	**83.52%**	**72.82%**

Table 3: Quantitative comparison of different acceleration methods on **Wan2.1-T2V-1.3B**. We report Latency and TFLOPs for efficiency, PSNR, SSIM, and LPIPS for reconstruction fidelity, along with VBench, Quality, and Semantic scores for video generation quality. The best results within each acceleration regime are highlighted in **bold**.

	Method	Efficiency		Reconstruction Quality			VBench Evaluation		
		Latency (s) ↓	TFLOPs ↓	PSNR ↑	SSIM ↑	LPIPS ↓	VBench Score ↑	Quality Score ↑	Semantic Score ↑
	Original (50 steps)	316.7	31949	–	–	–	81.49%	83.41%	73.78%
≀ 2.00×	TeaCache ($\delta = 0.09$)	156.2	15758	20.72	0.724	0.258	81.05%	82.70%	73.45%
	TaylorSeer ($S = 2$)	155.4	15677	16.21	0.547	0.332	80.45%	82.20%	72.95%
	SeaCache ($\delta = 0.20$)	154.9	15626	22.67	0.808	0.217	81.42%	82.96%	74.85%
	TAFC ($\delta = 0.03$)	**154.2**	**15556**	**25.56**	**0.862**	**0.092**	**81.53%**	**83.04%**	**75.50%**
≀ 3.00×	TeaCache ($\delta = 0.15$)	105.6	10653	19.02	0.661	0.396	80.73%	82.98%	71.55%
	TaylorSeer ($S = 3$)	105.1	10603	14.24	0.458	0.449	80.45%	82.50%	71.30%
	SeaCache ($\delta = 0.35$)	104.4	10532	17.52	0.728	0.242	80.61%	82.76%	71.95%
	TAFC ($\delta = 0.15$)	**103.8**	**10471**	**21.17**	**0.779**	**0.176**	**81.18%**	**83.13%**	**73.38%**

4.2 Quantitative Results

Image generation. Table 1 shows that TAFC achieves the best reconstruction metrics in each reported regime. Near $2\times$ acceleration, it improves PSNR over SeaCache from 26.29 to 29.87 dB and reduces LPIPS from 0.106 to 0.051, while lowering latency from 9.82 to 9.37 s. Near $3\times$, PSNR is similar (21.35 vs. 21.33 dB), but LPIPS improves to 0.196 from 0.226 at lower cost. Near $4\times$, TAFC retains 19.41 dB PSNR at 4.61 s, versus SeaCache's 18.49 dB at 5.07 s. TAFC also leads the reported semantic and preference metrics in these groups; its HPSv3 scores on 12,000 samples are 10.6623, 10.8605, and 10.9129. Category-level results appear in Appendix D. Preference scores can exceed the reference without implying better reconstruction; Appendix C discusses this distinction.

HunyuanVideo. In Table 2, TAFC reaches 29.24 dB PSNR and 0.103 LPIPS near $2\times$ acceleration, compared with SeaCache's 27.60 dB and 0.136. Its VBench score (81.70%) remains close to the reference (81.66%). Near $3\times$, TAFC retains 24.69 dB PSNR, whereas the competing caching methods obtain 20.07–20.57 dB. VBench decreases by only 0.28 percentage points from the reference, with latency reduced from 160.4 to 52.2 s.

Wan2.1-T2V-1.3B. Table 3 shows the same reconstruction advantage on a second video architecture. Near $2\times$, TAFC improves PSNR over SeaCache by 2.89 dB and reduces LPIPS from 0.217 to 0.092. Near $3\times$, it obtains 21.17 dB PSNR, versus 19.02, 17.52, and 14.24 dB for TeaCache, SeaCache, and TaylorSeer, respectively. Its VBench score remains 81.18%, compared with 81.49% for the reference, at 103.8 rather than 316.7 s. These results support transfer across the two evaluated video architectures; dimension-level results appear in Appendix E.

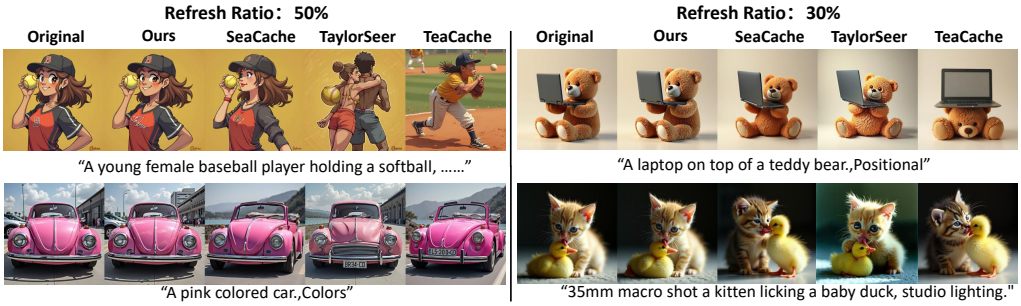


Figure 5: Qualitative comparison on FLUX.1-dev under different refresh ratios. We compare TAFC with SeaCache, TaylorSeer, and TeaCache. Existing caching methods can introduce semantic changes, incorrect object counts, or structural artifacts under aggressive caching. TAFC more faithfully preserves the composition, geometry, and fine-grained semantics of the original 50-step generation.

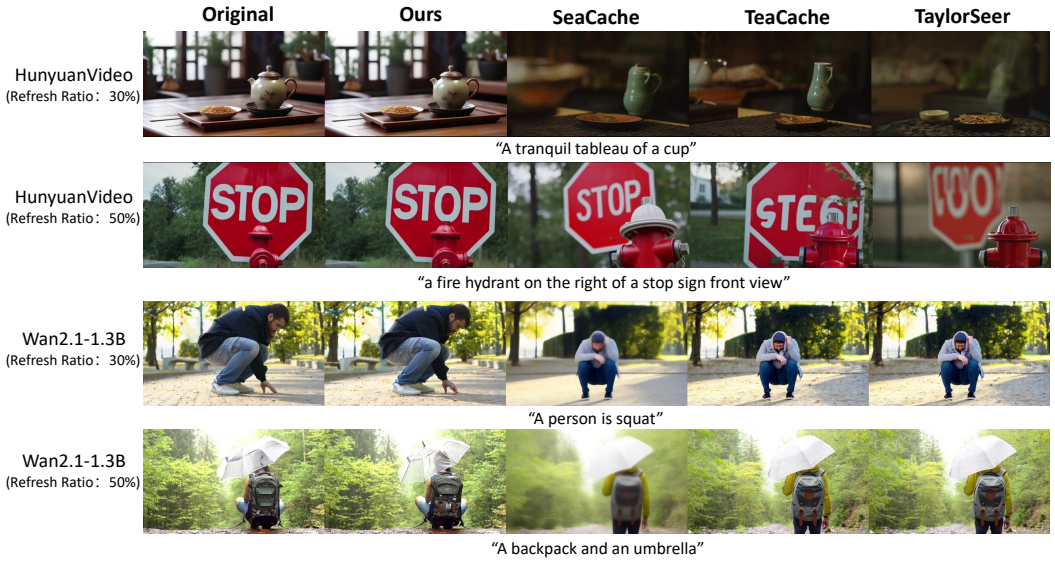


Figure 6: Qualitative reconstruction comparison on HunyuanVideo and Wan2.1-T2V-1.3B at approximately 30% and 50% refresh ratios. We compare the original 50-step generation with TAFC, SeaCache, TeaCache, and TaylorSeer. TAFC consistently remains closer to the original trajectory and better preserves object geometry, appearance, spatial layout, and fine visual details under aggressive caching.

