ACTCACHE: ACTION-GUIDED CROSS-CHUNK KV CACHE REUSE FOR INTERACTIVE WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Interactive world models generate continuous visual environments chunk by chunk while maintaining historical KV caches for temporal coherence. Although few-step distillation reduces denoising iterations, dense computation within each new chunk remains a major barrier to real-time interaction, while conventional step-wise caching offers limited reuse opportunities. We observe that adjacent chunks preserve substantial visual content and that explicit controls constrain its motion, making historical KV representations a structured source of reusable computation. Based on this insight, we introduce **ActCache**, an action-guided framework for fine-grained cross-chunk KV reuse. ActCache converts motion-related controls into coarse displacement priors and restricts correspondence search to layer-adaptive neighborhoods. Since these priors cannot fully capture parallax, object motion, occlusion, or newly revealed content, ActCache uses the initial clean estimate from the first denoising step as a self-generated probe to verify token-level correspondences. Verified tokens reuse historical self-attention representations, while unmatched regions are recomputed. A shift-equivariant calibration mechanism applies relative 3D-RoPE rotations to align reused keys with their current coordinates. Without additional training, ActCache delivers up to $1.43\times$ speedup across three interactive world models while preserving world-memory consistency. Combined with complementary acceleration strategies, it further achieves up to $1.75\times$ speedup.

1 INTRODUCTION

Interactive world models (Team et al., 2026b;a; Wang et al., 2026; Zheng et al., 2026; Hong et al., 2025) have emerged as powerful simulators for physical environments, generating observations responding to user action conditions. To enable infinite generation, these models commonly adopt an autoregressive, chunk-wise paradigm (Yin et al., 2025; Huang et al., 2026b). In this paradigm, each video chunk undergoes an iterative denoising phase, followed by a memory appending phase—an additional forward pass dedicated to storing the cleaned chunk into a historical KV cache pool. While modern interactive world models utilize few-step distillation (Yin et al., 2024) to meet basic throughput demands, computing full spatial-temporal attention across an ever-growing KV cache during the denoising of each high-resolution chunk remains a critical bottleneck for real-time applications.

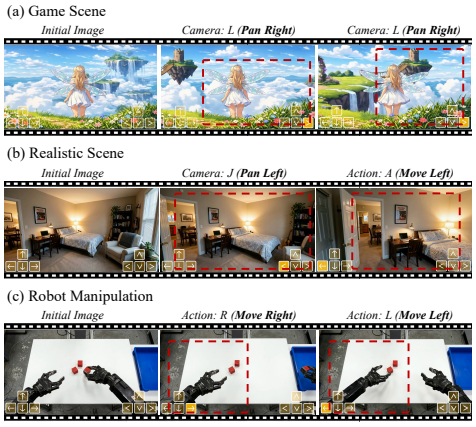
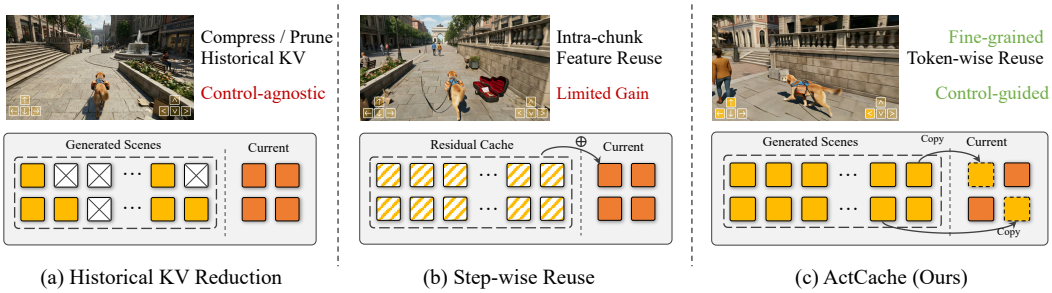


Figure 1: **Cross-chunk redundancy under interactive controls.** Consecutive frames share substantial content. Red dashed boxes highlight regions similar to the previous frame, and keyboard icons denote `WASD` navigation and `IJKL` camera panning.

To mitigate these costs, a range of acceleration techniques has emerged, which can be broadly categorized into two distinct paths, as illustrated in Figure 2. The first path focuses on *historical-context reduction*. Methods like WorldKV (Yi et al., 2026),

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.



(a) Historical KV Reduction (b) Step-wise Reuse (c) ActCache (Ours)

Figure 2: **Acceleration paradigms for autoregressive world models.** (a) Historical KV reduction compresses or prunes past context to reduce memory and attention costs, but does not reuse computation for the current chunk. (b) Step-wise reuse caches features across denoising steps, whose benefit is limited in few-step models. (c) ActCache performs control-guided, token-wise KV reuse across consecutive chunks, copying verified correspondences and recomputing unmatched regions.

Dummy Forcing (Guo et al., 2026), and Forcing-KV (Ji et al., 2026) compress historical KV cache pools or prune redundant attention heads to reduce storage and memory access overhead. While prominent speedups are achieved, these methods overlook the potential of exploiting retained historical representations as reusable computation for the current chunk. The second path involves *computation reuse*, including TokenCache (Lou et al., 2024) and ToCa (Zou et al., 2024), which selectively cache token-level representations across adjacent denoising steps. Their effectiveness is weakened in few-step interactive world models, where reuse opportunities are limited and consecutive denoising states undergo larger changes. Reusing computation across video chunks provides a complementary opportunity, but requires identifying fine-grained token correspondences under diverse action-induced spatial transformations. Without an informative structural prior, exhaustive matching incurs a prohibitive complexity of $O(H^2W^2)$.

To address this, we recognize that interactive world models are uniquely governed by explicit physical control signals. As demonstrated in Figure 1, visual transitions between consecutive chunks are not arbitrary, but are structured by user actions and camera motion. This physical prior reveals substantial spatiotemporal redundancy, as large portions of the scene remain similar under control-induced spatial transformations. Therefore, rather than passively guessing redundancy, we can proactively exploit action signals as a deterministic physical prior to predict correspondences. This control-guided routing restricts token matching from the full spatial grid to a compact local neighborhood, avoiding the $O(H^2W^2)$ cost of global search.

To this end, we introduce **ActCache**, an action-guided framework for reliable cross-chunk reuse of historical KV representations. ActCache derives coarse spatial displacements from motion-related controls to restrict correspondence search to layer-adaptive neighborhoods in the historical cache. Since this coarse localization can be invalidated by parallax, object motion, occlusion, and newly revealed content, ActCache uses the initial clean estimate ($\hat{x}_0$) from the first denoising step as a self-generated probe to verify candidate correspondences for reuse in subsequent steps. To correct the positional misalignment of reused keys, a shift-equivariant calibration mechanism applies the relative 3D-RoPE rotation between historical and current coordinates, preserving positional consistency without hardware-unfriendly de-rotation. A fully recomputed memory-appending pass further prevents reuse errors from propagating to future chunks. Across three interactive world models, ActCache achieves up to $1.43\times$ generation speedup with negligible quality degradation. Combining ActCache with historical-cache compression yields further acceleration, demonstrating the complementarity of cross-chunk computation reuse and historical-context reduction.

Our contributions are summarized as follows:

- We formulate historical KV caches as a source of reusable computation for interactive world models and exploit motion-related controls to establish cross-chunk correspondence.
- We introduce ActCache, which integrates motion-guided local search, prediction-probed correspondence verification, and shift-equivariant 3D-RoPE calibration for reliable KV reuse.

