NaviCache: Test-Time Self-Calibration Caching for Video Generation

Zheqi Lv[12] Zhibo Zhu[1] Jinke Wang[1] Qi Tian[1] Shengyu Zhang[1]* Zhengyu Chen[3]
Chengxi Zang[2] Zhou Zhao[1] Fei Wu[1]

[1] Zhejiang University, China [2] Cornell University, USA [3] Meituan, China

* *Corresponding Author*

ICML 2026 · Seoul, South Korea

Core idea: turn caching from a fixed heuristic into a navigation problem, using test-time self-calibration to know when a DiT block can be safely reused.

Offline calibration-free • Plug-and-play • Test-time self-calibration • Navigation-guided caching • Trajectory-aware caching

浙江大学 ZHEJIANG UNIVERSITY | Cornell University

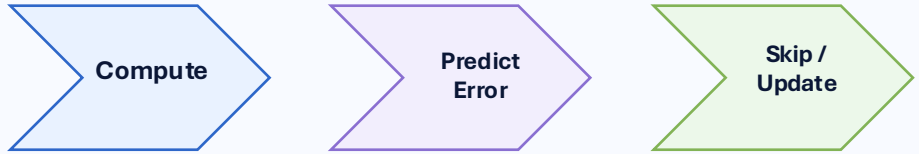
The bottleneck: video diffusion is powerful, but expensive

Large VDMs repeatedly execute massive DiT backbones across dozens of denoising steps.

Why it matters

1. Every generated video requires many sequential forward passes.
2. Each pass touches high-dimensional spatiotemporal features.
3. Small skip-decision errors can accumulate into visible artifacts.
4. The real deployment goal is not only speed, but safe speed.

Acceleration is a decision problem



The hardest part is judging whether a block output is still close enough to reuse.

Paper basis: Introduction motivates the computational tax of iterative VDM sampling.

Why existing caching methods are not enough

Prior caching methods either depend on offline calibration or lack runtime trajectory tracking.

Offline calibration
TeaCache / MagCache

- ✓ Learns a mapping before inference
- ! Calibration cost and dataset dependence
- ! Vulnerable to distribution shift

Offline calibration-free
EasyCache-style ZOH

- ✓ No calibration dataset required
- ≈ Uses previous ratio as current estimate
- ! Lags when feature dynamics drift

NaviCache
Test-time self-calibration

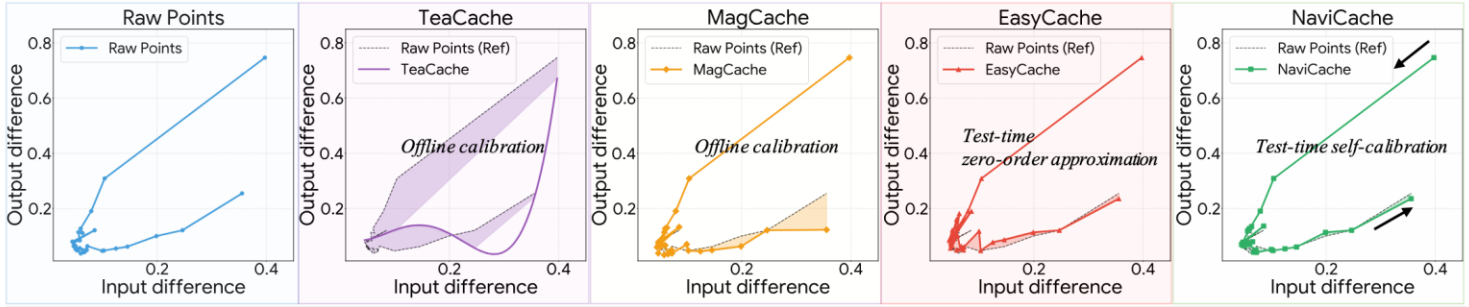
- ✓ No offline calibration
- ✓ Tracks state and uncertainty at runtime
- ✓ Updates only when the safety gate says so

Key gap: runtime dynamics are non-stationary

浙江大學 ZHEJIANG UNIVERSITY | Cornell University

Observation: feature evolution behaves like a trajectory

The input-output difference relationship is structured; it is not just random jitter.