Efficiency. TAFC has the lowest reported latency in each comparison group. TFLOPs and latency should be read jointly: on HunyuanVideo near $3\times$, for example, 15-step Vanilla uses fewer TFLOPs than TAFC but has slightly higher latency and substantially lower reconstruction fidelity. A refresh ratio alone therefore does not determine wall-clock speedup.

4.3 Qualitative Results

Figures 5 and 6 compare outputs at approximately 30% and 50% refresh ratios. On FLUX.1-dev, TAFC better preserves the reference composition, object count, and local geometry in the selected examples. On HunyuanVideo and Wan2.1, it better retains object appearance, scene layout, and fine detail. These visual comparisons complement reconstruction scores; individual video frames do not establish temporal coherence on their own. Additional examples and their interpretation are provided in Appendix F.

4.4 Additional Study

Component ablations. Figure 7 shows that removing any major component consistently degrades the PSNR–refresh ratio trade-off on both FLUX and HunyuanVideo, especially under aggressive caching. First-order extrapolation and physical-time modeling improve trajectory prediction, while adaptive thresholding and closed-loop control enhance reuse reliability. Together, these compo-

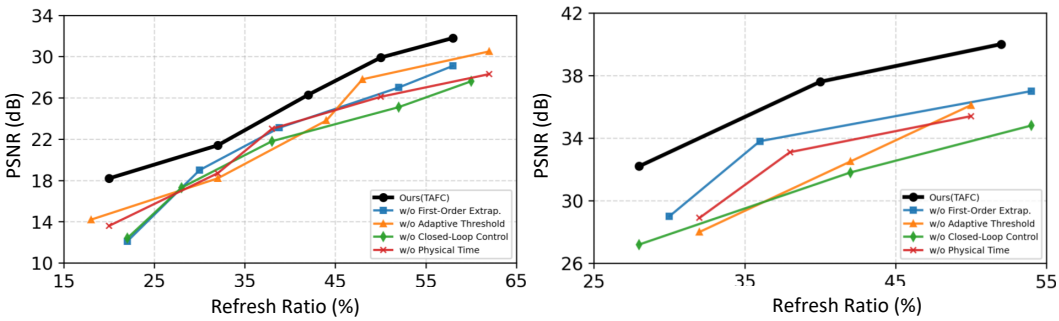


Figure 7: Component ablations on 200 DrawBench prompts for FLUX.1-dev and 20 VBench prompts for HunyuanVideo.

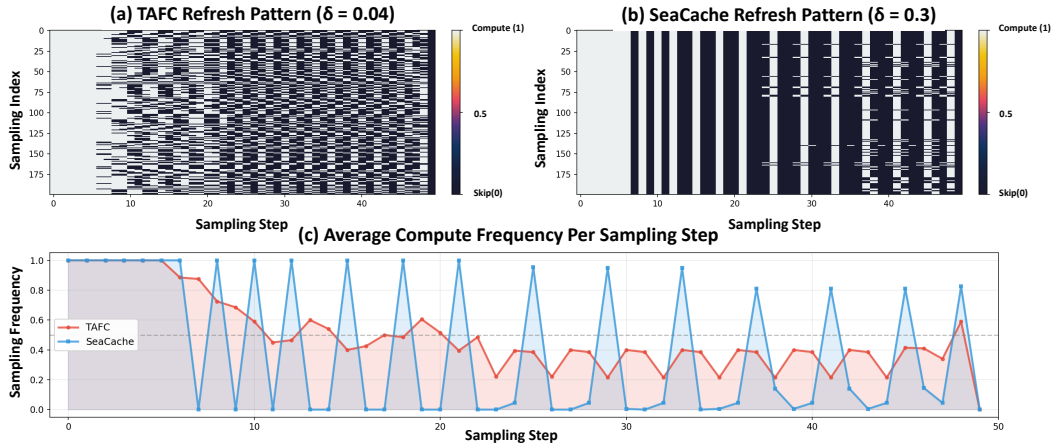


Figure 8: Refresh patterns for (a) TAFC ($\delta = 0.04$) and (b) SeaCache ($\delta = 0.3$). Light cells indicate computation; dark cells indicate reuse. Panel (c) averages decisions across samples.

nents enable TAFC to maintain higher reconstruction fidelity across refresh ratios.refresh ratios. **Refresh patterns.** Figure 8 illustrates the refresh patterns of TAFC and SeaCache across 200 DrawBench samples. TAFC shows dense early computation followed by selective, sample-dependent refreshes for TAFC. Its average compute frequency decreases toward intermediate levels, consistent with allocating computation to changing local dynamics after the initial structure-forming stage. In contrast, SeaCache exhibits more synchronized, approximately periodic stripes in this experiment, with sharper frequency oscillations and some later sample-specific variation. TAFC's greater cross-sample diversity is consistent with trajectory-dependent allocation rather than a shared refresh schedule; the heatmaps alone do not isolate curvature from the other controller components.

5 Conclusion

In this work, we introduce Trajectory-Aligned Flow Caching (TAFC), a training-free framework that shifts cache reuse from feature-level similarity estimation to the dynamics that directly govern the generative trajectory. TAFC predicts residual evolution through physical-time-aware first-order extrapolation, assesses the reliability of continued reuse using residual curvature, and uses a posteriori approximation errors observed at refresh steps to adapt reuse tolerances through closed-loop feedback. This combination turns caching into a predictive and adaptive trajectory-aware process in which each model evaluation not only updates the current state but also informs subsequent reuse decisions. Experiments on multiple visual generative models demonstrate favorable efficiency and fidelity trade-offs over existing training-free caching methods across image and video generation. These results highlight trajectory-aligned approximation error and online feedback as a promising basis for designing more reliable caching strategies for iterative generative models. More broadly, TAFC highlights the importance of modeling cache reuse over actual physical sampling intervals.

AI USE STATEMENT

We used AI-assisted tools to support manuscript editing, figure refinement, and LaTeX formatting. All AI-assisted content was reviewed and revised by the authors, who take full responsibility for the accuracy and integrity of the final manuscript.

REFERENCES

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

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.

Zhiwen Chen, Junhui Hou, Zhiyu Zhu, Jinjian Wu, and Guangming Shi. Scaling dense event-stream pretraining from visual foundation models. *arXiv preprint arXiv:2603.03969*, 2026.

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. *arXiv preprint arXiv:2602.18993*, 2026.

Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Genie: Higher-order denoising diffusion solvers. *Advances in Neural Information Processing Systems*, 35:30150–30166, 2022.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Huanlin Gao, Ping Chen, Fuyuan Shi, Ruijia Wu, Yantao Li, Qiang Hui, Yuren You, Ting Lu, Chao Tan, Shaoan Zhao, Zhaoxiang Liu, Fang Zhao, Kai Wang, and Shiguo Lian. Meancache: From instantaneous to average velocity for accelerating flow matching inference. 2026. URL `https://arxiv.org/abs/2601.19961`.