- We validate ActCache across three interactive world models, demonstrating consistent acceleration, preserved generation quality, and compatibility with historical-cache compression.

2 Related Work

Interactive World Model. Delivering real-time, spatiotemporally coherent visual feedback, interactive world models broadly span diverse domains from 3D synthesis (Shang et al., 2024; Team et al., 2025; HY-World et al., 2026) to video generation (Huang et al., 2025; Guo et al., 2025; Huang et al., 2026a). Within the interactive video space, early frameworks like Genie (Bruce et al., 2024) map discrete actions for 2D navigation, while its successor (Lin et al., 2024) unlocks prompt-based editing and 3D viewpoint manipulations. Matrix-Game (Zhang et al., 2025b) injects frame-level mouse and keyboard signals for complex sandbox environments, whereas AlayaWorld (Team et al., 2026a) diversifies gaming mechanics with open-ended explorations like spell-casting. Stepping beyond virtual realms, Yume 1.0 (Mao et al., 2025b) and 1.5 (Mao et al., 2025a) ground generation in real-world scenes with text-guided event editing, while Lingbot-World (Team et al., 2026c) and Hunyuan WorldPlay (Sun et al., 2025) leverage image conditions to scale toward consistent, long-horizon rollouts. Despite their diverse applications, these paradigms share a strict adherence to physical dynamics where injected action signals act as explicit kinematic priors. This action-driven progression dictates smooth state transitions, guaranteeing that adjacent visual contexts undergo only deterministic spatial translations or remain strictly preserved, thereby breeding massive spatiotemporal redundancy that our framework directly exploits to accelerate autoregressive decoding.

Efficient Autoregressive Video Diffusion. As modern video world models (Duan et al., 2026; Zhao et al., 2026) commonly adopt Diffusion Transformers (DiTs), accelerating these architectures historically relies on caching-based paradigms. Traditional reuse-based methods exploit intra-chunk temporal redundancy across adjacent denoising steps by caching attention or block features (Selvaraju et al., 2024; Zhao et al., 2025; Chen et al., 2024; Zhang et al., 2025a) or selectively reusing fine-grained informative tokens (Liu et al., 2025a; Zou et al., 2024; Lou et al., 2024), or extrapolating feature trajectories (Liu et al., 2025b). WorldCache (Feng et al., 2026) further assigns token-wise predictors according to trajectory curvature while using a global scheduler to skip full backbone evaluations. These methods primarily target multi-step diffusion, whereas few-step interactive world models provide fewer opportunities for reuse along the denoising trajectory.

Recent methods adapt caching to autoregressive video generation. FlowCache (Ma et al., 2026) makes independent cache-or-recompute decisions for concurrently denoised chunks, accounting for their heterogeneous denoising states. SCOPE (Cui et al., 2026) introduces a step-wise cache–predict–recompute scheduler and restricts computation to active frame intervals under asynchronous denoising. Both methods exploit redundancy across denoising steps and are designed for many-step autoregressive generation. X-Cache (Zeng et al., 2026) applies caching to few-step driving simulation but makes reuse decisions at the block level, limiting its ability to capture fine-grained spatial redundancy. Furthermore, to handle the long-horizon memory specific to autoregressive world models, recent efforts target KV cache compression. WorldKV (Yi et al., 2026) dynamically compresses historical tokens to retain critical context, while Dummy Forcing (Guo et al., 2026) and Forcing-KV (Ji et al., 2026) prune redundant attention heads by classifying them into static and dynamic states. Nevertheless, these memory-centric mechanisms reduce the attention cost of *historical context*, but typically retain full computation for the current chunk. In contrast, ActCache combines action-guided cross-chunk routing with self-speculative verification to identify reusable tokens in few-step interactive world models, thereby skipping their attention computation while recomputing only the regions that change locally.

3 Method

3.1 Overview

Interactive world models synthesize videos through *inter-chunk autoregression* and *intra-chunk diffusion*. Given a sequence of N video chunks $\mathbf{x}^{1:N}$ and their corresponding conditions $\mathbf{c}^{1:N}$, the joint

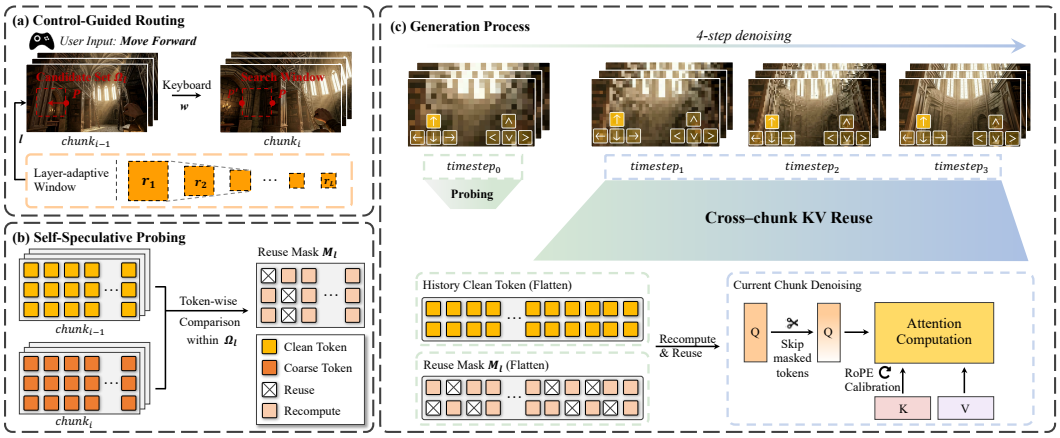


Figure 3: **Overview of ActCache framework.** (a) Control-guided routing converts user inputs into a spatial displacement prior and constructs layer-adaptive local search windows. (b) Self-speculative probing compares the first-step clean prediction with clean tokens from the preceding chunk to derive a token-wise reuse mask. (c) The mask is estimated once and reused throughout denoising: verified tokens fetch historical KV states, with relative RoPE calibration applied to shifted keys, while unmatched tokens are recomputed before attention. The final memory-appending pass remains fully computed.

distribution is factorized as:

$$p_\theta\left(\mathbf{x}^{1:N} \mid \mathbf{c}^{1:N}\right) = \prod_{t=1}^{N} p_\theta\left(\mathbf{x}^t \mid \mathbf{x}^{<t}, \mathbf{c}^t\right), \tag{1}$$

where $\mathbf{c}^t$ may include user actions, camera controls, or prompt-based instructions that specify the motion and content of the t-th chunk. Each conditional term is instantiated by a few-step diffusion model conditioned on a historical KV cache constructed from preceding clean chunks. In practice, generating a chunk involves 2 to 4 denoising forward passes, followed by one additional forward pass that appends its fully denoised tokens to the KV cache (Yuan et al., 2026; Team et al., 2026b).

Existing acceleration methods either reduce the cost of historical context $\mathbf{x}^{<t}$ without exploiting cached representations as reusable computation, or perform coarse computation reuse that overlooks action-induced fine-grained correspondence. To this end, we propose an **action-guided fine-grained KV reuse framework** that exploits spatial-temporal redundancy between adjacent chunks. As shown in Figure 3, our method reuses computation only during iterative denoising and retains full computation in the final memory-appending pass. Specifically, we first translate action signals into a motion prior that constrains matching to a local candidate window. We then use the clean prediction from the first denoising step as a self-speculative probe to distinguish reusable regions from those requiring recomputation. Finally, we calibrate the 3D-RoPE of shifted cached keys through an incremental relative rotation, enabling position-consistent reuse without modifying optimized attention kernels.

