



WorldCache: Content-Aware Caching for Accelerated Video World Models

Umair Nawaz^{1*}, Ahmed Heakl¹, Ufaq Khan¹, Abdelrahman Shaker¹,
Salman Khan¹, and Fahad Shahbaz Khan^{1,2}

¹ Mohamed bin Zayed University of Artificial Intelligence, UAE

² Linköping University, Sweden



Project Page: umair1221.github.io/World-Cache



Github: github.com/umair1221/WorldCache

Abstract. Diffusion Transformers (DiTs) power high-fidelity video world models but remain computationally expensive due to sequential denoising and costly spatio-temporal attention. Training-free feature caching accelerates inference by reusing intermediate activations across denoising steps; however, existing methods largely rely on a Zero-Order Hold assumption i.e., reusing cached features as static snapshots when global drift is small. This often leads to ghosting artifacts, blur, and motion inconsistencies in dynamic scenes. We propose **WorldCache**, a Perception-Constrained Dynamical Caching framework that improves both *when* and *how* to reuse features. WorldCache introduces motion-adaptive thresholds, saliency-weighted drift estimation, optimal approximation via blending and warping, and phase-aware threshold scheduling across diffusion steps. Our cohesive approach enables adaptive, motion-consistent feature reuse without retraining. On Cosmos-Predict2.5-2B evaluated on PAI-Bench, WorldCache achieves **2.3×** inference speedup while preserving **99.4%** of baseline quality, substantially outperforming prior training-free caching approaches.

1 Introduction

World models predict future visual states that are physically consistent and useful for downstream decision-making, enabling agents to plan and act within simulated environments [33]. Large-scale Diffusion Transformers (DiTs) have become the dominant backbone for such models [6, 30, 31], because spatio-temporal attention over latent tokens captures the long-range dependencies central to world consistency (*e.g.*, object permanence and causal motion). However, this expressiveness comes at a steep computational cost: world-model rollouts require many frames, and each frame is produced by sequentially invoking deep transformer blocks across dozens of denoising steps [7, 24]. The resulting latency is the primary obstacle to interactive world simulation and closed-loop deployment.

* Corresponding author: umair.nawaz@mbzuai.ac.ae

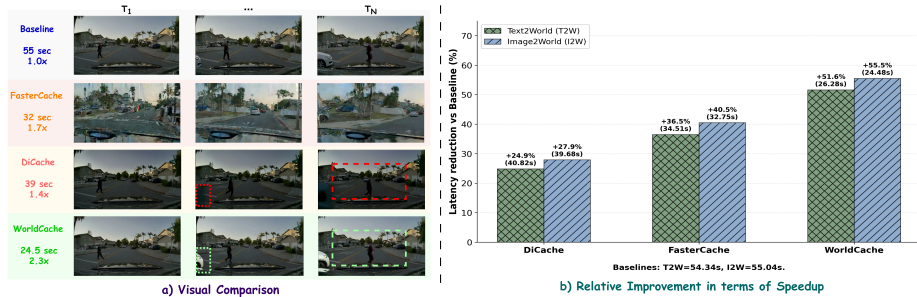


Fig. 1: Qualitative and quantitative comparison of acceleration methods on video world model generation using Cosmos-Predict2.5-2B. **Left:** Visual comparison of Baseline (no acceleration), FasterCache, DiCache, and WorldCache (ours) across three representative timesteps ($T_1, \dots, T_N$) of a driving scene from the City Street domain. FasterCache achieves $1.7\times$ speedup but introduces severe visual artifacts and scene hallucinations. DiCache ($1.4\times$) better preserves scene fidelity but exhibits noticeable spatial artifacts at later timesteps (red dashed boxes). WorldCache (ours) achieves $2.3\times$ speedup while faithfully reproducing scene content, motion, and background structure across all timesteps (green dashed boxes). **Right:** Relative latency reduction (%) over Baseline for Text2World (T2W) and Image2World (I2W) tasks, showing WorldCache achieves up to 55.5% reduction compared to 40.5% for FasterCache and 27.9% for DiCache. All results are reported on a single H200 GPU.

A natural remedy is to exploit redundancy along the denoising trajectory. Consecutive steps often produce only small changes in intermediate features [9], so recomputing every block at every step is wasteful. Training-free caching methods exploit this observation: they estimate a step-to-step drift using a lightweight probe, then skip expensive layers when drift falls below a threshold, reusing cached activations instead. FasterCache [21] applies this idea to video DiTs with a fixed skip schedule, and DiCache [5] makes it adaptive via shallow-layer probes that decide both *when* and *how* to reuse cached states. For world models, however, this “skip-and-reuse” paradigm fails precisely where it matters most: scenes with significant motion and salient interactions [17]. The failure has a single root cause. Existing methods treat cache reuse as a *zero-order hold*: when probe drift is small, they copy stale features verbatim into the next step. Under motion, this produces ghosting, semantic smearing, and incoherent trajectories (as shown in Fig. 1), exactly the artifacts that break world-model rollouts, where errors compound across autoregressive generation. Three specific blindspots make the problem worse. *First*, global drift metrics average over the entire spatial map, so a static background can mask large foreground changes, causing the method to skip when it should recompute. *Second*, all spatial locations are weighted equally, even though errors on salient entities (agents, hands, manipulated objects) dominate both perceptual and functional quality. *Third*, a single static threshold ignores that early denoising steps establish global structure while late steps only refine high-frequency detail; a threshold tuned for the early phase becomes wastefully conservative in the late phase.

We propose **WorldCache**, a training-free caching framework that replaces the zero-order hold with a *perception-constrained dynamical approximation* designed for DiT-based world models. WorldCache addresses each blindspot above with a lightweight, composable module. Causal Feature Caching (CFC) adapts the skip threshold to latent motion magnitude, preventing stale reuse during fast dynamics. Saliency-Weighted Drift (SWD) reweights the probe signal toward perceptually important regions, so caching decisions reflect foreground fidelity rather than background noise. Optimal Feature Approximation (OFA) replaces verbatim copying with least-squares optimal blending and motion-compensated warping, reducing approximation error when skipping does occur. Adaptive Threshold Scheduling (ATS) progressively relaxes the threshold during late denoising, where aggressive reuse is both safe and highly effective. Together, these modules convert caching from a brittle shortcut into a controlled approximation strategy aligned with world-model requirements.