Category	Metric	TeaCache	MagCache	EasyCache	NaviCache (Ours)
Characteristic	Offline calibration-free	✗	✗	✓	✓
	Test-time Self-calibration	✗	✗	✗	✓
Point-level	MAE (↓)	0.0415	0.0181	0.0197	**0.0164**
	RMSE (↓)	0.0488	0.0245	0.0288	**0.0243**
	Difference Region Area (↓)	0.0101	0.0048	0.0043	**0.0036**
Distribution-level	KL Divergence (↓)	0.1267	0.0431	0.0426	**0.0256**
	Pearson Correlation (↑)	-0.2302	0.8535	0.8314	**0.9084**
	Cosine Similarity (↑)	0.8743	0.9673	0.9652	**0.9810**

Figure 1. Comparison of prediction accuracy for determining whether to skip a specific DiT block output in VDMs. We visualize the relationship between input and output differences as a 2D manifold. The shaded areas are constructed by the segments connecting predicted coordinates and ground-truth coordinates (Raw Points). Based on quantitative evaluation across all prompts in VBench (Huang et al., 2024), we employ Point-level metrics (MAE, RMSE, and Difference Region Area) to measure instantaneous error, and Distribution-level metrics (KL Divergence, Pearson Correlation, and Cosine Similarity) to assess the alignment between predicted and real coordinates. *TeaCache* (Liu et al., 2025) and *MagCache* (Ma et al., 2026) rely on offline calibration, failing to capture runtime dynamics; *EasyCache* (Zhou et al., 2025) uses a test-time zero-order approximation which leads to significant lag. In contrast, our *NaviCache* achieves plug-and-play test-time self-calibration, accurately tracking the trajectory with minimal deviation.

From points to navigation

$$\phi_{t-1} = \frac{\Delta \boldsymbol{O}_{t-1}}{\Delta \boldsymbol{I}_{t-1}}$$

NaviCache tracks the changing sensitivity ratio between output variation and input variation.

Fig. 1 shows NaviCache has the lowest MAE and KL divergence, and the highest Pearson correlation and cosine similarity among compared methods.

Core insight: cache reliability is a state-estimation problem, not a static fitting or zero-order reuse problem

Method design: NaviCache as test-time navigation

A dual-state estimator tracks both the ratio and its uncertainty during inference.