Yasaman Haghighi and Alexandre Alahi. Sencache: Accelerating diffusion model inference via sensitivity-aware caching. *arXiv preprint arXiv:2602.24208*, 2026.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jinhui Hou, Zhiyu Zhu, and Junhui Hou. Energy-oriented diffusion bridge for image restoration with foundational diffusion models. *arXiv preprint arXiv:2604.10983*, 2026.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024.

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.

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. `https://github.com/black-forest-labs/flux`, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7353–7363. IEEE, 2025a.

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

Joseph Liu, Joshua Geddes, Ziyu Guo, Haomiao Jiang, and Mahesh Kumar Nandwana. Smooth cache: A universal inference acceleration technique for diffusion transformers. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3220–3229. IEEE, 2025c.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022a.

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*, 2022b.

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.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22 (4):730–751, 2025a.

Jianrong Lu, Zhuoya Gu, Haobo Li, Zhiyu Zhu, Yechao Zhang, Jianhai Chen, Minghui Yang, Junwei Liu, Jian Wang, Qinming He, et al. Paso: Step parallel stochastic optimization. In *Forty-third International Conference on Machine Learning*.

Jianrong Lu, Zhiyu Zhu, and Junhui Hou. Parasolver: A hierarchical parallel integral solver for diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 41678–41704, 2025b.

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

Zhengyao 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*, volume 2025, pp. 33132–33156, 2025.

Xinyin Ma, Gongfan Fang, Michael Bi Mi, and Xinchao Wang. Learning-to-cache: Accelerating diffusion transformer via layer caching. *Advances in Neural Information Processing Systems*, 37: 133282–133304, 2024a.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15762–15772. IEEE, 2024b.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15086–15095. IEEE, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

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.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pp. 87–103. Springer, 2024.

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.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *International Conference on Learning Representations*, volume 2024, pp. 15078–15097, 2024.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

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.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6211–6220. IEEE, 2024.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:49842–49869, 2023.

Xuanlei Zhao, Xiaolong Jin, Kai Wang, and Yang You. Real-time video generation with pyramid attention broadcast, 2024b. *URL https://arxiv. org/abs/2408.12588*.

Qianang Zhou, Zhiyu Zhu, Junhui Hou, Yongjian Deng, Youfu Li, and Junlin Xiong. Resflow: Fine-tuning residual optical flow for event-based high temporal resolution motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 36(3):3527–3538, 2026. doi: 10.1109/TCSVT.2025.3613559.

Zhiyu Zhu, Jinhui Hou, Hui Liu, Huanqiang Zeng, and Junhui Hou. Learning efficient and effective trajectories for differential equation-based image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(10):9150–9168, 2025.

Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. *arXiv preprint arXiv:2410.05317*, 2024.

A Additional Analysis and Practical Details

A.1 Propagation of Residual Approximation Error

In this appendix, we further characterize how such local residual errors propagate through the sampling trajectory. Our purpose is not to establish convergence of the numerical ODE solver itself. Instead, we compare TAFC with full model evaluation under the same sampling schedule and the same numerical solver, so that the resulting deviation isolates the additional error introduced by caching.

Recall that $\mathbf{r}(t_k)$ denotes the residual obtained from a full evaluation of the Transformer mapping at sampling time t_k, whereas $\widehat{\mathbf{r}}(t_k)$ denotes the residual used by the cached computation. We define the residual approximation error at step k as $\eta_k = \|\mathbf{r}(t_k) - \widehat{\mathbf{r}}(t_k)\|$. For a refresh step, $\widehat{\mathbf{r}}(t_k) = \mathbf{r}(t_k)$ and hence $\eta_k = 0$.

Let $\mathbf{x}_k$ and $\widehat{\mathbf{x}}_k$ denote the latent states generated by full evaluation and TAFC, respectively, at sampling time t_k. To describe how a residual perturbation affects the numerical update, let $\mathbf{V}_\theta(\mathbf{x}, \mathbf{r}, t)$ denote the model output used by the sampler when the latent state is $\mathbf{x}$ and the Transformer residual is $\mathbf{r}$. This notation absorbs the feature construction and the remaining downstream network computation after the residual is inserted.

Assumption A.1 (Lipschitz Propagation of Cache Error)**.** Consider the set of latent states and residuals visited by the fully evaluated and cached trajectories over the sampling horizon. Assume that there exist constants $L_{\mathbf{x}} \geq 0$ and $L_{\mathrm{r}} \geq 0$ such that, for every sampling time t_k and any admissible states $\mathbf{x}, \mathbf{y}$ and residuals $\mathbf{r}, \widetilde{\mathbf{r}}$,

$$\|\mathbf{V}_\theta(\mathbf{x}, \mathbf{r}, t_k) - \mathbf{V}_\theta(\mathbf{y}, \mathbf{r}, t_k)\| \leq L_{\mathbf{x}} \|\mathbf{x} - \mathbf{y}\|, \tag{16}$$

and

$$\|\mathbf{V}_\theta(\mathbf{x}, \mathbf{r}, t_k) - \mathbf{V}_\theta(\mathbf{x}, \widetilde{\mathbf{r}}, t_k)\| \leq L_{\mathrm{r}} \|\mathbf{r} - \widetilde{\mathbf{r}}\|. \tag{17}$$

Equivalently, by the triangle inequality,

$$\|\mathbf{V}_\theta(\mathbf{x}, \mathbf{r}, t_k) - \mathbf{V}_\theta(\mathbf{y}, \widetilde{\mathbf{r}}, t_k)\| \leq L_{\mathbf{x}} \|\mathbf{x} - \mathbf{y}\| + L_{\mathrm{r}} \|\mathbf{r} - \widetilde{\mathbf{r}}\|. \tag{18}$$

The assumption only requires local Lipschitz regularity over the region visited by the two trajectories. It does not require the neural network to be globally Lipschitz over its entire input space.

Proposition A.2 (Propagation of Residual Approximation Error)**.** *Consider the same first-order numerical sampler applied to the fully evaluated and cached model outputs. Let $\Delta t_k = |t_{k+1} - t_k|$, and suppose that both trajectories start from the same initial latent state, $\widehat{\mathbf{x}}_0 = \mathbf{x}_0$. Under Assumption A.1, their deviation after N sampling steps satisfies*

$$\|\widehat{\mathbf{x}}_N - \mathbf{x}_N\| \leq L_{\mathrm{r}} e^{L_{\mathbf{x}} T} \sum_{k=0}^{N-1} \Delta t_k \, \eta_k, \tag{19}$$

where $T = \sum_{k=0}^{N-1} \Delta t_k$.

Consequently, if $\eta_k \leq \eta$ for every cached step, then

$$\|\widehat{\mathbf{x}}_N - \mathbf{x}_N\| \leq L_{\mathrm{r}} T e^{L_{\mathbf{x}} T} \eta. \tag{20}$$

Proposition A.2 connects the residual approximation studied in Proposition 3.1 to the resulting sampling trajectory. In particular, the bound depends on the weighted accumulation of the residual errors η_k, while refresh steps contribute zero cache error. Since both trajectories use the same numerical solver, the result does not include the discretization error of the solver itself and instead isolates the additional deviation caused by cached model evaluations.