On the Physical AI Bench (PAI-Bench) [34], WorldCache achieves a $2.3\times$ speedup on Cosmos-Predict2.5 (2B) while preserving 99.4% of baseline quality, outperforming both DiCache and FasterCache in speed-quality trade-off. Our contributions are:

1. We formalize feature caching for DiT-based world models as a *dynamical approximation* problem and identify the zero-order hold assumption in prior methods as the primary source of ghosting, blur, and motion incoherence in dynamic rollouts.
2. We introduce WorldCache, a unified framework that improves both *when* to skip (motion and saliency-aware decisions) and *how* to approximate (optimal blending and motion compensation), while adapting to the denoising phase.
3. We demonstrate state-of-the-art training-free acceleration on multiple DiT backbones, achieving up to $2.3\times$ speedup with 99.4% quality retention on Cosmos-Predict2.5, and show that the approach transfers across model scales and conditioning modalities.

2 Related Work

Video diffusion and world simulators. Diffusion models have become a leading approach for high-fidelity video generation, from early formulations [12] to scalable latent/cascaded pipelines [3, 10, 11] and large-scale text-to-video systems [27]. Recently, video generation models have also been studied as *world simulators*, evaluated for physical consistency and action-relevant prediction [25]. In this direction, NVIDIA’s Cosmos platform/Cosmos-Predict target physical AI simulation [1, 2], with benchmarks such as PAI-Bench to assess physical plausibility and controllability [34]. Related efforts include interactive environment world models [4] and large token-based models for video generation [15].

Caching and reuse in diffusion transformers. Caching methods exploit redundancy across denoising timesteps, transformer layers, and guidance passes to

reduce inference cost without retraining. DeepCache [22] reuses high-level features across adjacent steps, primarily in U-Net diffusion models. For video diffusion transformers, FasterCache [21] reuses attention features across timesteps and introduces CFG-Cache to exploit redundancy between conditional and unconditional branches. DiCache [5] further makes caching adaptive by using an online probe to decide *when* to refresh and a trajectory-aligned reuse strategy to decide *how* to combine cached states. More recent methods explore complementary cache-selection rules. DuCa [37] combines aggressive and conservative caching cycles, TeaCache [18] uses timestep-embedding-aware differences to determine cache reuse, EasyCache [35] performs runtime-adaptive reuse of transformation vectors, MagCache [23] leverages magnitude patterns in residual outputs, and TaylorSeer [19] predicts future features using Taylor expansion instead of directly copying cached activations. These methods improve the speed-quality trade-off of diffusion inference, but they are not specifically designed for physical world-model rollouts, where motion, salient object interactions, and compounding temporal errors make stale cache reuse especially harmful. WorldCache is complementary to this line of work as it introduces motion-adaptive cache decisions, saliency-weighted drift estimation, motion-compensated feature approximation, and adaptive threshold scheduling to make caching more robust for dynamic video world models.

Motion-compensated and perception-aware reuse. Feature reuse has also been explored in video recognition via propagation with optical flow [36] and multi-rate update schedules [26], motivating alignment-aware reuse rather than fixed-coordinate copying. Classic and modern flow methods (Lucas–Kanade, RAFT) [20, 28] illustrate the accuracy/efficiency trade-off for motion compensation. Perceptual quality can be tracked with deep perceptual metrics and structure/texture-aware measures [8, 32], while Laplacian pyramids provide a classical multi-scale view of high-frequency detail. WorldCache builds on these ideas with motion-aligned reuse, saliency-aware monitoring, and principled temporal extrapolation.

3 Method

3.1 Preliminaries: DiT Denoising in World Models

We consider a DiT-based world model that predicts future visual states by iteratively denoising a latent video representation. Let $\mathbf{z}_t \in \mathbb{R}^{B \times T \times H \times W \times D}$ denote the latent tensor at denoising step t (not to be confused with video frame index), where B is batch size, T is the number of latent frames, $H \times W$ is spatial resolution, and D is the channel dimension. The denoiser is a stack of N transformer blocks $\{F_i\}_{i=1}^N$, producing $\mathbf{z}_t^{(i)} = F_i(\mathbf{z}_t^{(i-1)})$, with $\mathbf{z}_t^{(0)} = \mathbf{z}_t$ and $\mathbf{z}_t^{(N)}$ used by the sampler to obtain $\mathbf{z}_{t+1}$. Throughout this section, superscripts in parentheses denote layer indices and subscripts denote denoising steps.

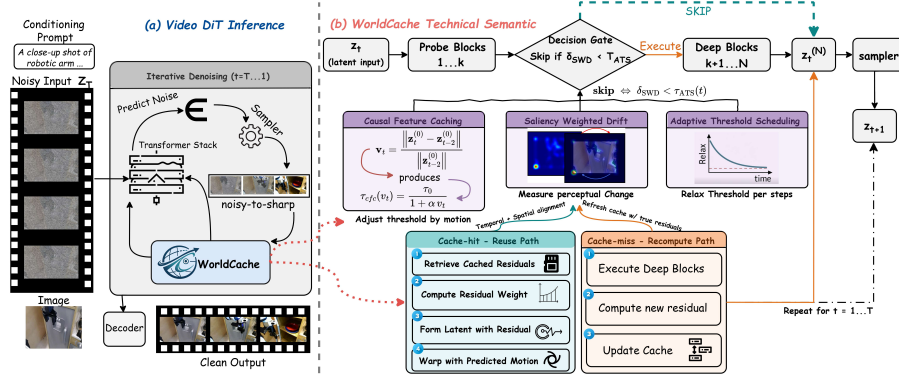


Fig. 2: A DiT world model denoises latent video states (z_t) by running probe blocks ($1 \dots k$) followed by deep blocks ($k+1 \dots N$). WorldCache inserts a decision gate that skips deep blocks when the saliency-weighted probe drift (δ_{SWD}) is below a motion and step-adaptive threshold ($\tau_{\text{ATS}}(t)$) (computed by CFC+ATS), enabling cache hits. On a cache hit, OFA reuses computation by aligning cached residuals via optimal interpolation and optional motion-compensated warping to approximate ($z_t^{(N)}$). On a cache miss, the model executes deep blocks, updates the residual cache in a ping-pong buffer, and continues the denoising loop for ($t=1 \dots T$).

3.2 Foundation: Probe-Then-Cache

WorldCache inherits its architectural skeleton from the probe-then-cache paradigm introduced by DiCache [5], but *replaces* both its skip criterion and its reuse mechanism. We first describe the shared skeleton, then identify the two components we redesign.

Probe (inherited). At each step t , only the first k blocks (probe depth) are evaluated to obtain $z_t^{(k)}$. A drift indicator approximates the deep-layer change:

$$\delta_t = \frac{\|z_t^{(k)} - z_{t-1}^{(k)}\|_1}{\|z_{t-1}^{(k)}\|_1 + \epsilon}. \quad (1)$$

If δ_t falls below a threshold, blocks $k+1, \dots, N$ are skipped and cached deep states are reused; otherwise the full network executes and the cache is refreshed.

Skip criterion (replaced). DiCache uses a fixed global threshold on δ_t . WorldCache replaces this with motion-adaptive, saliency-weighted decisions (Secs. 3.4–3.5).