3.1.1 Control-Guided Coarse Routing

Consecutive chunks in an interactive world model are related by explicit controls, such as user actions or camera movements. These controls provide a coarse yet informative prior on inter-chunk correspondence. Instead of globally matching each current token against all historical locations, we convert the control condition $\mathbf{c}^t$ for the t-th chunk into a spatial displacement field $\boldsymbol{\delta}_t : \mathcal{P} \to \mathbb{R}^2$, where $\mathcal{P} = [1, H] \times [1, W]$ denotes the spatial token grid. For a token at position $\mathbf{p} \in \mathcal{P}$, its likely correspondence in the preceding chunk is given by $\mathbf{p} - \boldsymbol{\delta}_t(\mathbf{p})$. Unlike a global translation, the displacement is position-dependent: for example, camera rotation induces different shifts at different spatial locations. This control-derived displacement field therefore serves as a coarse correspondence prior, reducing matching from the entire spatial grid to a local neighborhood.

3.1.2 LAYER-WISE ADAPTIVE SEARCH WINDOW

The control-derived displacement field is shared across layers, but the sensitivity to correspondence errors differs with representation depth. As shown in Figure 4, a narrow search window can miss valid correspondences in shallow layers, leading to larger deviations from full-computation outputs. In deeper layers, an overly broad window may instead introduce spatially incorrect yet semantically similar candidates. We therefore adopt a coarse-to-fine search allocation, using layer-dependent radii $r_1 \geq r_2 \geq \cdots \geq r_L$ for L Transformer layers. In practice, we partition the Transformer layers into shallow, middle, and deep groups, and select the radius of each group on held-out validation trajectories.

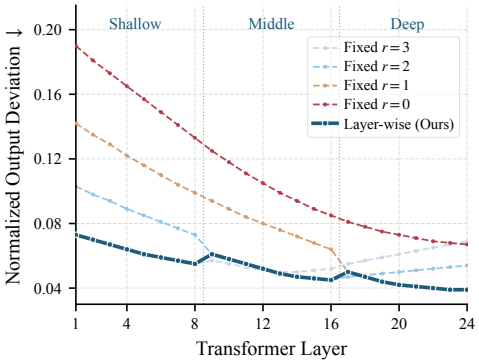


Figure 4: **Layer-wise sensitivity to the search radius.** Normalized output deviation from full computation under different radii r (lower is better). Shallow layers favor larger windows, while deeper layers favor smaller ones, motivating our coarse-to-fine schedule. Results are obtained with MatrixGame2 (He et al., 2025) on the WBench (Ying et al., 2026) dataset.

Moreover, the control signal provides directional visibility cues in addition to the correspondence center. For example, under a rightward camera motion, regions newly revealed at the right boundary have no valid historical correspondence, whereas tokens in the retained left region should preferentially search toward the right in the preceding chunk. We incorporate this directional prior by biasing candidate sampling within the local window toward the control-induced displacement, while leaving newly exposed or occluded regions to be rejected by the subsequent verification stage.

Specifically, for a target position $\mathbf{p}$ at layer l, we define its candidate set in the historical cache as

$$\Omega_l(\mathbf{p}) = \left\{ \mathbf{q} \in \mathcal{P} \;\middle|\; \left\| \mathbf{q} - \left(\mathbf{p} - \boldsymbol{\delta}_t(\mathbf{p}) \right) \right\|_\infty \leq r_l \right\}. \tag{2}$$

Here, $\mathbf{q}$ denotes a candidate historical location. Larger windows in shallow layers improve tolerance to local geometric misalignment, whereas smaller windows in deeper layers suppress ambiguous matches. The resulting candidates are subsequently verified by fine-grained feature matching.

3.1.3 SELF-SPECULATIVE FINE-GRAINED VERIFICATION

The control-guided candidate set $\Omega_l(\mathbf{p})$ provides only a coarse correspondence prior. It cannot fully account for local deviations caused by object motion, occlusion, or newly revealed content. We therefore verify the candidates using clean-latent similarity before reusing their KV states.

Inspired by speculative decoding in large language models (Leviathan et al., 2023; Li et al., 2024; Chen et al., 2023), we use an inexpensive preliminary prediction to propose reuse decisions and verify them using reliable historical content. Our mechanism is *self-speculative* because the probe is produced by the world model itself, without an auxiliary draft model. Specifically, the clean prediction $\hat{\mathbf{x}}_t^0$ obtained at the first denoising step is already structurally informative. We compare it with the fully denoised latent $\mathbf{x}_{t-1}^0$ of the immediately preceding chunk, which provides the temporally closest reference. For each target position $\mathbf{p}$, we select the most similar historical location within the control-guided window and accept it only if their similarity exceeds a threshold:

$$\begin{aligned} \mathbf{q}_l^*(\mathbf{p}) &= \arg\max_{\mathbf{q} \in \Omega_l(\mathbf{p})} \cos\left(\hat{\mathbf{x}}_t^0(\mathbf{p}), \mathbf{x}_{t-1}^0(\mathbf{q}) \right), \\ M_l(\mathbf{p}) &= \mathbb{I}\left[\cos\left(\hat{\mathbf{x}}_t^0(\mathbf{p}), \mathbf{x}_{t-1}^0\left(\mathbf{q}_l^*(\mathbf{p}) \right) \right) \geq \tau \right], \end{aligned} \tag{3}$$

where $\cos(\cdot,\cdot)$ denotes cosine similarity and $\mathbb{I}[\cdot]$ is the indicator function. Although the probe is shared across layers, the matched location $\mathbf{q}_l^*(\mathbf{p})$ and reuse mask $M_l(\mathbf{p})$ remain layer-specific because each layer uses a different candidate window $\Omega_l(\mathbf{p})$.

The masks are computed once from the first-step prediction and reused throughout the remaining denoising steps. We omit the denoising-step index below because the same masks and matching

locations are used at every subsequent step. Let $\mathcal{P}$ denote all token positions. For layer l, we define the reused and recomputed position sets as

$$\mathcal{A}_l = \{\mathbf{p} \in \mathcal{P} : M_l(\mathbf{p}) = 1\}, \qquad \mathcal{R}_l = \{\mathbf{p} \in \mathcal{P} : M_l(\mathbf{p}) = 0\}. \tag{4}$$

Positions in $\mathcal{A}_l$ reuse the cached representations at their matched historical locations, whereas positions in $\mathcal{R}_l$ are recomputed from the current chunk.

Let $\mathbf{h}_{l,t}(\mathbf{p})$ denote the input to the self-attention module at position $\mathbf{p}$ in layer l. Queries are required only at recomputed positions and are given by

$$\mathbf{Q}_{l,t}(\mathbf{p}) = \mathrm{Q}_l(\mathbf{h}_{l,t}(\mathbf{p})), \qquad \mathbf{p} \in \mathcal{R}_l. \tag{5}$$

No query projection is evaluated for positions in $\mathcal{A}_l$, since their self-attention outputs are directly reused.