Combined with Proposition 3.1, this provides the following interpretation: within a local temporal region where the first-order residual model is accurate, improving the residual approximation from a linear-in-Δ bound to a quadratic-in-Δ bound correspondingly reduces the perturbation injected into the sampling trajectory. The result is conditional on the stated local regularity assumptions and does not imply a uniform global guarantee for arbitrary cache horizons.

A.2 Complete Practical TAFC Procedure

The main text presents an abstract version of TAFC that highlights the core Trajectory-Aligned forecasting mechanism and the refresh/cache branches. For completeness, Algorithm 2 gives the full practical procedure used in our implementation. In addition to the first-order residual forecast, it explicitly includes anchor initialization, calibration, reliability-based refresh decisions, consecutive-cache control, posterior forecast-error measurement, and closed-loop budget adaptation.

The algorithm follows the same notation as the main text: only freshly evaluated residuals update the anchor pair, whereas cached predictions leave the anchors unchanged and extrapolate from the most recent evaluated target. The refresh indicator combines the practical reliability criteria introduced in Section 3.3, while the feedback controller is updated only when a fresh evaluation makes the posterior prediction error observable.

B Deferred Proofs

B.1 Proof of Proposition 3.1

Proof. Let $h = s_b - s_a$ and $\delta = s - s_b$. By assumption, $0 < h \le \Delta$ and $0 < \delta \le \Delta$.

The trajectory-aligned slope is

$$\mathbf{d}_b = \frac{\mathbf{r}(s_b) - \mathbf{r}(s_a)}{h}. \tag{21}$$

Using the fundamental theorem of calculus,

$$\mathbf{d}_b = \frac{1}{h}\int_{s_a}^{s_b} \mathbf{r}'(u)\, du. \tag{22}$$

Therefore,

$$\mathbf{d}_b - \mathbf{r}'(s_b) = \frac{1}{h}\int_{s_a}^{s_b} (\mathbf{r}'(u) - \mathbf{r}'(s_b))\, du. \tag{23}$$

Since $\|\mathbf{r}''(u)\| \le M$ on $[s_a, s_b]$, we have

$$\|\mathbf{r}'(u) - \mathbf{r}'(s_b)\| \le M(s_b - u). \tag{24}$$

Hence,

$$\|\mathbf{d}_b - \mathbf{r}'(s_b)\| \le \frac{M}{h}\int_{s_a}^{s_b} (s_b - u)\, du = \frac{Mh}{2}. \tag{25}$$

Next, Taylor's theorem with integral remainder gives

$$\mathbf{r}(s) = \mathbf{r}(s_b) + \mathbf{r}'(s_b)\delta + \int_{s_b}^{s} (s-u)\mathbf{r}''(u)\, du. \tag{26}$$

The trajectory-aligned forecast is $\widehat{\mathbf{r}}_1(s) = \mathbf{r}(s_b) + \mathbf{d}_b\delta$. Subtracting the two expressions yields

$$\mathbf{r}(s) - \widehat{\mathbf{r}}_1(s) = (\mathbf{r}'(s_b) - \mathbf{d}_b)\,\delta + \int_{s_b}^{s} (s-u)\mathbf{r}''(u)\, du. \tag{27}$$

Taking norms and applying Eq. equation 25,

$$\|\mathbf{r}(s) - \widehat{\mathbf{r}}_1(s)\| \le \frac{Mh}{2}\delta + M\int_{s_b}^{s} (s-u)\, du \tag{28}$$

$$= \frac{M}{2}\left(h\delta + \delta^2\right). \tag{29}$$

Since $h \le \Delta$ and $\delta \le \Delta$,

$$\|\mathbf{r}(s) - \widehat{\mathbf{r}}_1(s)\| \le M\Delta^2. \tag{30}$$

For constant residual reuse, $\widehat{\mathbf{r}}_0(s) = \mathbf{r}(s_b)$. Again by the fundamental theorem of calculus,

$$\mathbf{r}(s) - \mathbf{r}(s_b) = \int_{s_b}^{s} \mathbf{r}'(u)\, du. \tag{31}$$

Algorithm 2 Practical Trajectory-Aligned Flow Caching (TAFC)

Require: Sampling schedule $\{t_k\}_{k=0}^{N}$, Transformer mapping $\mathcal{F}_\theta$, base budgets e_{base} and κ_{base}, maximum cache run M, calibration length N_{calib}, and controller coefficients K_P, K_I, K_D.

$G \leftarrow 1,\ c \leftarrow 0,\ q \leftarrow 0,\ S_0 \leftarrow 0,\ e_0 \leftarrow 0,\ \mathcal{E}_{\text{calib}} \leftarrow \emptyset$ *{Initialize cache and controller states}*
Initialize the two residual anchors as empty.
for $k = 0$ **to** N **do**
  Obtain the current hidden representation $\mathbf{h}(t_k)$.
  if fewer than two evaluated anchors are available **then**
    $\mathbf{h}^{\text{out}}(t_k) \leftarrow \mathcal{F}_\theta\big(\mathbf{h}(t_k), t_k\big)$
    $\mathbf{r}(t_k) \leftarrow \mathbf{h}^{\text{out}}(t_k) - \mathbf{h}(t_k)$ *{Bootstrap evaluated residuals}*
    Append $(t_k, \mathbf{r}(t_k))$ to the anchor history.
    Continue the remaining model computation and sampling update.
    continue
  end if
  Let $(t_a, \mathbf{r}(t_a))$ and $(t_b, \mathbf{r}(t_b))$ be the two most recent evaluated anchors.
  $\mathbf{d}_b \leftarrow \dfrac{\mathbf{r}(t_b) - \mathbf{r}(t_a)}{|t_b - t_a|}$
  $\widehat{\mathbf{r}}(t_k) \leftarrow \mathbf{r}(t_b) + \mathbf{d}_b |t_k - t_b|$ *{trajectory-aligned first-order forecast}*
  $\widehat{E}_k \leftarrow \dfrac{\|\widehat{\mathbf{r}}(t_k) - \mathbf{r}(t_b)\|}{\|\mathbf{r}(t_b)\| + \epsilon}$ *{A priori extrapolation risk}*
  Compute κ_k from $\mathbf{r}(t_a)$ and $\mathbf{r}(t_b)$ using Eq. equation 8.
  Compute $B_e(t_k)$ and $B_\kappa(t_k)$ using Eq. equation 9.
  $\mathbb{I}_{\text{ref}}(t_k) \leftarrow \mathbf{1}\Big\{\mathcal{C}_{\text{calib}} \vee \mathcal{C}_{\text{cut}} \vee \widehat{E}_k \geq B_e(t_k) \vee \kappa_k \geq B_\kappa(t_k) \vee c \geq M\Big\}$ *{Reliability-aware refresh decision}*
  if $\mathbb{I}_{\text{ref}}(t_k) = 1$ **then**
    $\mathbf{h}^{\text{out}}(t_k) \leftarrow \mathcal{F}_\theta\big(\mathbf{h}(t_k), t_k\big)$
    $\mathbf{r}(t_k) \leftarrow \mathbf{h}^{\text{out}}(t_k) - \mathbf{h}(t_k)$ *{Fresh evaluation}*
    $E_{\text{post}} \leftarrow \dfrac{\|\mathbf{r}(t_k) - \widehat{\mathbf{r}}(t_k)\|}{\|\mathbf{r}(t_k)\| + \epsilon}$ *{Observe forecast error}*
    if t_k is in the calibration region **then**
      $\mathcal{E}_{\text{calib}} \leftarrow \mathcal{E}_{\text{calib}} \cup \{E_{\text{post}}\}$
      if calibration is completed **then**
        $E_{\text{target}} \leftarrow \max\big\{\text{median}\big(\mathcal{E}_{\text{calib}}\big), \epsilon\big\}$
      end if
    else
      $q \leftarrow q + 1$
      $e_q \leftarrow \log \dfrac{E_{\text{target}} + \epsilon}{E_{\text{post}} + \epsilon}$
      $S_q \leftarrow S_{q-1} + e_q$
      $\Delta G_q \leftarrow K_P e_q + K_I S_q + K_D (e_q - e_{q-1})$
      $\log G \leftarrow \log G + \Delta G_q$ *{Closed-loop budget adaptation}*
    end if
    $(t_a, \mathbf{r}(t_a)) \leftarrow (t_b, \mathbf{r}(t_b))$
    $(t_b, \mathbf{r}(t_b)) \leftarrow (t_k, \mathbf{r}(t_k))$ *{Advance evaluated anchors}*
    $c \leftarrow 0$
  else
    $\mathbf{h}^{\text{out}}(t_k) \leftarrow \mathbf{h}(t_k) + \widehat{\mathbf{r}}(t_k)$ *{Skip $\mathcal{F}_\theta$ using the forecast}*
    $c \leftarrow c + 1$ *{Keep the evaluated anchors fixed}*
  end if
  Continue the remaining model computation from $\mathbf{h}^{\text{out}}(t_k)$ and perform the sampling update.
end for