Reuse mechanism (replaced). DiCache estimates a scalar blending coefficient from L1 residual ratios and interpolates between cached states from steps $t-1$ and $t-2$. This captures the *magnitude* of feature evolution but discards directional information. WorldCache replaces this with a vector-projection-based approximation and optional motion-compensated warping (Sec. 3.6).

3.3 WorldCache Overview

Fig. 2 summarizes the full pipeline. At each denoising step t , the probe computes shallow features $\mathbf{z}_t^{(k)}$. **CFC** (Sec. 3.4) and **SWD** (Sec. 3.5) jointly determine whether to skip by combining a motion-adaptive threshold with a saliency-weighted drift signal. On a cache hit, **OFA** (Sec. 3.6) approximates the deep output via least-squares optimal blending and optional spatial warping. **ATS** (Sec. 3.7) modulates the skip threshold across the denoising trajectory, tightening it during structure-formation steps and relaxing it during late refinement. All four modules are training-free and add negligible overhead to the probe computation.

3.4 Causal Feature Caching (CFC): Motion-Adaptive Decisions

When is reuse safe? In world-model video, the amount of motion varies substantially across prompts and across denoising steps. A fixed threshold is overly permissive during fast motion (risking ghosting) and overly conservative during static intervals (missing speedups). CFC adapts the skip threshold using an inexpensive motion proxy derived from the raw latent input. We define a “velocity” as the normalized two-step input change:

$$v_t = \frac{\left\| \mathbf{z}_t^{(0)} - \mathbf{z}_{t-2}^{(0)} \right\|_1}{\left\| \mathbf{z}_{t-2}^{(0)} \right\|_1 + \epsilon}. \quad (2)$$

We use a two-step gap because step $t-1$ may itself be a cached approximation; anchoring to $t-2$ (the most recent fully-computed input) yields a more reliable velocity estimate. The motion-adaptive threshold is:

$$\tau_{\text{CFC}}(v_t) = \frac{\tau_0}{1 + \alpha \cdot v_t}, \quad (3)$$

where τ_0 is the base threshold and α controls sensitivity. When dynamics are fast (v_t large), τ_{CFC} tightens, making skips less likely; when dynamics are slow, $\tau_{\text{CFC}} \approx \tau_0$. We maintain a ping-pong buffer (two alternating cache slots indexed by step parity) so that reuse is always anchored to one of the two most recent fully-computed states.

3.5 Saliency-Weighted Drift (SWD): Perception-Aware Probing

Is the drift signal measuring the right thing? The global drift δ_t (Eq. 1) treats every spatial location equally, so it cannot distinguish between harmless background fluctuation and critical foreground change. SWD reweights drift toward perceptually important regions, ensuring that the method recomputes when salient content changes and skips when only the background drifts.

We define a spatial saliency map from the channel-wise variance of probe features:

$$S_{h,w} = \text{Var}_d \left(\bar{\mathbf{z}}_t^{(k)}[h, w, :] \right), \quad (4)$$

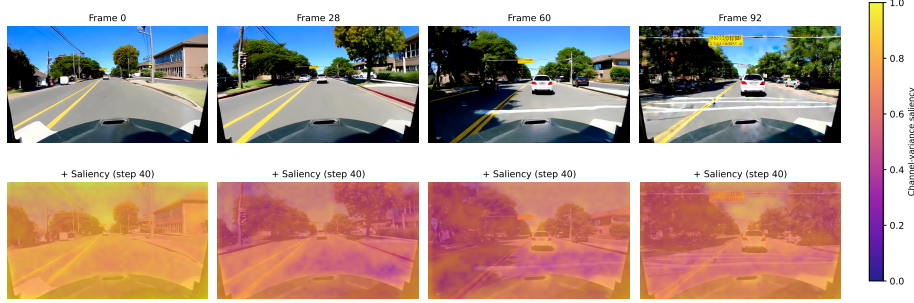


Fig. 3: Saliency overlay (Cosmos-Predict2.5, 2B). Channel-variance saliency from a 70-step run at denoising step 40 overlaid on four video frames. High saliency (yellow) marks structurally complex regions where caching errors are most visible; low saliency (purple) marks smooth areas where reuse is safe. SWD reweights drift using this signal to prioritize fidelity on detail-rich content.

where $\bar{\mathbf{z}}_t^{(k)}$ is the probe output averaged over the batch and temporal axes, and the variance is taken over the channel dimension d . High channel variance indicates spatially complex, information-rich regions (edges, textures, object boundaries) where caching errors are most perceptually visible (Fig. 3). We normalize to $\hat{S} \in [0, 1]$ and define the saliency-weighted drift:

$$\delta_t^{\text{SWD}} = \frac{1}{HW} \sum_{h,w} \left\| \mathbf{z}_t^{(k)}(h, w) - \mathbf{z}_{t-1}^{(k)}(h, w) \right\|_1 \cdot \left(1 + \beta_s \hat{S}_{h,w} \right), \quad (5)$$

where β_s controls saliency emphasis. The weighting term $(1 + \beta_s \hat{S}_{h,w})$ amplifies drift contributions from salient regions and attenuates those from featureless backgrounds. Consequently, a scene where only the static sky changes produces a low δ_t^{SWD} (safe to skip), while one where a foreground agent moves, even slightly, produces a high δ_t^{SWD} (triggering recomputation). The final skip decision combines SWD with the motion-adaptive threshold from CFC:

$$\text{skip} \iff \delta_t^{\text{SWD}} < \tau_{\text{CFC}}(v_t). \quad (6)$$

3.6 Optimal Feature Approximation (OFA): Improved Reuse Quality

When we skip, can we produce a better approximation? CFC and SWD decide *when* to skip. OFA improves *what* is produced on a cache hit, via two complementary operators: one temporal (least-squares optimal blending) and one spatial (motion-compensated warping).

Optimal State Interpolation (OSI) On a cache hit, the deep output $\mathbf{z}_t^{(N)}$ must be approximated from cached history. DiCache [5] estimates a scalar blending coefficient γ from L1 distance ratios between probe residuals. This captures

the magnitude of feature evolution but discards directional information: when motion causes the feature trajectory to curve, the scalar ratio extrapolates along a stale direction, and the resulting errors accumulate over consecutive cache hits.