We next construct a position-aligned hybrid key–value set over $\mathcal{P}$. Let $\bar{\mathbf{K}}_{l,t-1}$ and $\bar{\mathbf{V}}_{l,t-1}$ denote the keys and values stored by the final memory-appending pass of chunk $t-1$. For each current position, the hybrid set contains either the cached KV states at its verified historical match or newly computed KV states:

$$(\mathbf{K}_{l,t}(\mathbf{p}), \mathbf{V}_{l,t}(\mathbf{p})) = \begin{cases} \left(\mathcal{C}_l\left[\bar{\mathbf{K}}_{l,t-1}(\mathbf{q}_l^*(\mathbf{p})); \mathbf{q}_l^*(\mathbf{p}) \to \mathbf{p}\right], \bar{\mathbf{V}}_{l,t-1}(\mathbf{q}_l^*(\mathbf{p}))\right), & \mathbf{p} \in \mathcal{A}_l, \\ \mathrm{KV}_l(\mathbf{h}_{l,t}(\mathbf{p})), & \mathbf{p} \in \mathcal{R}_l. \end{cases} \tag{6}$$

Here, $\mathcal{C}_l[\cdot\,; \mathbf{q} \to \mathbf{p}]$ calibrates a cached key originally encoded at position $\mathbf{q}$ to the target position $\mathbf{p}$. This calibration is detailed in the next section.

For each recomputed position, attention is evaluated against this full hybrid key–value set. At a verified position, the self-attention branch output is instead copied from its matched location in the preceding chunk:

$$\mathbf{O}_{l,t}(\mathbf{p}) = \begin{cases} \mathrm{Attn}_l(\mathbf{Q}_{l,t}(\mathbf{p}), \mathbf{K}_{l,t}(\mathcal{P}), \mathbf{V}_{l,t}(\mathcal{P})), & \mathbf{p} \in \mathcal{R}_l, \\ \bar{\mathbf{O}}_{l,t-1}(\mathbf{q}_l^*(\mathbf{p})), & \mathbf{p} \in \mathcal{A}_l, \end{cases} \tag{7}$$

where $\bar{\mathbf{O}}_{l,t-1}$ denotes the self-attention output cached from the final memory-appending pass of chunk $t-1$. Here, Attn_l includes the attention operation and its output projection, and $\mathbf{O}_{l,t}$ denotes the resulting output. Thus, verified positions skip the current query projection, key–value projection, and attention computation, while their calibrated keys and cached values remain available to all recomputed queries. Camera injection, cross-attention, and the FFN remain dense and are evaluated at every position. Positions that fail verification, including occluded or newly exposed regions, follow the current computation path.

3.2 Relative RoPE Calibration for Shifted KV Reuse

For each verified position $\mathbf{p}$, we reuse the KV pair at its matched historical location $\mathbf{q}_l^*(\mathbf{p})$. Directly copying the cached key, however, causes positional misalignment: 3D RoPE encodes the original spatial-temporal coordinate of $\mathbf{q}_l^*(\mathbf{p})$, whereas the reused key is required at $\mathbf{p}$.

We correct the key with a relative RoPE rotation. Let $\mathbf{R}(\cdot)$ denote the 3D RoPE rotation and let $\Delta_l(\mathbf{p})$ be the relative spatial-temporal displacement from $\mathbf{q}_l^*(\mathbf{p})$ to $\mathbf{p}$. The calibrated key is

$$\widetilde{\mathbf{K}}_{t,n}^{l}(\mathbf{p}) = \mathbf{R}\big(\Delta_l(\mathbf{p})\big)\, \mathbf{K}_{t-1}^{l}\big(\mathbf{q}_l^*(\mathbf{p})\big). \tag{8}$$

This follows from the compositionality of RoPE rotations: the relative rotation directly migrates a cached key to its target coordinate, without explicitly undoing and reapplying RoPE. The value is copied unchanged because RoPE is applied only to keys and queries. Thus, reused KV states remain position-consistent with the current attention computation while preserving standard attention kernels.

4 Experiment

4.1 Experimental Setup

Models. We evaluate ActCache on three state-of-the-art interactive video world models: LingBot-World-Fast (Team et al., 2026b), MatrixGame2 (He et al., 2025), and Inspatio-World (Team, 2026).

Table 1: **Main results on WBench.** We report throughput in FPS, with speedup over native inference shown in parentheses, and WBench scores for video quality, setting adherence, interaction adherence, consistency, and physical compliance. Avg. is the average WBench score. Higher is better. Within each backbone, the best and second-best results are highlighted in **bold** and underline, respectively.

Model	Params	Throughput (FPS)↑	WBench					
			Quality ↑	Setting ↑	Interaction ↑	Consistency ↑	Physical ↑	Avg. ↑
HY-World 1.5	8B	2.0	78.1	72.2	87.5	86.9	66.3	78.2
SANA-WM	2.6B	25.1	79.3	76.1	82.1	80.7	61.9	76.0
DreamX-World	5B	4.6	77.5	80.8	78.4	74.9	63.3	75.0
LTX 2.3	22B	1.4	77.1	85.2	67.6	77.2	64.9	74.4
YUME 1.5	5B	4.3	77.6	72.4	72.0	80.1	65.2	73.5
LingBot-World-Fast	14B	1.7 (1.00×)	79.4	77.9	79.4	84.9	65.7	77.5
+ TeaCache	–	1.9 (1.12×)	79.1	75.3	76.1	84.5	64.3	75.9
+ Dummy Forcing	–	2.2 (1.30×)	78.4	76.2	**78.3**	83.2	64.7	76.2
+ WorldKV	–	2.1 (1.24×)	78.1	75.4	77.2	84.3	63.7	75.7
+ X-Cache	–	2.1 (1.24×)	77.2	75.7	77.4	83.8	63.8	75.6
+ ActCache (Ours)	–	**2.4 (1.41×)**	**79.2**	**76.5**	77.9	**84.5**	**64.9**	**76.6**
MatrixGame2	1.3B	13.5 (1.00×)	73.8	67.1	80.6	65.1	57.2	68.8
+ TeaCache	–	16.1 (1.19×)	71.2	66.4	77.3	64.2	56.3	67.1
+ Dummy Forcing	–	18.0 (1.33×)	72.4	**66.9**	79.7	64.1	56.2	67.9
+ WorldKV	–	18.1 (1.34×)	72.5	65.6	78.9	64.6	56.7	67.7
+ X-Cache	–	17.7 (1.31×)	71.9	65.7	78.3	63.4	55.4	67.0
+ ActCache (Ours)	–	**19.3 (1.43×)**	**73.2**	66.6	**80.3**	**64.8**	**57.7**	**68.5**
Inspatio-World	1.3B	16.4 (1.00×)	71.5	71.4	72.8	88.4	65.2	73.9
+ TeaCache	–	20.2 (1.23×)	68.9	69.4	71.3	86.4	62.7	71.7
+ Dummy Forcing	–	21.5 (1.31×)	69.2	69.4	71.5	87.1	63.4	72.1
+ WorldKV	–	21.1 (1.29×)	70.1	70.4	70.3	86.8	64.1	72.3
+ X-Cache	–	20.8 (1.27×)	67.4	69.2	71.0	80.5	64.2	70.5
+ ActCache (Ours)	–	**22.8 (1.39×)**	**70.5**	**70.9**	**71.5**	**87.6**	**64.7**	**73.0**

LingBot-World-Fast is a 14B model for open-world exploration, while MatrixGame2 and Inspatio-World are 1.3B models designed for real-time gameplay and trajectory-driven novel-view synthesis, respectively. All models perform few-step, autoregressive video generation conditioned on historical KV caches and external controls, such as camera poses or keyboard actions. Their diverse scales, applications, and control interfaces provide a comprehensive testbed for evaluating the generality of ActCache.

Evaluation Benchmark. We evaluate ActCache on WBench (Ying et al., 2026), a multi-turn benchmark comprising 289 scenarios and 1,058 interaction turns with motion prompts, 6-DoF camera poses, and keyboard actions. During evaluation, WBench maps each control sequence to the native interface of each backbone. ActCache derives its displacement prior from the same controls without modifying the original conditioning mechanism. We report video quality, setting adherence, interaction adherence, temporal consistency, and physical compliance, together with end-to-end throughput and speedup.