Ensure: Generated sample after the final sampling step.

Using $\|\mathbf{r}'(u)\| \le L$,

$$\|\mathbf{r}(s) - \widehat{\mathbf{r}}_0(s)\| \le L\delta \le L\Delta. \tag{32}$$

Therefore, within the stated local temporal neighborhood, the trajectory-aligned first-order forecast admits a quadratic residual approximation bound in Δ, whereas constant residual reuse admits a linear bound.

□

B.2 Proof of Proposition A.2

Proof. Let

$$e_k = \|\widehat{\mathbf{x}}_k - \mathbf{x}_k\| \tag{33}$$

denote the latent deviation at sampling step k. Both trajectories start from the same initial state, so $e_0 = 0$.

For the first-order numerical sampler, the fully evaluated trajectory and the cached trajectory can be written as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t_k \, \mathbf{V}_\theta\big(\mathbf{x}_k, \mathbf{r}(t_k), t_k\big), \tag{34}$$

and

$$\widehat{\mathbf{x}}_{k+1} = \widehat{\mathbf{x}}_k + \Delta t_k \, \mathbf{V}_\theta\big(\widehat{\mathbf{x}}_k, \widehat{\mathbf{r}}(t_k), t_k\big), \tag{35}$$

where $\Delta t_k = |t_{k+1} - t_k|$.

Subtracting the two updates gives

$$\widehat{\mathbf{x}}_{k+1} - \mathbf{x}_{k+1} = \widehat{\mathbf{x}}_k - \mathbf{x}_k \tag{36}$$

$$+ \Delta t_k \Big[\mathbf{V}_\theta\big(\widehat{\mathbf{x}}_k, \widehat{\mathbf{r}}(t_k), t_k\big) - \mathbf{V}_\theta\big(\mathbf{x}_k, \mathbf{r}(t_k), t_k\big)\Big]. \tag{37}$$

Taking norms and applying Assumption A.1,

$$e_{k+1} \le e_k + \Delta t_k \left[L_{\mathbf{x}} e_k + L_{\mathrm{r}} \|\widehat{\mathbf{r}}(t_k) - \mathbf{r}(t_k)\|\right] \tag{38}$$

$$= (1 + L_{\mathbf{x}} \Delta t_k)\, e_k + L_{\mathrm{r}} \Delta t_k \eta_k. \tag{39}$$

Iterating Eq. equation 39 from $e_0 = 0$ yields

$$e_N \le L_{\mathrm{r}} \sum_{k=0}^{N-1} \Delta t_k \eta_k \prod_{j=k+1}^{N-1} (1 + L_{\mathbf{x}} \Delta t_j)\,. \tag{40}$$

Using $1 + z \le e^z$ for $z \ge 0$,

$$\prod_{j=k+1}^{N-1} (1 + L_{\mathbf{x}} \Delta t_j) \le \exp\left(L_{\mathbf{x}} \sum_{j=k+1}^{N-1} \Delta t_j\right) \tag{41}$$

$$\le e^{L_{\mathbf{x}} T}, \tag{42}$$

where $T = \sum_{j=0}^{N-1} \Delta t_j$. Therefore,

$$\|\widehat{\mathbf{x}}_N - \mathbf{x}_N\| = e_N \le L_{\mathrm{r}} e^{L_{\mathbf{x}} T} \sum_{k=0}^{N-1} \Delta t_k \eta_k, \tag{43}$$

which proves Eq. equation 19.

If $\eta_k \le \eta$ for all cached steps, while refresh steps satisfy $\eta_k = 0$, then

$$\sum_{k=0}^{N-1} \Delta t_k \eta_k \le \eta \sum_{k=0}^{N-1} \Delta t_k \tag{44}$$

$$= \eta T. \tag{45}$$

Consequently,

$$\|\widehat{\mathbf{x}}_N - \mathbf{x}_N\| \le L_{\mathrm{r}} T e^{L_{\mathbf{x}} T} \eta, \tag{46}$$

which proves Eq. equation 20.

□

C Evaluation Protocol and Interpretation

This section complements the experimental setup in Section 4.1. Appendix E discusses the existing dimension-level VBench results, and Appendix F provides additional visual comparisons. The deferred proofs above are retained separately from this empirical discussion.

Reference fidelity and generation quality. Reconstruction metrics measure agreement with the original 50-step output, rather than with a ground-truth photograph or video. Higher PSNR and SSIM and lower LPIPS indicate better preservation of that reference. CLIP Score, ImageReward, HPSv3, and VBench evaluate different aspects of the generated output, so their rankings need not coincide with reconstruction rankings. In particular, a preference-score increase above the unaccelerated model does not imply that the original trajectory was reconstructed more accurately. Conversely, high reference fidelity does not correct errors already present in the reference generation.

Compute budgets. The tables group methods into nominal acceleration regimes, with the corresponding thresholds or skip parameters shown in each row. These parameters are method-specific and are not directly comparable across algorithms. For a reported latency L and reference latency L_{ref}, the measured speedup is L_{ref}/L. Thus the three FLUX operating points for TAFC correspond to approximately $2.23\times$, $3.52\times$, and $4.53\times$ speedup. A refresh ratio describes the fraction of steps that perform a model refresh; its reciprocal need not equal measured speedup because cached steps and auxiliary operations also incur cost. Latencies should be compared within each model's reported setting, rather than across architectures running on different hardware configurations.