We reformulate the estimation as a least-squares vector projection. Define the deep computation residual:

$$\mathbf{r}_t = \mathbf{z}_t^{(N)} - \mathbf{z}_t^{(0)}, \quad (7)$$

and on a cache hit, let $\tilde{\mathbf{r}}_t = \mathbf{z}_t^{(k)} - \mathbf{z}_t^{(0)}$ be the probe-derived partial residual. We seek a gain γ^* that best aligns the recent residual trajectory with the current probe signal:

$$\Delta_{\text{tgt}} = \tilde{\mathbf{r}}_t - \mathbf{r}_{t-2}, \quad \Delta_{\text{src}} = \mathbf{r}_{t-1} - \mathbf{r}_{t-2}, \quad (8)$$

$$\gamma^* = \arg \min_{\gamma} \|\Delta_{\text{tgt}} - \gamma \Delta_{\text{src}}\|^2 = \frac{\langle \Delta_{\text{tgt}}, \Delta_{\text{src}} \rangle}{\|\Delta_{\text{src}}\|^2 + \epsilon}. \quad (9)$$

We clamp γ^* to $[0, \gamma_{\text{max}}]$ (we use $\gamma_{\text{max}}=2$) to prevent blow-up when $\|\Delta_{\text{src}}\|$ is small. The deep output is approximated as:

$$\hat{\mathbf{z}}_t^{(N)} = \mathbf{z}_t^{(0)} + \mathbf{r}_{t-2} + \gamma^* (\mathbf{r}_{t-1} - \mathbf{r}_{t-2}). \quad (10)$$

The inner product in Eq. 9 is the key difference from scalar-ratio methods. When the feature trajectory curves (*e.g.*, a moving object changes direction), the dot product naturally attenuates γ^* , preventing extrapolation along a stale direction. When the trajectory is linear, OSI recovers the same estimate as scalar-ratio methods. OSI thus generalizes scalar-ratio alignment; we verify the improvement empirically in the ablation study.

Motion-Compensated Feature Warping OSI corrects *temporal* misalignment in the residual trajectory, but cached features from step $t-1$ may also be *spatially* misaligned when the scene contains motion. OFA optionally warps cached features to the current coordinate frame before applying OSI.

We estimate a displacement field between consecutive latent inputs via multi-scale correlation in latent space (no external network):

$$\mathbf{u}_{t \rightarrow t-1} = \text{LatentCorr}(\mathbf{z}_t^{(0)}, \mathbf{z}_{t-1}^{(0)}), \quad (11)$$

which adds less than 3% overhead per cached step. The cached deep features are then warped:

$$\tilde{\mathbf{z}}_{t-1}^{(N)} = \text{Warp}(\mathbf{z}_{t-1}^{(N)}, \mathbf{u}_{t \rightarrow t-1}), \quad (12)$$

and $\tilde{\mathbf{z}}_{t-1}^{(N)}$ replaces $\mathbf{z}_{t-1}^{(N)}$ in the residual computation of Eq. 10. That is, OSI operates on the spatially-corrected residuals $\tilde{\mathbf{r}}_{t-1} = \tilde{\mathbf{z}}_{t-1}^{(N)} - \mathbf{z}_{t-1}^{(0)}$, reducing compound spatial drift that is especially harmful in autoregressive world-model rollouts. We disable warping during the first five denoising steps, where the low signal-to-noise ratio makes displacement estimation unreliable.

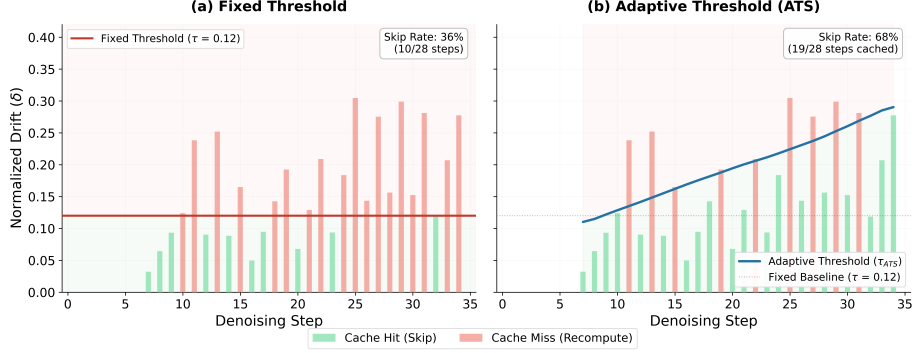


Fig. 4: Comparison of static vs. adaptive caching strategies. (a) A Fixed Threshold ($\tau = 0.12$) fails to accommodate the naturally increasing drift (δ) in later denoising steps, resulting in frequent recomputations and a low skip rate (36%). (b) Our Adaptive Threshold (ATS) dynamically scales τ_{ATS} with the expected drift. This allows ATS to capture significantly more cache hits (green bars) in later stages, nearly doubling the overall skip rate (68%) while maintaining generation quality.

3.7 Adaptive Threshold Scheduling (ATS): Phase-Aware Reuse

Can we push acceleration further without breaking fidelity? The preceding modules (CFC, SWD, OFA) establish a perception-aware caching infrastructure, but they operate with a fixed base threshold τ_0 . The denoising trajectory, however, has two distinct phases. During *structure formation* (early steps, high noise), the network makes large, semantically critical updates that establish global layout and motion. During *detail refinement* (late steps, low noise), updates become small, high-frequency corrections. A static threshold calibrated for the early phase becomes unnecessarily conservative in the late phase: empirically, after step 20 of 35, high-frequency residual variation and accumulated approximation error can push probe drift above the fixed threshold, triggering unnecessary recomputation even when the visual update is safe to approximate (Fig. 4). ATS addresses this mismatch with a step-dependent relaxation:

$$\tau_{ATS}(t) = \tau_{CFC}(v_t) \cdot \left(1 + \beta_d \cdot \frac{t}{T}\right), \quad (13)$$

where $t \in [0, T]$ is the denoising step, T is the total number of steps, and β_d controls the relaxation rate. At an early step (e.g., $t=2$, $T=35$, $\beta_d=4.0$), the multiplier is ≈ 1.2 , keeping the threshold tight and forcing full execution for structure-critical updates. At a late step (e.g., $t=32$), the multiplier reaches ≈ 4.6 , aggressively relaxing the threshold. Since the scene geometry is already established and the network produces only fine texture corrections, the cached OFA approximation is safely reused across consecutive steps. This aggressive late-stage relaxation is the primary source of WorldCache’s speedup gains. The

Table 1: Text2World (T2W) results on PAI-Bench across two model scales. We evaluate four methods, Baseline (no acceleration), DiCache, FasterCache, and WorldCache (ours), on Cosmos-Predict2.5 at both 2B and 14B parameter scales. Avg. denotes the overall score obtained by averaging the Domain and Quality averages. Lat. reports end-to-end wall-clock inference latency in seconds, and Speedup is measured relative to the unaccelerated Baseline. WorldCache achieves the best accuracy-efficiency trade-off with negligible quality degradation.