Implementation Detail. All experiments are conducted in PyTorch on one NVIDIA H20 GPU using each backbone's native sampling and KV-cache configuration. ActCache is applied only during iterative denoising, while the final memory-appending pass is fully computed. We divide Transformer layers into shallow, middle, and deep groups and use the radius schedule $[3, 2, 1]$, with $\tau = 0.85$ for verification. Throughput is measured end-to-end, including routing, matching, RoPE calibration, and cache operations. Further implementation details are provided in A.1.

4.2 Main Results

Interactive Evaluation on WBench. Table 1 compares ActCache with native inference and training-free methods across three interactive world-model backbones. ActCache achieves the highest throughput on all models, reaching $1.41\times$, $1.43\times$, and $1.39\times$ speedup on LingBot-World-Fast, MatrixGame2, and Inspatio-World, respectively. Among acceleration methods, ActCache achieves the highest WBench score average across all three backbones, with only 0.3–0.9 point degradation relative to native inference. All models are evaluated on 5-second videos at 720p resolution.

Table 2: **World-memory consistency under revisit trajectories.** We compare revisited frames with their corresponding first-visit frames. WorldKV results are reproduced from (Yi et al., 2026).

Backbone	Method	PSNR↑	SSIM↑	FID↓
LingBot-World-Fast	Native (Full KV)	**15.901**	0.472	85.705
	+ TeaCache	14.174	0.432	98.756
	+ WorldKV	15.660	0.463	85.644
	+ ActCache (Ours)	15.784	**0.475**	**85.281**
MatrixGame2	Native (Sliding Window)	11.422	0.280	157.261
	+ TeaCache	9.273	0.250	190.836
	+ WorldKV	14.101	**0.405**	93.561
	+ ActCache (Ours)	**14.610**	0.384	**91.493**

Table 3: **Compatibility with complementary methods.** We report throughput in FPS and WBench Avg. ActCache complements both denoising-step caching and historical-context reduction.

Method	LingBot		MatrixGame2	
	FPS (×)↑	Avg.↑	FPS (×)↑	Avg.↑
Native	1.7 (1.00)	77.5	13.5 (1.00)	68.8
TeaCache	1.9 (1.12)	75.9	16.1 (1.19)	67.1
WorldKV	2.1 (1.24)	75.7	18.1 (1.34)	67.7
ActCache	2.4 (1.41)	**76.6**	19.3 (1.43)	**68.5**
TeaCache + ActCache	2.6 (1.53)	76.3	21.5 (1.59)	67.3
WorldKV + ActCache	**2.9 (1.71)**	76.1	**23.6 (1.75)**	68.1

ActCache is effective in the few-step setting, where step-wise methods such as TeaCache have limited reuse opportunities. By exploiting spatial redundancy across action-conditioned chunks, ActCache outperforms X-Cache in both WBench performance and throughput. Control-guided routing and self-speculative verification further ensure reliable reuse, yielding a favorable quality–efficiency trade-off without retraining or modifying the base models. Additional qualitative results are provided in A.3.

World-Memory Consistency. Following WorldKV (Yi et al., 2026), we evaluate revisit consistency by comparing frames before and after leaving and returning to the same viewpoint. As shown in Table 2, ActCache maintains competitive memory consistency on both backbones. It achieves the best structural and distributional consistency on LingBot-World-Fast, and the best PSNR and FID on MatrixGame2.

We attribute this consistency to ActCache's state-preserving reuse mechanism. It explicitly reuses verified KV states for reliable correspondences and recomputes only changed regions, reducing redundant computation while retaining reliable historical states. Full computation during memory appending further prevents reuse errors from accumulating in the autoregressive cache.

4.3 Efficiency Analysis

Compatibility with orthogonal Methods. ActCache targets cross-chunk KV reuse and is complementary to methods that operate on other sources of redundancy. TeaCache reuses features across denoising steps, whereas WorldKV manages long-horizon historical context. In combined inference, TeaCache is applied after the first denoising step to preserve the self-speculative probe. For ActCache and WorldKV, the immediately preceding clean chunk is retained for local reuse, while WorldKV manages the remaining history.

Table 3 validates this compatibility. Combining ActCache with TeaCache improves throughput to 2.6 FPS ($1.53\times$) on LingBot-World-Fast and 21.5 FPS ($1.59\times$) on MatrixGame2, exceeding either method alone while retaining competitive WBench performance. Combining ActCache with WorldKV yields the highest throughput, reaching 2.9 FPS ($1.71\times$) and 23.6 FPS ($1.75\times$), respectively.

4.4 Ablation Study

Unless stated otherwise, all ablations use MatrixGame2 with the same WBench control sequences and inference configuration. We report throughput, KV reuse rate, WBench Consistency, and WBench Avg.

Fine-Grained Reuse Strategy. Table 4 validates the cross-chunk reuse design. Direct KV copying achieves high reuse but severely degrades consistency. Control-guided routing narrows the mismatch, self-speculative verification rejects unreliable matches, and relative RoPE calibration restores positional alignment. Their combination achieves the best quality–efficiency trade-off.

Table 4: **Ablation of fine-grained reuse.** CG denotes control-guided routing, SV denotes self-speculative verification, and RC denotes relative RoPE calibration. Results are evaluated on MatrixGame2.

Method	CG	SV	RC	Reuse Rate (%)↑	FPS ↑	Consist. ↑	Avg. ↑
Full computation	✗	✗	✗	0.0	13.5	65.1	68.8
Direct reuse	✗	✗	✗	100.0	**23.6**	35.8	31.5
Local reuse	✓	✗	✓	71.4	21.2	50.9	55.1
Verified reuse w/o RC	✓	✓	✗	43.5	19.5	62.9	66.8
ActCache	✓	✓	✓	43.5	19.3	**64.8**	**68.5**

Table 5: **Impact of control-prior modeling.** A spatially varying displacement field and directional sampling improve correspondence reliability while reducing the candidate space. Results are evaluated on MatrixGame2.

Prior	Candidate Ratio (%)↓	FPS ↑	Consist. ↑	Avg. ↑
Global search	100.0	16.7	63.7	67.4
Global shift	16.2	18.4	61.8	65.9
Displacement field	13.6	18.9	64.1	68.0
Directional bias	**10.8**	**19.3**	**64.8**	**68.5**

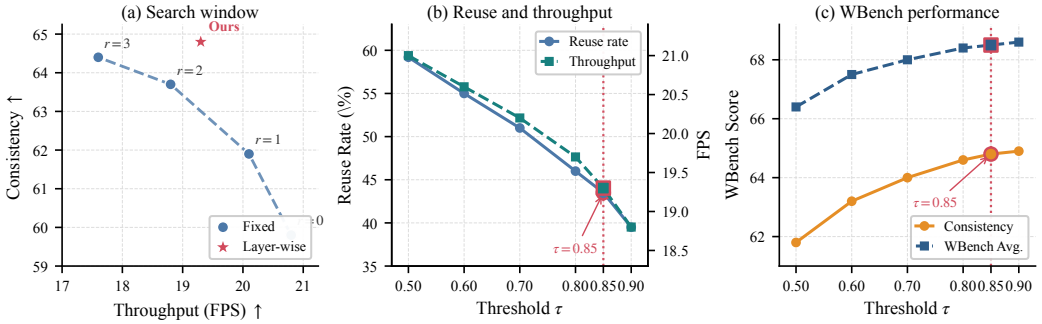


Figure 5: **Sensitivity to search-window and verification designs.** (a) Layer-wise windows achieve a better consistency–throughput trade-off than fixed-radius windows. (b) A higher verification threshold reduces reuse and throughput. (c) Lower verification thresholds admit more mismatched correspondences, while $\tau = 0.85$ provides a favorable quality–efficiency trade-off.