Evaluation coverage. The main video evaluation uses 944 VBench prompts, whereas the video ablation uses a separate, smaller set of 20 randomly sampled VBench prompts. The FLUX ablation uses 200 DrawBench prompts. The ablation curves therefore characterize component behavior on these subsets. The 12,000-sample HPSv3 evaluation spans 12 semantic categories, with category-level and overall scores reported in Table 4. Overall rankings do not imply uniform improvements in every category. Aggregate differences are descriptive: the supplied results do not include repeated-run uncertainty estimates or significance tests.

D Detailed HPSv3 Results

Table 4 reports HPSv3 scores on 12,000 samples across 12 semantic categories. The two panels follow the baseline and acceleration-regime grouping of Table 1, with the same method order in each group. TAFC occupies the top three overall positions, while TaylorSeer ($S = 7$) achieves the highest scores for Animals, Transportation, Products, and Food. These category differences reinforce the distinction between aggregate preference and uniform category-level improvements.

E Detailed Video Generation Results

Tables 5 and 6 report the individual VBench dimensions for the two video models. Each table retains the original two groups of accelerated settings. These results help distinguish preservation of appearance and composition from individual motion or style scores; TAFC need not rank first on every dimension to offer a favorable overall trade-off.

HunyuanVideo. At the lower TAFC threshold ($\delta = 0.02$), imaging quality matches the reference at 63.38%, while multiple-object performance remains close (71.49% vs. 71.27%). At $\delta = 0.15$, TAFC retains 62.44% imaging quality, compared with 59.49% for SeaCache and 59.80% for TaylorSeer. Its multiple-object score is 68.06%, exceeding the competing caching methods in that group but remaining below the reference. SeaCache achieves higher scores on several temporal dimensions, including motion smoothness. The detailed results therefore indicate a balance of visual fidelity and semantic preservation, rather than uniform dominance across metrics.

Wan2.1-T2V-1.3B. TAFC preserves imaging quality at 64.44% and 64.43% for its two settings, close to the reference's 64.59%. At $\delta = 0.15$, spatial-relationship performance is 72.68%, compared

Table 4: Category-level HPSv3 evaluation on 12,000 samples, grouped and ordered as in Table 1. Higher is better. Bold marks the best available score within each acceleration regime; blue shading identifies TAFC. – denotes an unavailable category score.

Method	Characters	Arts	Design	Architecture	Animals	Natural Scenery
Baseline (50 steps)	11.8670	10.3978	9.5248	11.1313	10.4751	10.2504
$\sim 2.00\times$ **Acceleration**						
TeaCache ($\delta = 0.3$)	11.8584	**10.4256**	**9.5514**	11.1009	10.4762	10.1346
TaylorSeer ($S = 3$)	11.2076	9.4480	8.6860	10.8385	10.3600	9.7472
SeaCache ($\delta = 0.3$)	11.7976	10.3391	9.4713	10.9866	10.4129	10.0362
TAFC ($\delta = 0.04$)	**11.9220**	10.4221	9.5373	**11.1981**	**10.5453**	**10.2952**
$\sim 3.00\times$ **Acceleration**						
TeaCache ($\delta = 0.6$)	11.6537	10.2176	9.2783	10.7847	10.2324	9.6920
TaylorSeer ($S = 5$)	11.3735	9.5055	8.7515	10.9478	10.4880	9.8657
SeaCache ($\delta = 0.6$)	11.6314	10.1197	9.3789	10.7576	10.2928	9.7320
TAFC ($\delta = 0.15$)	**12.0373**	**10.5439**	**9.6808**	**11.4208**	**10.8196**	**10.5042**
$\sim 4.00\times$ **Acceleration**						
TeaCache ($\delta = 0.8$)	11.4170	10.0509	9.0420	10.5610	10.0069	9.3578
TaylorSeer ($S = 7$)	11.6761	9.7113	8.7805	11.2410	**10.9676**	10.1529
SeaCache ($\delta = 0.8$)	11.4926	9.8780	9.2586	10.5544	10.1146	9.4294
TAFC ($\delta = 0.3$)	**12.0897**	**10.5806**	**9.6793**	**11.4887**	10.8731	**10.5881**

Method	Transportation	Products	Others	Plants	Food	Science	Overall
Baseline (50 steps)	10.9703	10.6828	10.4581	10.7312	11.3538	9.4291	10.6060
$\sim 2.00\times$ **Acceleration**							
TeaCache ($\delta = 0.3$)	10.9720	10.6846	10.3754	10.6440	11.3477	9.4056	10.5814
TaylorSeer ($S = 3$)	10.8079	10.6935	10.1745	10.5615	**11.5876**	8.6622	10.2312
SeaCache ($\delta = 0.3$)	10.8907	10.6463	10.3284	10.6215	11.2914	9.3294	10.5126
TAFC ($\delta = 0.04$)	**11.0558**	**10.7660**	**10.5076**	**10.8124**	11.4394	**9.4460**	**10.6623**
$\sim 3.00\times$ **Acceleration**							
TeaCache ($\delta = 0.6$)	10.6446	10.4294	10.0088	10.0365	11.0043	9.1000	10.2568
TaylorSeer ($S = 5$)	11.0017	10.8632	10.3469	10.6905	**11.7644**	8.8040	10.3669
SeaCache ($\delta = 0.6$)	10.7584	10.5606	10.1560	10.4856	11.2620	9.1527	10.3573
TAFC ($\delta = 0.15$)	**11.3100**	**10.9835**	**10.6848**	**11.0166**	11.6854	**9.6394**	**10.8605**
$\sim 4.00\times$ **Acceleration**							
TeaCache ($\delta = 0.8$)	10.4193	10.1863	9.7731	9.5095	10.6398	8.8690	9.9860
TaylorSeer ($S = 7$)	**11.4667**	**11.2030**	10.7296	10.7726	**12.0322**	9.0158	10.6458
SeaCache ($\delta = 0.8$)	10.6635	10.5065	9.9990	10.3144	11.1528	8.9704	10.1945
TAFC ($\delta = 0.3$)	11.2923	11.0391	**10.7405**	**11.1476**	11.7452	**9.6908**	**10.9129**

with 56.31%–57.63% for the other caching methods, and overall consistency reaches 70.80%. Nevertheless, TaylorSeer scores higher on motion smoothness and color in that group. Reading these dimensions alongside reconstruction metrics avoids treating any one aspect of VBench as a complete measure of video quality.

F Additional Qualitative Comparisons

Figures 9 and 10 extend the examples in Section 4.3. The original generation is the visual reference, and "Ours" denotes TAFC. Each panel identifies its refresh ratio and comparison method.

Image examples. The FLUX examples cover object counting, color, spatial relations, and more unusual prompt compositions. At 50% refresh, the stack of books and the yellow-book/red-vase scene illustrate preservation of object structure and layout. At 30%, the tomato-on-pumpkin and banana/apple examples expose changes in relative placement and shape. TAFC generally remains

Table 5: Detailed VBench evaluation on **HunyuanVideo**. We report fine-grained evaluation results across spatial, temporal, visual-quality, and semantic dimensions. The best result within each acceleration regime is highlighted in **bold**.