Method	Domain Score								Quality Score								Avg.	Lat. (s)	Speed	
	CS	AV	RO	IN	HU	PH	MI	Avg	SC	BC	MS	AQ	IQ	OC	IS	IB	Avg	↑	↓	↑
								↑									↑			
Cosmos-Predict2.5 - 2B																				
Baseline	0.75	0.64	0.72	0.81	0.76	0.85	0.84	0.77	0.91	0.93	0.98	0.50	0.71	0.20	0.79	0.81	0.73	0.75	54.3	1.0×
DiCache	0.76	0.63	0.71	0.80	0.77	0.85	0.83	0.76	0.90	0.93	0.98	0.49	0.71	0.20	0.78	0.84	0.73	0.75	40.8	1.3×
FasterCache	0.68	0.55	0.69	0.65	0.72	0.75	0.75	0.66	0.85	0.91	0.97	0.41	0.59	0.18	0.71	0.80	0.68	0.67	34.5	1.6×
WorldCache	0.76	0.64	0.74	0.81	0.76	0.85	0.84	0.77	0.90	0.93	0.98	0.49	0.70	0.20	0.78	0.83	0.73	0.75	26.3	2.1×
Cosmos-Predict2.5 - 14B																				
Baseline	0.78	0.64	0.76	0.83	0.79	0.90	0.88	0.80	0.94	0.95	0.99	0.52	0.72	0.20	0.81	0.85	0.75	0.77	216.3	1.0×
DiCache	0.80	0.65	0.76	0.82	0.79	0.91	0.88	0.80	0.94	0.95	0.99	0.52	0.71	0.20	0.81	0.85	0.75	0.77	148.4	1.5×
FasterCache	0.71	0.56	0.58	0.71	0.68	0.77	0.79	0.69	0.88	0.93	0.98	0.43	0.60	0.18	0.73	0.82	0.70	0.69	126.6	1.7×
WorldCache	0.79	0.66	0.75	0.84	0.79	0.91	0.88	0.80	0.94	0.95	0.99	0.52	0.72	0.20	0.80	0.86	0.75	0.78	98.6	2.2×

ablation in Sec. 4.5 confirms that it reduces quality by less than 0.6% relative to the baseline.

4 Experiments

4.1 Experimental Setup

Base models. We evaluate WorldCache on the Cosmos-Predict2.5 family of Video World Models [2] at two scales: the 2B-parameter and 14B-parameter variants. Both models employ a Video Diffusion Transformer backbone with 3D Rotary Positional Encoding (RoPE) and are trained with a Flow Matching (velocity prediction) objective. Each sampling run generates 93 frames (~ 5.8 s at 16 FPS), corresponding to 24 latent frames, using 35 denoising steps with Euler scheduling. To demonstrate that WorldCache is not specific to a single backbone, we additionally evaluate on **WAN2.1** [29], a DiT-based video generation model, using the official inference configuration for the corresponding checkpoints.

Benchmark. All methods are evaluated on PAI-Bench (Physical AI Benchmark) [34], a comprehensive evaluation suite designed specifically for Video World Models. PAI-Bench spans six physical domains, Robot, Autonomous Vehicles, Human Activity, Industry, Common Sense, and Physics, and reports a *Domain Score* (physical plausibility), a *Quality Score* (visual fidelity), and their average as the *Overall Score*. We report results on both the **Text-to-World (T2W)** and **Image-to-World (I2W)** generation tasks. More details on each evaluation metric are provided in Section A of supplementary material.

Baselines. We compare against two state-of-the-art training-free caching baselines: **DiCache** [5], which employs an online probe profiling scheme with trajectory-aligned residual reuse; and **FasterCache** [21], which uses a fixed step-skipping

Table 2: Image2World (I2W) generation results on PAI-Bench across two model scales. Compared to the T2W setting, I2W scores are generally higher owing to the additional visual grounding provided by the conditioning image. WorldCache achieves the best accuracy-efficiency trade-off consistently with negligible quality degradation.

	Domain Score								Quality Score								Avg. ↑	Lat. (s) ↓	Speedup ↑	
Method	CS	AV	RO	IN	HU	PH	MI	Avg ↑	SC	BC	MS	AQ	IQ	OC	IS	IB	Avg ↑			
Cosmos-Predict2.5 - 2B																				
Baseline	0.92	0.69	0.81	0.88	0.84	0.91	0.89	0.85	0.90	0.93	0.98	0.51	0.67	0.21	0.94	0.95	0.76	0.80	55.04	1.0×
DiCache	0.90	0.70	0.79	0.88	0.83	0.89	0.91	0.84	0.89	0.92	0.98	0.49	0.66	0.21	0.93	0.94	0.75	0.80	39.7	1.4×
FasterCache	0.86	0.68	0.70	0.83	0.74	0.85	0.85	0.79	0.80	0.87	0.97	0.43	0.58	0.20	0.89	0.92	0.71	0.75	32.8	1.7×
WorldCache	0.91	0.71	0.80	0.88	0.83	0.89	0.89	0.84	0.89	0.93	0.98	0.50	0.66	0.21	0.93	0.95	0.76	0.80	24.4	2.3×
Cosmos-Predict2.5 - 14B																				
Baseline	0.92	0.72	0.83	0.91	0.85	0.92	0.92	0.87	0.91	0.94	0.99	0.51	0.67	0.21	0.96	0.97	0.77	0.82	210.1	1.0×
DiCache	0.91	0.72	0.83	0.89	0.84	0.92	0.92	0.86	0.91	0.94	0.99	0.51	0.66	0.21	0.96	0.97	0.77	0.81	146.0	1.4×
FasterCache	0.86	0.69	0.72	0.84	0.74	0.87	0.86	0.80	0.81	0.87	0.98	0.44	0.57	0.20	0.91	0.93	0.71	0.75	123.8	1.7×
WorldCache	0.92	0.73	0.82	0.90	0.85	0.93	0.91	0.87	0.91	0.94	0.99	0.51	0.66	0.21	0.96	0.97	0.77	0.82	99.3	2.1×

schedule combined with CFG-Cache for unconditional branch reuse. All methods are applied as drop-in replacements on the same base model checkpoints. Wall-clock latency is measured on a single NVIDIA H200 GPU with identical batch size and precision settings.

WorldCache configuration. Unless otherwise stated, the full WorldCache pipeline combines four modules: Causal Feature Caching (CFC), Saliency Weighted Drift (SWD), Optimal Feature Approximation (OFA), and Adaptive Threshold Scheduling (ATS). We set the base threshold $\tau_0 = 0.08$, motion sensitivity $\alpha = 0.2$, saliency weight $\beta_s = 0.12$, and $\beta_d=4.0$ across the diffusion trajectory. All hyperparameters are fixed across models and tasks without per-prompt tuning. Unless otherwise stated, we also apply the same caching pipeline and hyperparameter selection protocol to WAN2.1 as in Cosmos, and report results on the same PAI-Bench [34] tasks for direct comparison.