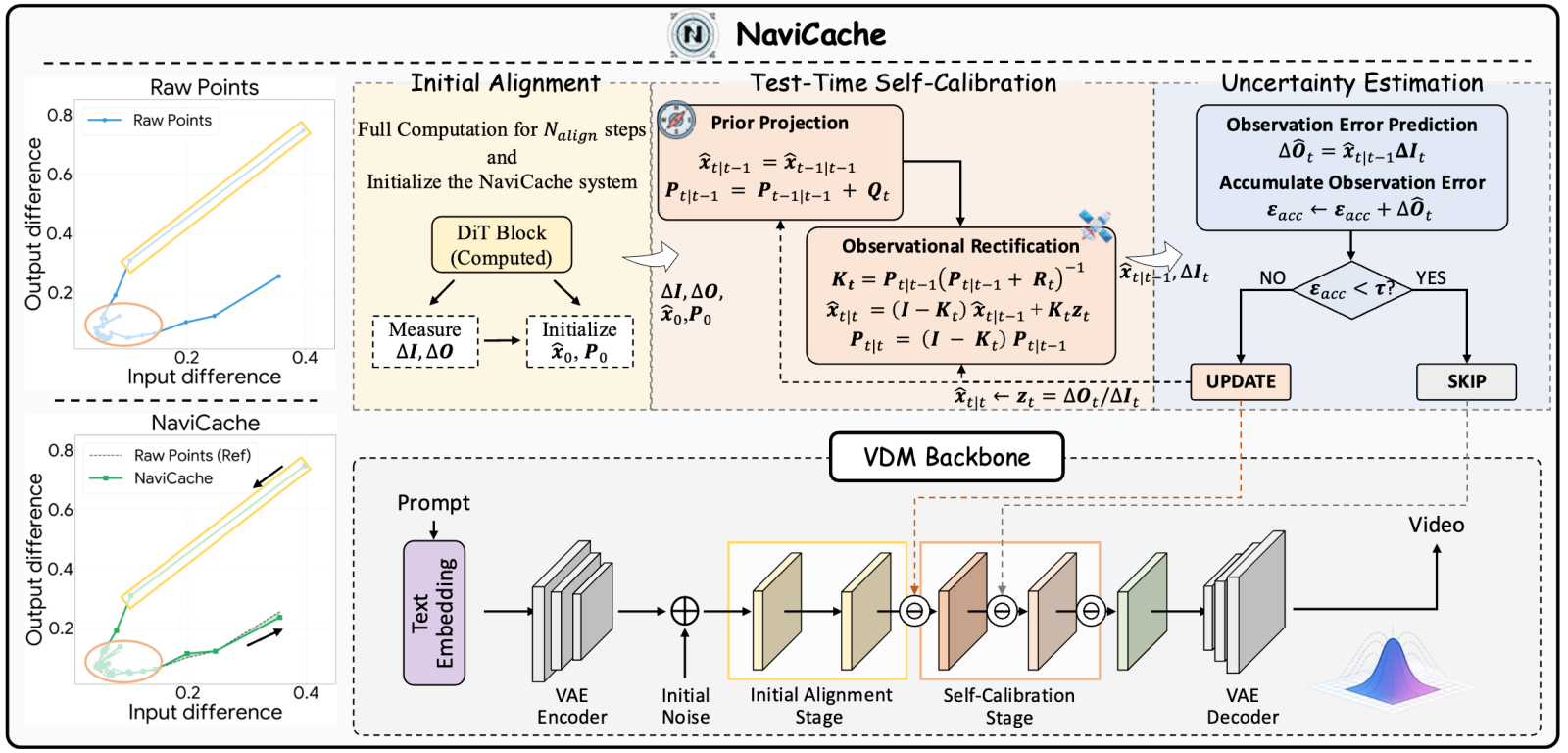


Figure 2. Overview of the *NaviCache* Framework. *NaviCache* reformulates feature caching as a recursive state-space tracking problem to enable offline calibration-free acceleration for VDMs. Left: Comparison between raw trajectories (top) and *NaviCache* predicted trajectories (bottom), categorized into the Initial Alignment Stage and the Self-Calibration Stage. Right: Detailed modular workflow. The former stage is dedicated to initializing the *NaviCache* system through full computation; the latter employs the Test-Time Self-Calibration engine to recursively track feature dynamics, integrated with an Uncertainty Estimation module to adaptively determine whether to SKIP the current execution. *NaviCache* ensures the VDM Backbone maintains high visual fidelity while significantly reducing cost.

Three-stage workflow

1. Initial Alignment: fully compute early turbulent steps
2. Self-Calibration: recursively track state and covariance
3. Safety Gate: skip only while estimated error stays bounded

Fig. 2: caching becomes recursive state-space tracking

Mechanism: uncertainty is the guardrail

The estimator fuses inertial prediction with measured correction when needed.

State and measurement model

$$\boldsymbol{x}_t = \boldsymbol{F}_t \boldsymbol{x}_{t-1} + \boldsymbol{w}_{t-1}, \qquad \boldsymbol{w}_{t-1} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_t)$$
$$\boldsymbol{z}_t = \boldsymbol{H}_t \boldsymbol{x}_t + \boldsymbol{v}_t, \qquad \boldsymbol{v}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_t)$$

Calibrated fusion factor

$$\boldsymbol{K}_t = \boldsymbol{P}_{t|t-1}(\boldsymbol{P}_{t|t-1} + \boldsymbol{R}_t)^{-1},$$
$$\boldsymbol{P}_{t|t} = (\boldsymbol{I} - \boldsymbol{K}_t)\boldsymbol{P}_{t|t-1}.$$

Prior projection

$$\hat{\boldsymbol{x}}_{t|t-1} = \hat{\boldsymbol{x}}_{t-1|t-1}, \quad \boldsymbol{P}_{t|t-1} = \boldsymbol{P}_{t-1|t-1} + \boldsymbol{Q}_t.$$

Safety gate and theory

$$\text{Action}_t = \begin{cases} \textbf{SKIP} & \text{if } \mathcal{E}_{acc} < \tau \\ \textbf{UPDATE} & \text{if } \mathcal{E}_{acc} \geq \tau \end{cases}$$

Stable regime: trust historical inertia.
Turbulent regime: trust new measurements.
Result: tighter error bound than zero-order offline calibration-free caching.

Main results: speed without sacrificing fidelity

NaviCache consistently improves the fidelity-speed trade-off across major VDM backbones.