Impact of Control Prior Modeling. Table 5 compares correspondence priors. Global search is expensive, while a global shift cannot capture spatially varying motion under camera rotation. A position-dependent displacement field improves matching, and directional bias further removes unlikely candidates near newly exposed boundaries.

Effect of the Layer-wise Search Window. Figure 5 (a) compares fixed and layer-wise search windows. Small fixed windows miss valid correspondences, whereas large windows increase matching cost. Our layer-wise schedule achieves the best consistency–throughput trade-off.

Impact of the Verification Threshold. Figure 5 (b)(c) show the effect of the acceptance threshold τ. Lower thresholds increase reuse and throughput but admit mismatched states. Performance saturates near $\tau = 0.85$. Higher thresholds reduce reuse with little additional gain. We therefore use $\tau = 0.85$ throughout.

5 Conclusion

In this work, we introduce **ActCache**, a framework for accelerating autoregressive interactive world models through action-guided, fine-grained KV reuse across consecutive video chunks. ActCache converts control signals into spatial correspondence priors, uses self-speculative probing to verify reusable regions, and applies relative 3D-RoPE calibration to preserve position-consistent attention for shifted keys. Extensive experiments across diverse world-model backbones show that ActCache improves generation throughput while maintaining interactive quality and world-memory consistency. Its compatibility with complementary methods further demonstrates that fine-grained current-chunk reuse is a practical path toward real-time interactive world generation.

AI USE STATEMENT

In this work, we used generative AI tools for language editing and limited assistance with code implementation and debugging. We did not use generative AI tools to formulate research ideas or hypotheses, conduct the literature review, generate data, or evaluate and interpret experimental results. Other disclosure categories are not applicable to this work. All AI-assisted text was carefully reviewed and revised by the authors. AI-assisted code was inspected, tested, and validated against the intended algorithms and experimental settings by the authors. The authors take full responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, personally identifiable information, or private data. We are not aware of any specific ethical concerns associated with this work. The authors have followed standard practices for research integrity and reproducibility throughout the study.

REPRODUCIBILITY STATEMENT

We have taken several steps to facilitate the reproducibility of this work. The proposed method is described in detail in the main paper, with pseudocode and additional implementation details provided in the appendix. Additional results and ablation studies are included to support the robustness of our findings.

REFERENCES

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

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.

Hanshuai Cui, Zhiqing Tang, Zhi Yao, Fanshuai Meng, Weijia Jia, and Wei Zhao. Not all frames deserve full computation: Accelerating autoregressive video generation via selective computation and predictive extrapolation. *CoRR*, abs/2604.02979, 2026. doi: 10.48550/ARXIV.2604.02979. URL https://doi.org/10.48550/arXiv.2604.02979.

Zicheng Duan, Jiatong Xia, Zeyu Zhang, Wenbo Zhang, Gengze Zhou, Chenhui Gou, Yefei He, Feng Chen, Xinyu Zhang, and Lingqiao Liu. Liveworld: Simulating out-of-sight dynamics in generative video world models. *arXiv preprint arXiv:2603.07145*, 2026.

Weilun Feng, Guoxin Fan, Haotong Qin, Chuanguang Yang, Mingqiang Wu, Yuqi Li, Xiangqi Li, Zhulin An, Libo Huang, Dingrui Wang, Longlong Liao, Michele Magno, and Yongjun Xu. Worldcache: Accelerating world models for free via heterogeneous token caching. *CoRR*, abs/2603.06331, 2026. doi: 10.48550/ARXIV.2603.06331. URL https://doi.org/10.48550/arXiv.2603.06331.

Hang Guo, Zhaoyang Jia, Jiahao Li, Bin Li, Yuanhao Cai, Jiangshan Wang, Yawei Li, and Yan Lu. Efficient autoregressive video diffusion with dummy head. *arXiv preprint arXiv:2601.20499*, 2026.

Junliang Guo, Yang Ye, Tianyu He, Haoyu Wu, Yushu Jiang, Tim Pearce, and Jiang Bian. Mineworld: a real-time and open-source interactive world model on minecraft. *arXiv preprint arXiv:2504.08388*, 2025.

Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, et al. Matrix-game 2.0: An open-source real-time and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025.

Yicong Hong, Yiqun Mei, Chongjian Ge, Yiran Xu, Yang Zhou, Sai Bi, Yannick Hold-Geoffroy, Mike Roberts, Matthew Fisher, Eli Shechtman, et al. Relic: Interactive video world model with long-horizon memory. *arXiv preprint arXiv:2512.04040*, 2025.

Junchao Huang, Ziyang Ye, Xinting Hu, Tianyu He, Guiyu Zhang, Shaoshuai Shi, Jiang Bian, and Li Jiang. Live: Long-horizon interactive video world modeling. *arXiv preprint arXiv:2602.03747*, 2026a.

Siqiao Huang, Jialong Wu, Qixing Zhou, Shangchen Miao, and Mingsheng Long. Vid2world: Crafting video diffusion models to interactive world models. *arXiv preprint arXiv:2505.14357*, 2025.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *Advances in Neural Information Processing Systems*, 38:167283–167308, 2026b.

Team HY-World, Chenjie Cao, Xuhui Zuo, Zhenwei Wang, Yisu Zhang, Junta Wu, Zhenyang Liu, Yuning Gong, Yang Liu, Bo Yuan, et al. Hy-world 2.0: A multi-modal world model for reconstructing, generating, and simulating 3d worlds. *arXiv preprint arXiv:2604.14268*, 2026.

Yicheng Ji, Zhizhou Zhong, Jun Zhang, Qinyu Yang, Xitai Jin, Yingsheng Qin, Wenhan Luo, Shuiyang Mao, Wei Liu, and Huan Li. Forcing-kv: Hybrid kv cache compression for efficient autoregressive video diffusion models. *ArXiv*, abs/2605.09681, 2026. URL `https://api.semanticscholar.org/CorpusID:288256630`.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pp. 19274–19286. PMLR, 2023.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*, 2024.

Yeqing Lin, Minji Lee, Zhao Zhang, and Mohammed AlQuraishi. Out of many, one: Designing and scaffolding proteins at the scale of the structural universe with genie 2. *arXiv preprint arXiv:2405.15489*, 2024.

Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7353–7363, 2025a.

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

Jinming Lou, Wenyang Luo, Yufan Liu, Bing Li, Xinmiao Ding, Weiming Hu, Yuming Li, and Chenguang Ma. Token caching for diffusion transformer acceleration. *arXiv preprint arXiv:2409.18523*, 2024.

Yuexiao Ma, Xuzhe Zheng, Jing Xu, Xiwei Xu, Feng Ling, Xiawu Zheng, Huafeng Kuang, Huixia Li, Xing Wang, Xuefeng Xiao, Fei Chao, and Rongrong Ji. Flow caching for autoregressive video generation. *CoRR*, abs/2602.10825, 2026. doi: 10.48550/ARXIV.2602.10825. URL `https://doi.org/10.48550/arXiv.2602.10825`.