4.2 Main Results

Text2World (T2W) on Cosmos-Predict2.5. Table 1 shows that WorldCache provides the strongest accuracy-efficiency trade-off at both scales. On **Cosmos-2B**, WorldCache reduces latency from 54.34s to 26.28s (**2.1×**) while preserving the overall average score (0.7485 vs. 0.7480; $\sim 99.9\%$ retention). DiCache is less aggressive (40.82s, 1.33 $\times$, Avg. 0.746), while FasterCache is faster (34.51s, $\sim 1.6\times$) but suffers a large quality drop (Avg. 0.665). WorldCache improves over DiCache most clearly in Domain categories tied to dynamic scene structure (e.g., RO, IN) while maintaining nearly identical Quality Avg. (0.727 vs. 0.726). On **Cosmos-14B**, WorldCache again achieves the best frontier point, reaching 98.61s (**2.20×**) and improving Avg. to 0.775 over the 0.772 baseline. DiCache achieves 1.46 $\times$ (148.36s, Avg. 0.772), and FasterCache reaches 1.7 $\times$ at substantially lower Avg. (0.690).

Image2World (I2W) on Cosmos-Predict2.5. As shown in Table 2, I2W scores are higher overall due to additional visual grounding, and WorldCache re-

Table 3: WAN2.1 results on PAI-Bench. We report Domain Score (Avg), Quality Score (Avg), Overall Score (computed as the mean of Domain Avg and Quality Avg), wall-clock latency (seconds), and speedup relative to the unaccelerated baseline. Best accelerated results per task/column are in **bold**.

Task	Method	Domain Avg $\uparrow$	Quality Avg $\uparrow$	Overall $\uparrow$	Latency (s) $\downarrow$	Speedup $\uparrow$
WAN2.1 - 1.3B						
T2W	Baseline	0.7862	0.7592	0.7727	120.04	1.00 $\times$
	DiCache	0.7841	0.7564	0.7703	61.57	1.95 $\times$
	WorldCache (Ours)	0.7853	0.7589	0.7721	50.84	2.36$\times$
	Δ (WC-DC)	+0.0012	+0.0025	+0.0018	-10.73	+0.41 $\times$
WAN2.1 - 14B						
I2W	Baseline	0.7065	0.7703	0.7384	475.60	1.00 $\times$
	DiCache	0.6949	0.7672	0.7311	291.91	1.63 $\times$
	WorldCache (Ours)	0.7069	0.7707	0.7388	206.73	2.30$\times$
	Δ (WC-DC)	+0.0120	+0.0035	+0.0077	-85.18	+0.67 $\times$

mains the best speed-quality trade-off. On **Cosmos-2B**, WorldCache achieves **2.3 $\times$** speedup (55.04 s $\rightarrow$ 24.48 s) with Avg. 0.80, close to the baseline 0.804. DiCache is slower ($\sim 1.4\times$, 39.7 s, Avg. 0.797), while FasterCache is faster ($\sim 1.7\times$) but drops sharply in Avg. (0.75). Here, WorldCache again outperforms DiCache across most Domain categories and boosts perceptual consistency metrics (SC/BC/MS) at a similar OC. On **Cosmos-14B**, WorldCache delivers **2.12 $\times$** speedup (210.1 s $\rightarrow$ 99.3 s) with negligible change in Avg. (0.817 vs. 0.816), outperforming DiCache (1.4 $\times$, Avg. 0.81) and FasterCache (1.7 $\times$, Avg. 0.75).

Transfer to WAN2.1. Table 3 demonstrates that the benefits of WorldCache transfer beyond Cosmos to WAN2.1. On **WAN2.1-1.3B (T2W)**, WorldCache improves over DiCache in both efficiency and score: 61.57 s $\rightarrow$ 50.84 s (1.95 $\times$ $\rightarrow$ **2.36 $\times$**) while increasing overall score from 0.7703 to 0.7721. On **WAN2.1-14B (I2W)**, WorldCache yields a large latency reduction over DiCache (291.91 s $\rightarrow$ 206.73 s; 1.63 $\times$ $\rightarrow$ **2.3 $\times$**) and recovers overall score to 0.7388, essentially even surpassing the baseline 0.7384. These results confirm that WorldCache consistently improves the DiCache frontier, trading lower fidelity for substantially greater speed across both conditioning modalities and model families. More results on the EgoDex benchmark [13] are included in the supplementary material.

Qualitative Results. Fig. 5 shows that DiCache introduces temporal artifacts (object ghosting, inconsistent motion) in dynamic scenes (red dashed boxes), whereas WorldCache maintains coherent object appearance and consistent trajectories (green dashed boxes) while accelerating inference. More qualitative examples are in the supplementary material. Across both T2W and I2W and at both 2B and 14B scales, we observe consistent gains on the speed-quality frontier, and the WAN2.1 results show these benefits transfer across model families, supporting perception- and dynamics-constrained caching as a generally useful inference primitive for world models. In short, the recipe is to invest in decision and approximation quality early, then exploit the refinement phase to harvest large speedups with minimal impact on temporal coherence.

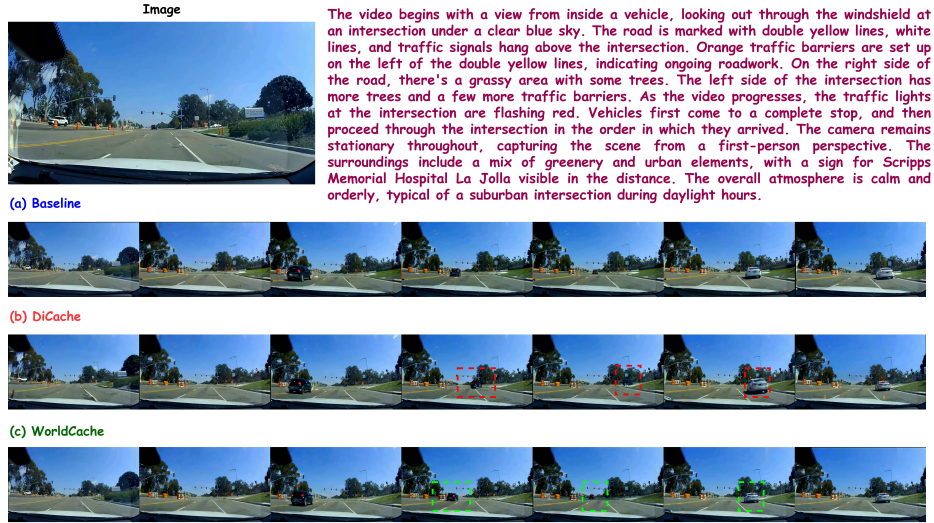


Fig. 5: Qualitative Image2World comparison (PAI-Bench). Given the same conditioning image and text description, we show evenly-spaced frames from the generated rollout for (a) Baseline, (b) DiCache, and (c) WorldCache. DiCache exhibits temporal artifacts under dynamic scene evolution, e.g., object ghosting/deformation and inconsistent motion of vehicle and foreground structures (highlighted with **red dashed boxes**). WorldCache preserves the scene layout and maintains more coherent object appearance and trajectories over time (highlighted with **green dashed boxes**), while achieving substantial inference acceleration. Better viewed zoomed in.