Model	Reference	Efficiency		Visual Retention			Vbench (%) ↑
		Latency(s) ↓	Speedup ↑	PSNR ↑	SSIM ↑	LPIPS ↓	
Wan 2.1-1.3B (81 frames, 832 × 480)							
Wan 2.1 ($T = 50$)	-	214.93	1.00×	-	-	-	80.86
+ 40% steps	-	100.37	2.14×	14.50	0.5226	0.4374	80.30
+ Random 0.4	-	102.75	2.09×	11.92	0.4204	0.5911	78.68
+ Static cache	-	102.14	2.10×	14.18	0.5007	0.4789	79.58
+ PAB (Zhao et al., 2025)	ICLR'25	141.28	1.52×	18.84	0.6484	0.3010	77.60
+ TeaCache (Liu et al., 2025)	CVPR'25	121.57	1.77×	22.79	0.8169	0.0952	**80.67**
+ MagCache (Ma et al., 2026)	NeurIPS'25	105.05	2.05×	23.33	0.8331	0.0958	80.26
+ EasyCache (Zhou et al., 2025)	-	99.21	2.17×	22.96	0.7935	0.1053	80.18
+ ***NaviCache (Ours) - fast***	-	96.40	2.23×	23.46	0.8011	0.0956	80.21
+ ***NaviCache (Ours) - mid***	-	106.97	2.01×	24.08	0.8333	0.0839	80.38
+ ***NaviCache (Ours) - slow***	-	115.86	1.86×	**25.10**	**0.8638**	**0.0686**	80.58
HunyuanVideo (129 frames, 960 × 544)							
HunyuanVideo ($T = 50$)	-	2363.83	1.00×	-	-	-	82.53
+ 50% steps	-	1203.44	1.96×	18.79	0.7101	0.3319	81.78
+ Random 0.5	-	1213.09	1.95×	19.85	0.7201	0.3214	81.04
+ Static cache	-	1338.51	1.77×	18.74	0.7081	0.3309	81.76
+ PAB (Zhao et al., 2025)	ICLR'25	1700.60	1.39×	18.58	0.7023	0.3827	76.98
+ TeaCache (Liu et al., 2025)	CVPR'25	1070.14	2.21×	24.05	0.8046	0.1830	**82.32**
+ MagCache (Ma et al., 2026)	NeurIPS'25	882.76	2.68×	29.83	0.8904	0.0941	81.99
+ EasyCache (Zhou et al., 2025)	-	1100.30	2.15×	32.53	0.9241	0.0589	82.04
+ ***NaviCache (Ours) - fast***	-	928.45	2.55×	30.64	0.8982	0.0860	82.03
+ ***NaviCache (Ours) - mid***	-	1089.43	2.17×	32.65	0.9256	0.0571	82.07
+ ***NaviCache (Ours) - slow***	-	1150.87	2.05×	**33.87**	**0.9357**	**0.0462**	82.15
Open-Sora 1.2 (51 frames, 848 × 480)							
Open-Sora 1.2 ($T = 30$)	-	56.48	1.00×	-	-	-	79.25
+ 50% steps	-	29.84	1.89×	15.82	0.6961	0.3363	77.36
+ Random 0.5	-	29.71	1.90×	16.51	0.7037	0.3264	76.78
+ Static cache	-	30.47	1.85×	15.73	0.6961	0.3382	77.37
+ PAB (Zhao et al., 2025)	ICLR'25	45.51	1.24×	23.58	0.8220	0.1743	76.95
+ TeaCache (Liu et al., 2025)	CVPR'25	41.38	1.36×	23.16	0.8335	0.1429	**79.10**
+ MagCache (Ma et al., 2026)	NeurIPS'25	26.07	2.17×	22.33	0.8207	0.1676	77.97
+ EasyCache (Zhou et al., 2025)	-	34.55	1.63×	23.63	0.8458	0.1320	78.83
+ ***NaviCache (Ours) - fast***	-	31.80	1.78×	22.53	0.8255	0.1551	79.05
+ ***NaviCache (Ours) - mid***	-	35.29	1.60×	23.67	0.8463	0.1309	79.03
+ ***NaviCache (Ours) - slow***	-	40.98	1.38×	**26.37**	**0.8860**	**0.0917**	79.08

Main results

W Wan 2.1: fast mode reaches 2.23× speedup with stronger PSNR than zero-order EasyCache.

H HunyuanVideo: mid mode reaches 2.17× speedup, PSNR 32.65, LPIPS 0.0571.

O Open-Sora: slow mode improves PSNR to 26.37 with strong perceptual retention.

No offline calibration is required, yet runtime self-calibration preserves strong visual fidelity.

Qualitative evidence: safer skipping produces cleaner videos

The visual case study shows fewer structural artifacts in hands, pages, and motion.