Method	Subject Consistency	Background Consistency	Temporal Flickering	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Imaging Quality	Object Class
Vanilla	94.21%	95.22%	97.25%	97.12%	38.20%	60.15%	63.38%	82.04%
TeaCache ($\delta = 0.12$)	94.13%	94.92%	97.30%	97.15%	37.50%	**60.58%**	61.73%	82.36%
TaylorSeer ($S = 2$)	92.34%	93.74%	**97.68%**	96.95%	37.50%	59.07%	60.55%	81.41%
SeaCache ($\gamma = 0.19$)	**94.44%**	95.13%	97.38%	**97.36%**	**38.20%**	60.25%	62.38%	**83.07%**
TAFC ($\delta = 0.02$)	94.28%	**95.17%**	97.38%	97.08%	**38.20%**	60.06%	**63.38%**	81.57%
TeaCache ($\delta = 0.2$)	93.84%	94.71%	97.38%	97.05%	**38.89%**	59.81%	60.51%	81.57%
TaylorSeer ($S = 3$)	91.15%	92.82%	97.45%	96.42%	36.80%	58.42%	59.80%	80.25%
SeaCache ($\gamma = 0.35$)	**94.67%**	**95.18%**	**97.76%**	**97.77%**	36.11%	59.39%	59.49%	79.51%
TAFC ($\delta = 0.15$)	94.24%	95.13%	97.33%	97.29%	36.11%	**60.32%**	**62.44%**	**82.36%**

Method	Multiple Objects	Human Action	Color	Spatial Relationship	Scene	Temporal Style	Appearance Style	Overall Consistency
Vanilla	71.27%	97.00%	79.46%	71.79%	45.79%	64.67%	69.29%	71.87%
TeaCache ($\delta = 0.12$)	61.13%	95.00%	83.56%	70.33%	46.14%	64.67%	69.18%	**71.76%**
TaylorSeer ($S = 2$)	61.97%	95.00%	**87.16%**	60.30%	**48.61%**	**64.95%**	**70.03%**	71.10%
SeaCache ($\gamma = 0.19$)	67.15%	95.00%	86.04%	69.50%	48.53%	64.51%	69.33%	71.46%
TAFC ($\delta = 0.02$)	**71.49%**	**96.00%**	81.01%	**70.74%**	47.73%	64.92%	69.75%	71.57%
TeaCache ($\delta = 0.2$)	58.84%	95.00%	82.95%	**65.91%**	47.02%	**65.49%**	69.47%	71.51%
TaylorSeer ($S = 3$)	59.50%	94.00%	86.20%	59.10%	49.15%	65.10%	69.80%	70.45%
SeaCache ($\gamma = 0.35$)	61.20%	95.00%	**87.20%**	61.79%	**50.12%**	**65.30%**	**70.48%**	71.13%
TAFC ($\delta = 0.15$)	**68.06%**	**98.00%**	86.15%	64.44%	49.85%	64.67%	70.13%	**71.68%**

Table 6: Detailed VBench evaluation on **Wan2.1-T2V-1.3B**. We report fine-grained evaluation results across spatial, temporal, visual-quality, and semantic dimensions. The best result within each acceleration regime is highlighted in **bold**.

Method	Subject Consistency	Background Consistency	Temporal Flickering	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Imaging Quality	Object Class
Vanilla	94.65%	96.43%	97.63%	93.89%	33.34%	61.68%	64.59%	86.95%
TeaCache ($\delta = 0.09$)	93.95%	95.15%	96.25%	93.95%	32.10%	53.80%	52.50%	81.20%
TaylorSeer ($S = 2$)	93.82%	94.85%	96.44%	**94.34%**	31.95%	52.96%	51.28%	79.27%
SeaCache ($\delta = 0.2$)	94.14%	95.42%	96.36%	94.10%	**32.64%**	54.47%	53.77%	82.44%
TAFC ($\delta = 0.03$)	**94.34%**	**95.87%**	**97.46%**	93.62%	**32.64%**	**61.40%**	**64.44%**	**87.10%**
TeaCache ($\delta = 0.15$)	93.50%	94.70%	95.90%	93.50%	31.50%	52.80%	51.10%	79.80%
TaylorSeer ($S = 3$)	93.83%	94.96%	96.79%	**94.55%**	31.95%	52.43%	50.29%	80.14%
SeaCache ($\delta = 0.35$)	93.93%	94.99%	96.47%	94.27%	31.95%	53.48%	51.95%	80.62%
TAFC ($\delta = 0.15$)	**94.39%**	**96.06%**	**97.44%**	93.55%	**33.34%**	**61.13%**	**64.43%**	**84.97%**

Method	Multiple Objects	Human Action	Color	Spatial Relationship	Scene	Temporal Style	Appearance Style	Overall Consistency
Vanilla	66.69%	84.00%	83.36%	75.17%	46.23%	68.60%	82.54%	70.52%
TeaCache ($\delta = 0.09$)	55.20%	80.50%	86.10%	58.10%	34.80%	63.10%	82.10%	65.10%
TaylorSeer ($S = 2$)	55.11%	80.00%	**88.06%**	58.44%	35.17%	62.42%	82.43%	64.26%
SeaCache ($\delta = 0.2$)	56.71%	81.00%	86.74%	59.01%	35.62%	63.79%	82.29%	65.88%
TAFC ($\delta = 0.03$)	**70.05%**	**86.00%**	84.95%	**78.31%**	**51.45%**	**68.41%**	**82.61%**	**70.66%**
TeaCache ($\delta = 0.15$)	54.10%	79.00%	85.50%	57.00%	33.50%	62.00%	81.90%	64.10%
TaylorSeer ($S = 3$)	54.04%	78.00%	**88.49%**	56.31%	30.94%	61.24%	**82.75%**	63.60%
SeaCache ($\delta = 0.35$)	54.88%	80.00%	86.19%	57.63%	31.65%	62.66%	82.50%	64.97%
TAFC ($\delta = 0.15$)	**64.33%**	**80.81%**	87.19%	**72.68%**	**49.67%**	**68.10%**	81.83%	**70.80%**