Xiaofeng Mao, Zhen Li, Chuanhao Li, Xiaojie Xu, Kaining Ying, Tong He, Jiangmiao Pang, Yu Qiao, and Kaipeng Zhang. Yume-1.5: A text-controlled interactive world generation model. *arXiv preprint arXiv:2512.22096*, 2025a.

Xiaofeng Mao, Shaoheng Lin, Zhen Li, Chuanhao Li, Wenshuo Peng, Tong He, Jiangmiao Pang, Mingmin Chi, Yu Qiao, and Kaipeng Zhang. Yume: An interactive world generation model. *arXiv preprint arXiv:2507.17744*, 2025b.

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.

Yu Shang, Yuming Lin, Yu Zheng, Hangyu Fan, Jingtao Ding, Jie Feng, Jiansheng Chen, Li Tian, and Yong Li. Urbanworld: An urban world model for 3d city generation. *arXiv preprint arXiv:2407.11965*, 2024.

Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. Worldplay: Towards long-term geometric consistency for real-time interactive world modeling. *arXiv preprint arXiv:2512.14614*, 2025.

AlayaWorld Team, Kaipeng Zhang, Chuanhao Li, Yifan Zhan, Yongtao Ge, Yuanyang Yin, Jiaming Tan, Kang He, Liaoyuan Fan, Ruicong Liu, et al. Alayaworld: Long-horizon and playable video world generation. *arXiv preprint arXiv:2607.06291*, 2026a.

HunyuanWorld Team, Zhenwei Wang, Yuhao Liu, Junta Wu, Zixiao Gu, Haoyuan Wang, Xuhui Zuo, Tianyu Huang, Wenhuan Li, Sheng Zhang, et al. Hunyuanworld 1.0: Generating immersive, explorable, and interactive 3d worlds from words or pixels. *arXiv preprint arXiv:2507.21809*, 2025.

InSpatio Team. Inspatio-world: A real-time 4d world simulator via spatiotemporal autoregressive modeling, 2026.

Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, Yihang Chen, Jie Liu, Yansong Cheng, Yao Yao, Jiayi Zhu, Yihao Meng, Kecheng Zheng, Qingyan Bai, Jingye Chen, Zehong Shen, Yue Yu, Xing Zhu, Yujun Shen, and Hao Ouyang. Advancing open-source world models. *arXiv preprint arXiv:2601.20540*, 2026b.

Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, et al. Advancing open-source world models. *arXiv preprint arXiv:2601.20540*, 2026c.

Zile Wang, Zexiang Liu, Jiaxing Li, Kaichen Huang, Baixin Xu, Fei Kang, Mengyin An, Peiyu Wang, Biao Jiang, Yichen Wei, et al. Matrix-game 3.0: Real-time and streaming interactive world model with long-horizon memory. *arXiv preprint arXiv:2604.08995*, 2026.

Jung Yi, Minjae Kim, Paul Hyunbin Cho, Wooseok Jang, Sangdoo Yun, and Seungryong Kim. Worldkv: Efficient world memory with world retrieval and compression. *arXiv preprint arXiv:2605.22718*, 2026.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6613–6623, 2024.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22963–22974, 2025.

Kaining Ying, Hengrui Hu, Siyu Ren, Jiamu Li, Fengjiao Chen, Ziwen Wang, Xuezhi Cao, Xunliang Cai, and Henghui Ding. Wbench: A comprehensive multi-turn benchmark for interactive video world model evaluation. *arXiv preprint arXiv:2605.25874*, 2026.

Shenghai Yuan, Yuanyang Yin, Zongjian Li, Xinwei Huang, Xiao Yang, and Li Yuan. Helios: Real real-time long video generation model. *arXiv preprint arXiv:2603.04379*, 2026.

Yixiao Zeng, Jianlei Zheng, Chaoda Zheng, Shijia Chen, Mingdian Liu, Tongping Liu, Tengwei Luo, Yu Zhang, Boyang Wang, Linkun Xu, et al. X-cache: Cross-chunk block caching for few-step autoregressive world models inference. *arXiv preprint arXiv:2604.20289*, 2026.

Hui Zhang, Tingwei Gao, Jie Shao, and Zuxuan Wu. Blockdance: Reuse structurally similar spatio-temporal features to accelerate diffusion transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12891–12900, 2025a.

Yifan Zhang, Chunli Peng, Boyang Wang, Puyi Wang, Qingcheng Zhu, Fei Kang, Biao Jiang, Zedong Gao, Eric Li, Yang Liu, et al. Matrix-game: Interactive world foundation model. *arXiv preprint arXiv:2506.18701*, 2025b.

Min Zhao, Hongzhou Zhu, Bokai Yan, Zihan Zhou, Yimin Chen, Wenqiang Sun, Kaiwen Zheng, Guande He, Xiao Yang, Chongxuan Li, et al. minwm: A full-stack open-source framework for real-time interactive video world models. *arXiv preprint arXiv:2605.30263*, 2026.

Xuanlei Zhao, Xiaolong Jin, Kai Wang, and Yang You. Real-time video generation with pyramid attention broadcast. In *International Conference on Learning Representations*, volume 2025, pp. 3296–3319, 2025.

Chaoda Zheng, Sean Li, Jinhao Deng, Zhennan Wang, Shijia Chen, Liqiang Xiao, Ziheng Chi, Hongbin Lin, Kangjie Chen, Boyang Wang, et al. X-world: Controllable ego-centric multi-camera world models for scalable end-to-end driving. *arXiv preprint arXiv:2603.19979*, 2026.

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 Appendix

A.1 Implementation Details

A.1.1 Control-to-Displacement Mapping

Table 6: **Reliability of self-speculative probing on MatrixGame2.** The oracle mask is derived from the final clean latent $\mathbf{x}_t^0$ under full computation. Higher is better for all metrics except none. The final-step oracle is used only for offline analysis.

Probe	IoU (%)↑	Precision (%)↑	Recall (%)↑	Reuse (%)↑	FPS↑
First-step $\hat{\mathbf{x}}_t^{0,(1)}$ (Ours)	88.7	91.2	97.0	43.5	**19.3**
Second-step $\hat{\mathbf{x}}_t^{0,(2)}$	91.1	94.3	96.4	41.8	17.1
Final clean latent $\mathbf{x}_t^0$ (Oracle)	100.0	100.0	100.0	40.9	–

For action-controlled world models, we map each discrete input to its canonical image-plane direction to guide local search. Specifically, `W`/`S` indicate forward/backward navigation, `A`/`D` indicate left/right translation, and `I`/`K` and `J`/`L` indicate up/down and left/right camera panning, respectively. These directions determine the center and directional bias of the search window in the preceding chunk. The mapping provides only a coarse routing prior. It does not model scene depth, parallax, or object motion, which are subsequently handled by self-speculative verification.

A.1.2 Algorithm Details

Algorithm 1 summarizes ActCache inference for one autoregressive chunk. It first derives control-guided local windows and fully computes the initial denoising step to obtain the self-speculative probe. The probe determines token-wise correspondences and reuse masks, which are shared by the remaining denoising steps for selective KV reuse. The final memory-appending pass is fully computed to maintain a clean historical cache.

A.2 More Quantitative Results

We provide additional analyses of probe reliability, video quality, and human preference to complement the main results.