4.3 More Caching Comparison

To further contextualize WorldCache against recent training-free caching baselines, we also compare with EasyCache [35], TeaCache [18], MagCache [23], and TaylorSeer [19], under the same Cosmos-Predict2.5-2B setup for Image2World (I2W) on PAI-Bench (Table 4). Among prior caching baselines, EasyCache gives the best Domain score (0.8399), while TeaCache (Slow) gives the best Quality and Overall scores (0.7562 and 0.7979), but only reaches $1.10\times$ speedup. In contrast, WorldCache achieves the lowest latency, reducing runtime from 55.04 s to 24.48 s ($\sim 2.3\times$), while maintaining a competitive Overall score of 0.7977 and only being 0.0002 below the best accelerated quality result. This shows that WorldCache offers the strongest efficiency–quality trade-off, exceeding $2\times$ speedup without leaving the top quality band.

4.4 VBench Evaluation

We further evaluate WorldCache on VBench [14] using HunyuanVideo [16] and Cosmos-Predict2.5-2B (Table 5). On HunyuanVideo, WorldCache achieves the best accelerated VBench score (82.23) while reaching $2.95\times$ speedup, outperforming EasyCache, TeaCache, and MagCache in efficiency. On Cosmos-Predict2.5-2B, WorldCache improves the VBench score over the unaccelerated baseline

Table 4: PAI-Bench comparison with other caching methods (Cosmos-2B-I2W). We report Domain, Quality, Overall, latency (s), and speedup vs. baseline for both Image2World (I2W). **Bold** and underline denote the best and second-best *among accelerated methods only* (excluding Baseline) within each block.

Method	Domain $\uparrow$	Quality $\uparrow$	Overall $\uparrow$	Lat. (s) $\downarrow$	Speedup $\uparrow$
Image2World (I2W) — Cosmos-Predict2.5-2B					
Baseline	0.8450	0.7610	0.8030	55.04	1.0 $\times$
EasyCache	0.8399	0.7552	0.7975	40.25	1.37 $\times$
DiCache	0.8352	0.7522	0.7941	39.68	1.39 $\times$
TeaCache (Fast)	0.8381	0.7549	0.7965	41.00	1.34 $\times$
TeaCache (Slow)	<u>0.8396</u>	0.7562	0.7979	49.59	1.10 $\times$
MagCache	0.7682	0.7410	0.7546	26.87	2.05 $\times$
TaylorSeer	0.7106	0.6956	0.7031	29.49	1.86 $\times$
WorldCache (Ours)	0.8395	<u>0.7559</u>	<u>0.7977</u>	24.48	2.30$\times$

Table 5: VBench comparison on HunyuanVideo and Cosmos-Predict2.5-2B. We report VBench score, speedup relative to the unaccelerated baseline, and wall-clock latency. Bold indicates the best accelerated method in each block.

Method	VBench $\uparrow$	Speedup $\uparrow$	Latency (s) $\downarrow$
HunyuanVideo			
Baseline	82.34	1.00 $\times$	984
EasyCache	81.98	2.03 $\times$	485
TeaCache	81.46	2.52 $\times$	391
MagCache	82.18	2.05 $\times$	480
TaylorSeer	Out-of-Memory		
WorldCache (Ours)	82.23	2.95$\times$	334

Method	VBench $\uparrow$	Speedup $\uparrow$	Latency (s) $\downarrow$
Cosmos-Predict2.5-2B			
Baseline	76.35	1.00 $\times$	54.87
EasyCache	75.66	1.30 $\times$	42.14
DiCache	74.00	1.46 $\times$	37.68
MagCache	76.29	2.11 $\times$	26.05
TaylorSeer	75.70	1.85 $\times$	29.71
WorldCache (Ours)	76.47	2.25$\times$	24.42

(76.47 vs. 76.35) while achieving the lowest latency and highest speedup among all accelerated methods (24.42s, 2.25 $\times$). These results show that WorldCache can generalize well beyond PAI-Bench and maintain strong perceptual quality under standard video-generation evaluation.

4.5 Ablation Study

To isolate the contribution of each WorldCache module, we conduct incremental ablations on the 2B model under the I2W setting (Table 6). Besides, we also provide further ablations related to the key components in supplementary.

CFC improves safety under motion. Adding CFC yields a 1.52 $\times$ speedup (55 $\rightarrow$ 36s) with essentially unchanged overall score (0.8020 vs. 0.8027). This indicates that motion-adaptive thresholding is an effective first-order control signal: it preserves fidelity by tightening reuse during fast dynamics, while still enabling frequent cache hits in stable intervals.

SWD increases cache hits by focusing on salient regions. With SWD, speed increases further to 1.70 $\times$ (33.24s). Although the global scores remain close to baseline, SWD improves decision quality by preventing background drift from dominating the skip criterion, thereby yielding additional cache hits without sacrificing foreground stability.

Table 6: Incremental ablation on Cosmos-Predict2.5-2B (I2W). Each row adds one module to the previous configuration. *Legend:* CFC = Causal Feature Caching (motion-adaptive thresholding); SWD = Saliency-Weighted Drift; OFA = Optimal Feature Approximation (Optimal State Interpolation); ATS = Adaptive Threshold Scheduling (dynamic threshold decay).

Configuration	Domain $\uparrow$	Quality $\uparrow$	Overall $\uparrow$	Speedup $\uparrow$	Latency (s) $\downarrow$
Base	0.8447	0.7607	0.8027	1.00 $\times$	55.04
+ CFC	0.8457	0.7583	0.8020	1.52 $\times$	36.06
+ CFC + SWD	0.8414	0.7592	0.8003	1.70 $\times$	33.24
+ CFC + SWD + OFA	0.8468	0.7602	0.8035	1.47 $\times$	37.38
+ CFC + SWD + OFA + ATS (WorldCache)	0.8395	0.7559	0.7977	2.3$\times$	24.48

OFA “invests” in approximation quality. Introducing OFA yields the highest fidelity configuration (Overall 0.8035), slightly exceeding the baseline. The improved approximation, least-squares optimal blending (and motion-aligned reuse when enabled), reduces error accumulation on cache hits. This comes with added overhead and a more conservative hit pattern, which explains the reduced net speedup (1.47 $\times$, 37.38s). In other words, OFA intentionally trades some throughput to raise the quality margin.