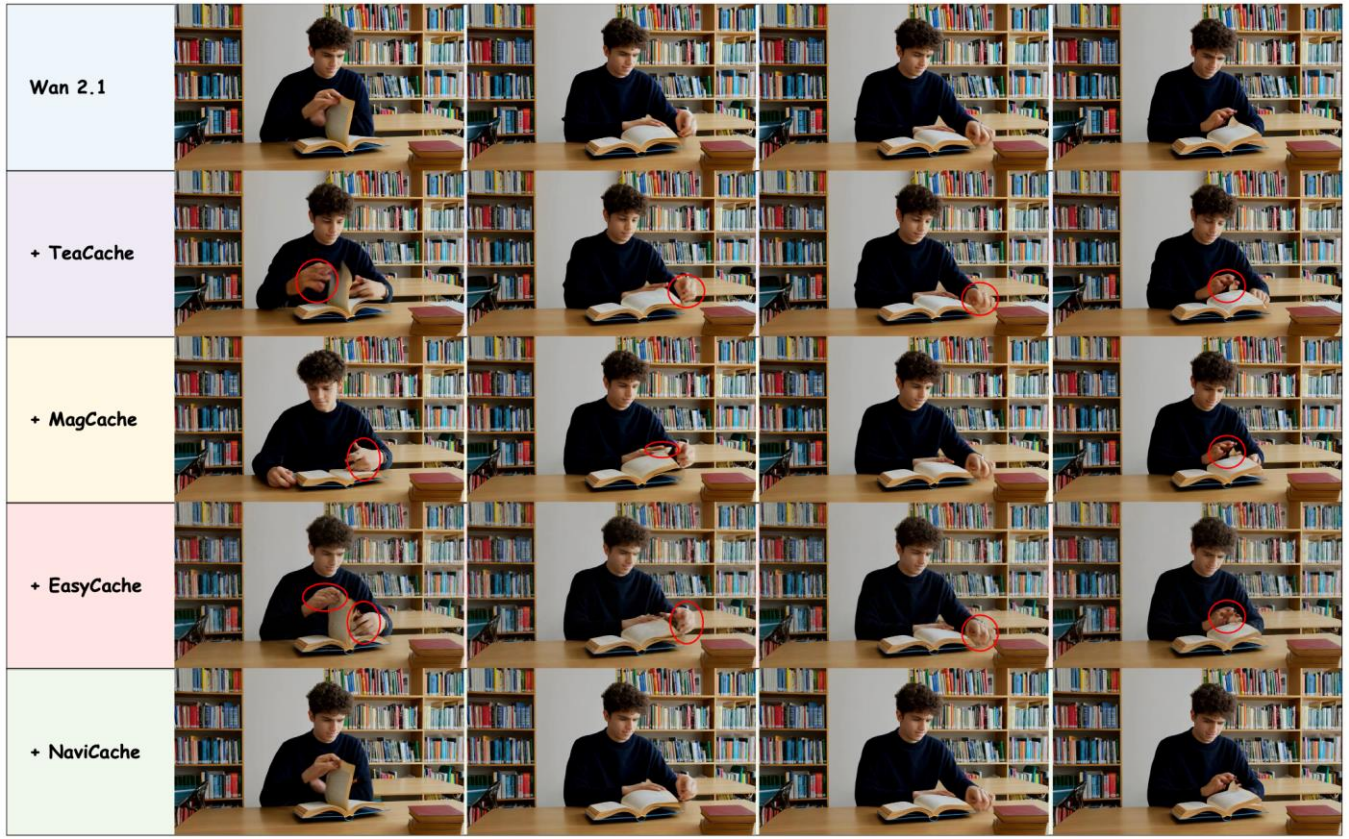


Figure 3. Visualization and Case Study of video generation. For the prompt "*A young man sitting at a desk in a library reading*", compared to existing methods that suffer from blurriness and structural artifacts (highlighted in red), our *NaviCache* preserves high visual fidelity and motion continuity, closely matching the unaccelerated baseline.

Why this matters

- Offline calibration-based methods can show hand or object deformation.
- Zero-order offline calibration-free caching can lag behind dynamics, producing blur or distortion.
- NaviCache tracks uncertainty and re-anchors when the trajectory drifts.

Fig. 3: NaviCache stays closer to the unaccelerated baseline

Adaptive computation: each prompt gets its own schedule

NaviCache does not allocate compute by a fixed global template; it reacts to runtime uncertainty.