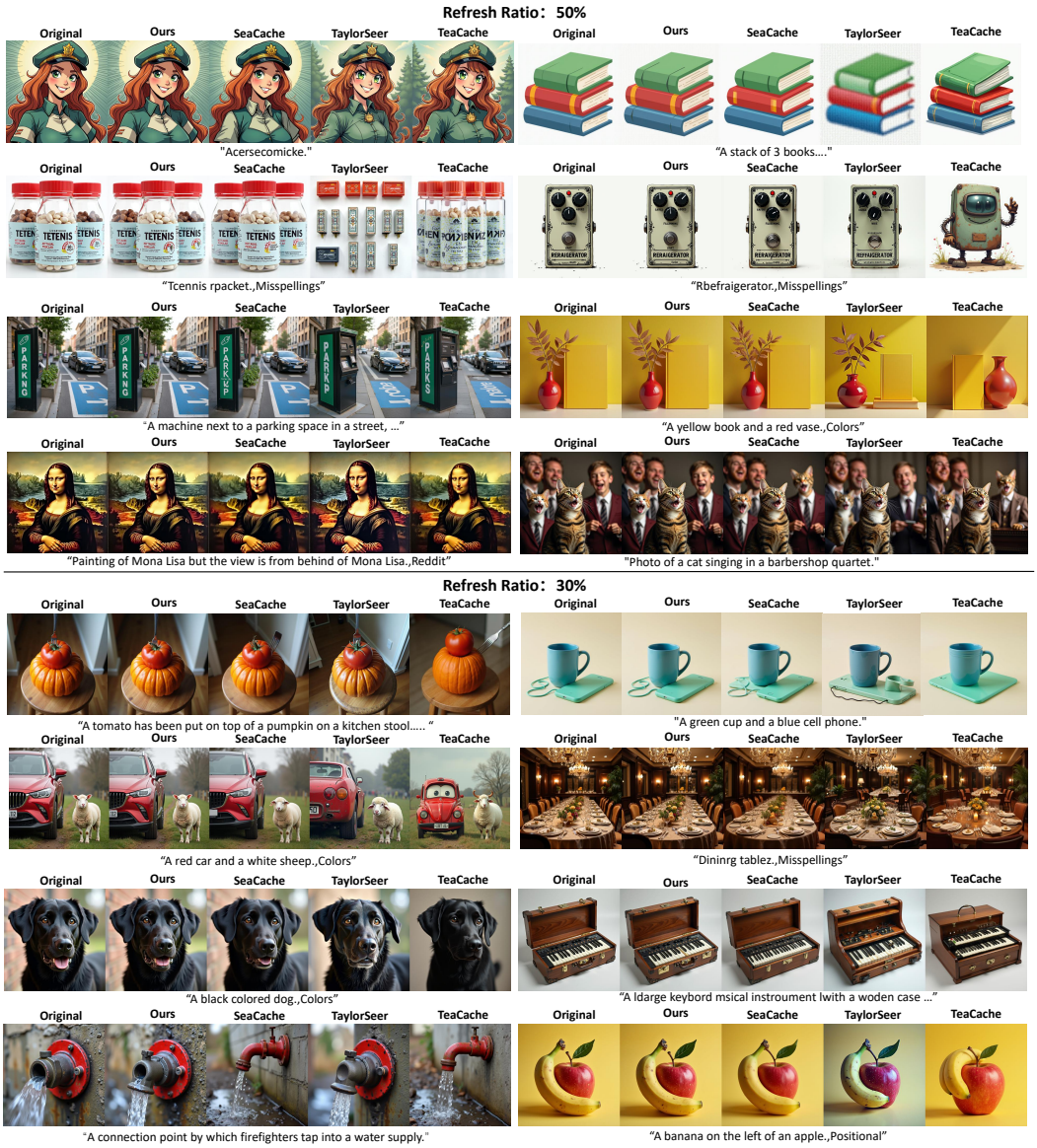


Figure 9: Additional qualitative comparisons on FLUX. We provide more generated examples to demonstrate the effectiveness of TAFC under different acceleration settings while preserving visual quality and semantic consistency.

closer to the reference in these panels. Reference agreement should be distinguished from prompt correctness, particularly for unusual or misspelled prompts that the original model may already interpret imperfectly.

Video examples. The supplementary video panels compare HunyuanVideo and Wan2.1 at 30% and 50% refresh ratios. They allow inspection of object shape, appearance, and background layout under caching. These are selected frames, so claims about motion continuity should also be assessed through the temporal VBench dimensions rather than inferred solely from still images.

Scope and limitations. Qualitative examples illustrate possible outcomes, not their frequency across the prompt distribution. The evaluations cover the three specified backbones and generation settings; they do not establish behavior for other schedulers, resolutions, or video lengths. Finally, refresh-time feedback observes approximation error only when the model is evaluated. It

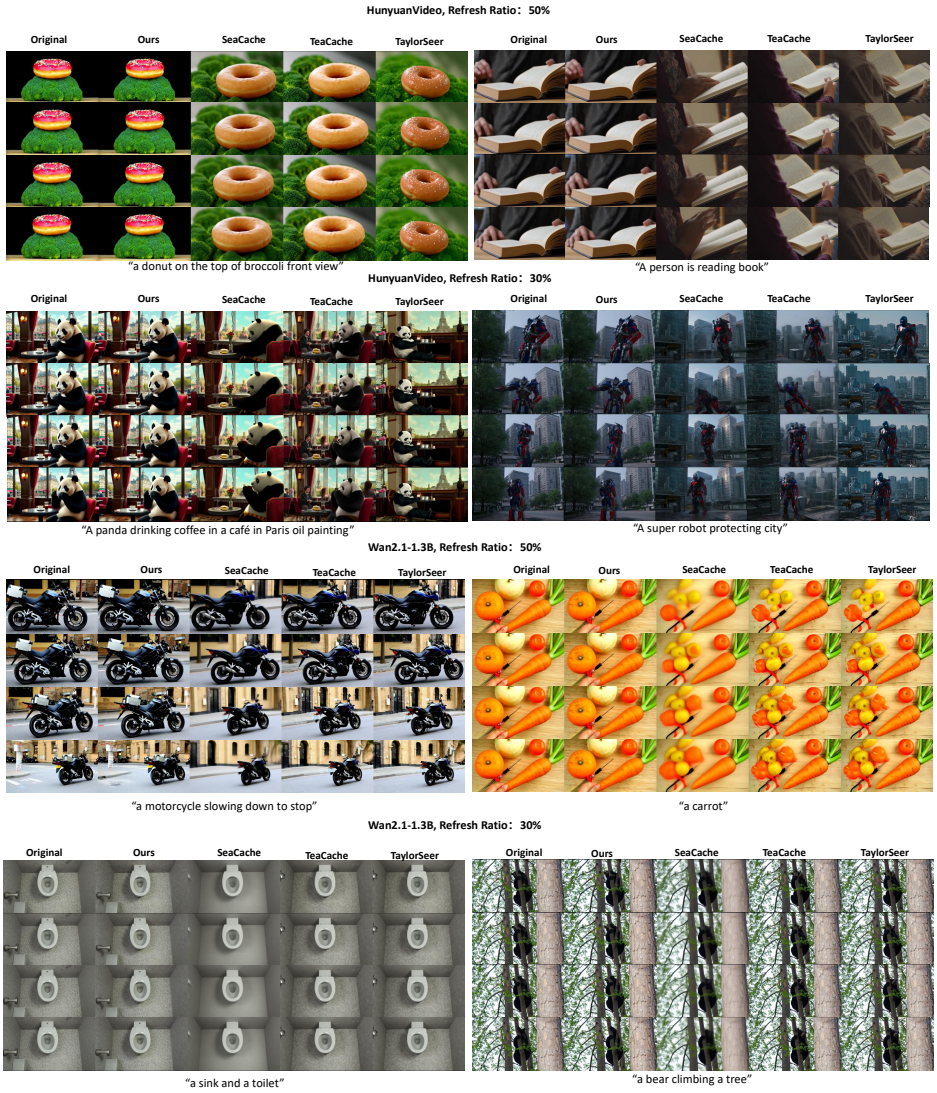


Figure 10: Additional qualitative comparisons on HunyuanVideo and Wan2.1-1.3B under different refresh ratios. We compare TAFC with SeaCache, TeaCache, and TaylorSeer at refresh ratios of 30% and 50%. TAFC consistently preserves temporal coherence, object structure, and fine-grained visual details while maintaining high similarity to the original generation.

cannot directly measure errors at skipped steps or independently certify the oracle error condition. The reported fidelity gains are empirical evidence for the practical controller within the evaluated settings.