ATS “spends” the quality margin for speed. Finally, ATS unlocks the largest acceleration by relaxing thresholds late in denoising, where refinement updates are small. With the stabilizing effects of CFC+SWD+OFA in place, ATS increases cache hits in the low-noise phase, improving speed to **2.30 $\times$** (24.48s) while keeping overall within 0.6% of baseline (0.7977 vs. 0.8027), matching the intended invest-and-spend effect.

5 Conclusion

We presented **WorldCache**, a unified framework for *perception-constrained dynamical caching* in DiT-based video generation. By identifying the Zero-Order Hold assumption as the core source of artifacts in prior diffusion caching methods, we redesigned caching around motion-aware decisions (CFC), saliency-aligned drift estimation (SWD), improved approximation operators (OFA), and denoising-phase scheduling (ATS). WorldCache requires no training and no architectural changes, making it immediately deployable for accelerating next-generation video world models.

Acknowledgements

The computations were enabled by resources provided by LUMI hosted by CSC (Finland) and LUMI consortium, and by Berzelius resource provided by the Knut and Alice Wallenberg Foundation at the NSC.

References

1. Agarwal, N., et al.: Cosmos world foundation model platform for physical ai. arXiv preprint arXiv:2501.03575 (2025)
2. Ali, A., et al.: World simulation with video foundation models for physical ai. arXiv preprint arXiv:2511.00062 (2025)
3. Blattmann, A., Others: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023)
4. Bruce, J., et al.: Genie: Generative interactive environments. ICML (2024)
5. Bu, J., Ling, P., Zhou, Y., Wang, Y., Zang, Y., Lin, D., Wang, J.: Dicache: Let diffusion model determine its own cache. In: ICLR (2026)
6. Chen, S., et al.: Gentron: Diffusion transformers for image and video generation. In: CVPR (2024)
7. Chi, X., et al.: Mind: Learning a dual-system world model for real-time planning and implicit risk analysis. arXiv preprint arXiv:2506.18897 (2025)
8. Ding, K., Ma, K., Wang, S., Simoncelli, E.P.: Image quality assessment: Unifying structure and texture similarity. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
9. Fuest, M., Ma, P., Gui, M., Schusterbauer, J., Hu, V.T., Ommer, B.: Diffusion models and representation learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (2026)
10. He, Y., Yang, T., Zhang, Y., Shan, Y., Chen, Q.: Latent video diffusion models for high-fidelity long video generation. arXiv preprint arXiv:2211.13221 (2022)
11. Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D.P., Poole, B., Norouzi, M., Fleet, D.J., Salimans, T.: Imagen video: High definition video generation with diffusion models. arXiv preprint arXiv:2210.02303 (2022)
12. Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models. In: NeurIPS (2022)
13. Hoque, R., Huang, P., Yoon, D.J., Sivapurapu, M., Zhang, J.: Egodex: Learning dexterous manipulation from large-scale egocentric video. arXiv preprint arXiv:2505.11709 (2025)
14. Huang, Z., et al.: Vbench: Comprehensive benchmark suite for video generative models. In: CVPR (2024)
15. Kondratyuk, D., et al.: Videopoet: A large language model for zero-shot video generation. In: ICML (2024)
16. Kong, W., et al.: Hunyuanvideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603 (2024)
17. Li, X., He, X., Zhang, L., Wu, M., Li, X., Liu, Y.: A comprehensive survey on world models for embodied ai. arXiv preprint arXiv:2510.16732 (2025)
18. Liu, F., Zhang, S., Wang, X., Wei, Y., Qiu, H., Zhao, Y., Zhang, Y., Ye, Q., Wan, F.: Timestep embedding tells: It’s time to cache for video diffusion model. In: CVPR (2025)
19. Liu, J., Zou, C., Lyu, Y., Chen, J., Zhang, L.: From reusing to forecasting: Accelerating diffusion models with taylorseers. In: ICCV (2025)
20. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI). vol. 2, pp. 674–679 (1981)
21. Lv, Z., Si, C., Song, J., Yang, Z., Qiao, Y., Liu, Z., Wong, K.Y.K.: Fastercache: Training-free video diffusion model acceleration with high quality. In: ICLR (2025)

22. Ma, X., Fang, G., Wang, X.: Deepcache: Accelerating diffusion models for free. In: CVPR (2024)
23. Ma, Z., Wei, L., Wang, F., Zhang, S., Tian, Q.: Magcache: Fast video generation with magnitude-aware cache. *Advances in Neural Information Processing Systems* (2025)
24. Ma, Z., Zhang, Y., Jia, G., Zhao, L., Ma, Y., Ma, M., Liu, G., Zhang, K., Ding, N., Li, J., et al.: Efficient diffusion models: A comprehensive survey from principles to practices. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025)
25. Qin, Y., et al.: Worldsimbench: Towards video generation models as world simulators. *ICML* (2025)
26. Shelhamer, E., Rakelly, K., Hoffman, J., Darrell, T.: Clockwork convnets for video semantic segmentation. In: *ECCV* (2016)
27. Singer, U., et al.: Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792* (2022)
28. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: *ECCV* (2020)
29. Wan, T., et al.: Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314* (2025)
30. Wang, Z., Xia, X., Chen, R., Yu, D., Wang, C., Gong, M., Liu, T.: Lavin-dit: Large vision diffusion transformer. In: *CVPR* (2025)
31. Yang, Z., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. In: *ICLR* (2025)
32. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018)
33. Zhao, C., Zhang, R., Wang, J., Zhao, G., Niyato, D., Sun, G., Mao, S., Kim, D.I.: World models for cognitive agents: Transforming edge intelligence in future networks. *arXiv preprint arXiv:2506.00417* (2025)
34. Zhou, F., Huang, J., Li, J., Ramanan, D., Shi, H.: Pai-bench: A comprehensive benchmark for physical ai. *CVPR* (2026)
35. Zhou, X., Liang, D., Chen, K., Feng, T., Chen, X., Lin, H., Ding, Y., Tan, F., Zhao, H., Bai, X.: Less is enough: Training-free video diffusion acceleration via runtime-adaptive caching. *arXiv preprint arXiv:2507.02860* (2025)
36. Zhu, X., Xiong, Y., Dai, J., Yuan, L., Wei, Y.: Deep feature flow for video recognition. In: *CVPR* (2017)
37. Zou, C., Zheng, S., Zhang, E., Guo, R., Xu, H., Shi, Z., He, C., Hu, X., Zhang, L.: Rethinking token-wise feature caching: Accelerating diffusion transformers with dual feature caching. *IEEE Transactions on Image Processing* (2026)