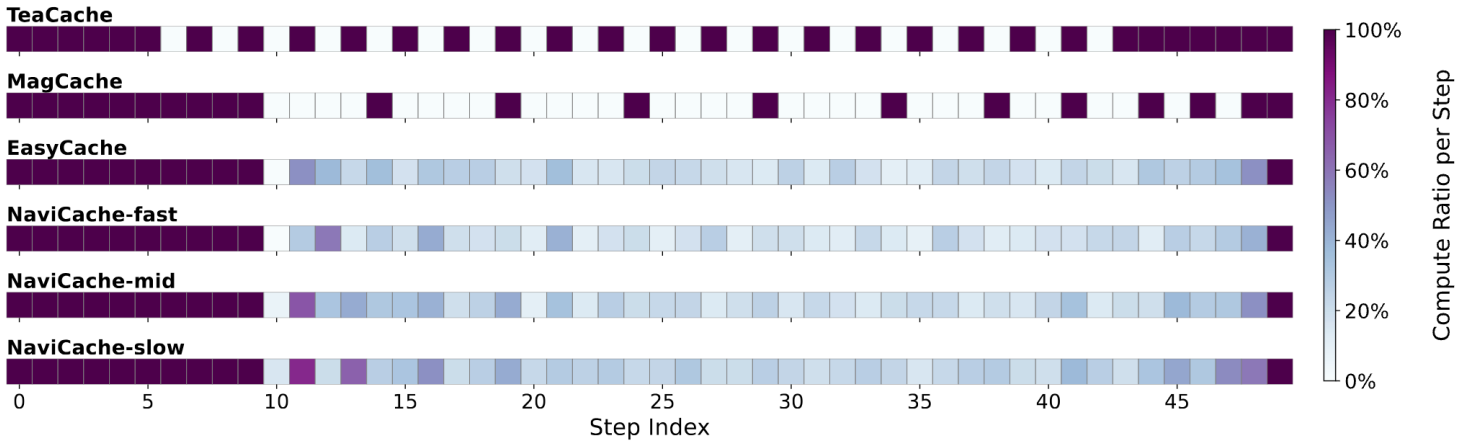


Figure 4. Comparison of skip frequency across timesteps based on Wan 2.1. The distribution illustrates how *NaviCache* adaptively allocates computation for individual samples.

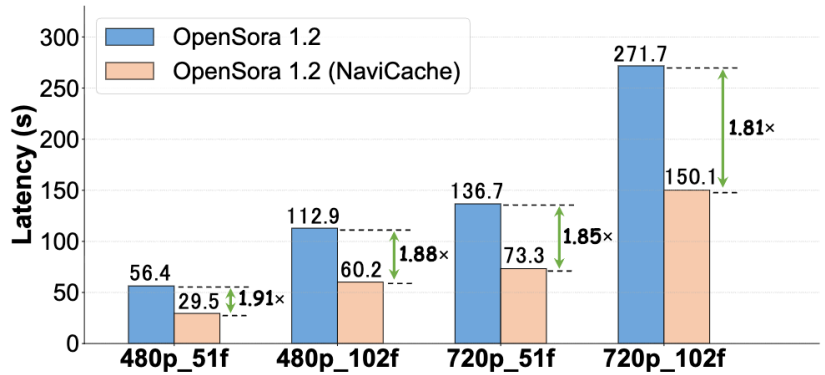


Figure 5. Inference latency under varying spatial and temporal configurations. Evaluations are conducted on 40 VBench prompts, with "p" and "f" denoting pixels and frames, respectively.

Depth analysis

1. Skip frequency varies by timestep and mode, indicating prompt-aware compute allocation.
2. Spatial-temporal scaling remains stable: Open-Sora speedups stay around 1.81×–1.91×.
3. Auxiliary overhead is negligible: 0s offline calibration and 0.0109s skip justification.

This supports the central claim: acceleration should be adaptive, uncertainty-aware, and deployment-friendly.

Takeaways: a principled path toward real-time video generation

NaviCache reframes caching as navigation, and uncertainty as the guardrail.

1. Problem: Repeated DiT execution is the core bottleneck of video diffusion inference.

2. Limitation: Existing caches either rely on offline calibration or use zero-order offline calibration-free reuse.

3. Method: Initial Alignment + Dual-State Self-Calibration + Uncertainty-aware Safety Gate.

4. Impact: Strong speed-fidelity balance across Wan, HunyuanVideo, and Open-Sora with zero offline calibration.

Thank You