A.2.1 Reliability of Self-Speculative Probing

We examine whether the clean estimate predicted at the first denoising step contains sufficient structure to determine KV reuse. Let $\hat{\mathbf{x}}_t^{0,(s)}$ denote the $\mathbf{x}_0$ prediction obtained at the s-th denoising step. For offline analysis, we construct an oracle reuse mask M_l^{ora} by applying the same local cosine-matching and thresholding procedure to the final clean latent $\mathbf{x}_t^0$ obtained from full computation. The oracle uses the same candidate window $\Omega_l(\mathbf{p})$ and threshold τ as the first-step probe. We then compare the mask $M_l^{(s)}$ estimated from $\hat{\mathbf{x}}_t^{0,(s)}$ with M_l^{ora}.

We report mask IoU, reuse precision, and reuse recall, aggregated over all layers and spatial positions. Specifically, IoU measures the overlap between the predicted and oracle reuse regions; precision measures the fraction of predicted reusable positions that are accepted by the oracle, and recall measures the fraction of oracle-reusable positions recovered by the probe. High precision is particularly important because it prevents unreliable KV reuse. As shown in Table 6, the first-step $\mathbf{x}_0$ prediction already yields strong agreement with the oracle, while probing at later steps provides only modest gains at a lower throughput. This validates the first-step prediction as an efficient one-time probe for reuse-mask estimation.

A.2.2 Additional Video-Quality Metrics

We further evaluate generated rollouts using the prompt-free custom evaluation of VBench. This protocol measures subject consistency, background consistency, motion smoothness, dynamic de-

Algorithm 1 ActCache Inference for Chunk t

Input: Noisy latent $\mathbf{x}_t^{(S)}$, control $\mathbf{c}^t$, previous clean latent $\mathbf{x}_{t-1}^0$, and historical KV cache $\mathcal{C}_{t-1}$
Parameters: Layer-wise search radii $\{r_l\}_{l=1}^L$ and verification threshold τ
Output: Clean chunk $\mathbf{x}_t^0$ and updated historical cache $\mathcal{C}_t$

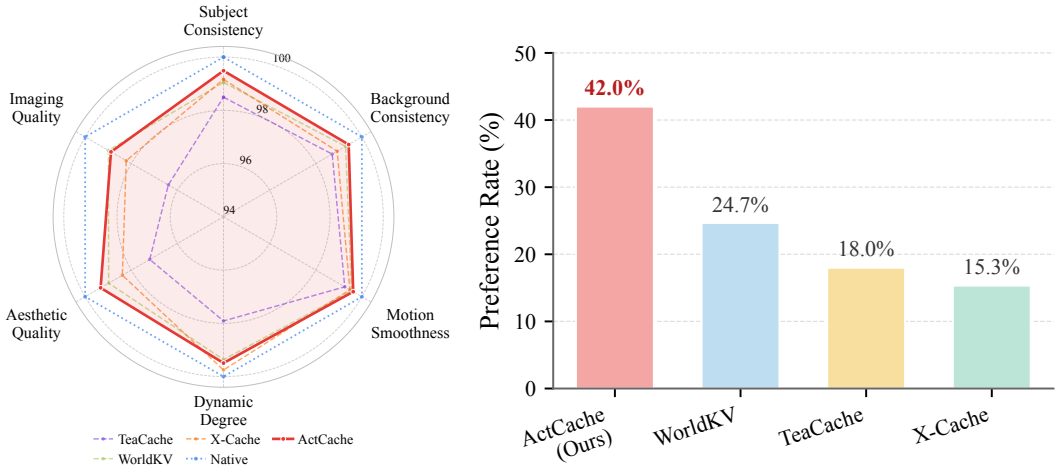
$\boldsymbol{\delta}_t \leftarrow \textsc{ControlToDisplacement}(\mathbf{c}^t)$
for $l = 1, \dots, L$ **do**
  $\boldsymbol{\delta}_{t,l} \leftarrow \textsc{AlignToLayer}(\boldsymbol{\delta}_t, l)$
  for each target position $\mathbf{p} \in \mathcal{P}_l$ **do**
    $\Omega_l(\mathbf{p}) \leftarrow \textsc{LocalWindow}(\mathbf{p}, \boldsymbol{\delta}_{t,l}(\mathbf{p}), r_l)$
  end for
end for
$(\mathbf{x}_t^{(S-1)}, \hat{\mathbf{x}}_t^{0,(1)}) \leftarrow \textsc{FullDenoiseStep}(\mathbf{x}_t^{(S)}, \mathcal{C}_{t-1}, \mathbf{c}^t, S)$ {Fully compute the first denoising step}
for $l = 1, \dots, L$ **do**
  $\hat{\mathbf{Z}}_{t,l} \leftarrow \textsc{AlignToLayer}(\hat{\mathbf{x}}_t^{0,(1)}, l)$
  $\mathbf{Z}_{t-1,l} \leftarrow \textsc{AlignToLayer}(\mathbf{x}_{t-1}^0, l)$
  for each target position $\mathbf{p} \in \mathcal{P}_l$ **do**
    $\mathbf{q}_l^*(\mathbf{p}) \leftarrow \arg\max_{\mathbf{q} \in \Omega_l(\mathbf{p})} \cos\big(\hat{\mathbf{Z}}_{t,l}(\mathbf{p}), \mathbf{Z}_{t-1,l}(\mathbf{q})\big)$
    $M_l(\mathbf{p}) \leftarrow \mathbb{I}\Big[\cos\big(\hat{\mathbf{Z}}_{t,l}(\mathbf{p}), \mathbf{Z}_{t-1,l}(\mathbf{q}_l^*(\mathbf{p}))\big) \geq \tau\Big]$
  end for
  $\mathcal{R}_l \leftarrow \{\mathbf{p} \in \mathcal{P}_l \mid M_l(\mathbf{p}) = 1\}$; $\bar{\mathcal{R}}_l \leftarrow \mathcal{P}_l \setminus \mathcal{R}_l$
end for
for $s = S-1, \dots, 1$ **do**
  $\mathbf{H}_{t,0}^{(s)} \leftarrow \textsc{InputEmbed}(\mathbf{x}_t^{(s)}, \mathbf{c}^t, s)$
  for $l = 1, \dots, L$ **do**
    $\mathbf{Q}_{t,l}^{(s)} \leftarrow \textsc{ProjectQ}_l(\mathbf{H}_{t,l-1}^{(s)})$ {Compute queries at all current positions}
    $(\mathbf{K}_{t,l}^{\mathrm{new},(s)}, \mathbf{V}_{t,l}^{\mathrm{new},(s)}) \leftarrow \textsc{ProjectKV}_l(\mathbf{H}_{t,l-1}^{(s)}[\bar{\mathcal{R}}_l])$ {Compute fresh KV only for rejected positions}
    $(\mathbf{K}_l^{\mathrm{src}}, \mathbf{V}_l^{\mathrm{src}}) \leftarrow \textsc{Gather}\big(\mathcal{C}_{t-1,l}, \mathbf{q}_l^*(\mathcal{R}_l)\big)$
    $\mathbf{K}_{t,l}^{\mathrm{reuse},(s)} \leftarrow \textsc{RelRoPE}\big(\mathbf{K}_l^{\mathrm{src}}, \mathbf{q}_l^*(\mathcal{R}_l), \mathcal{R}_l\big)$ {Calibrate keys from source to target coordinates}
    $\mathbf{V}_{t,l}^{\mathrm{reuse},(s)} \leftarrow \mathbf{V}_l^{\mathrm{src}}$
    $(\mathbf{K}_{t,l}^{(s)}, \mathbf{V}_{t,l}^{(s)}) \leftarrow \textsc{ScatterKV}\left(\begin{array}{ll} (\mathbf{K}_{t,l}^{\mathrm{reuse},(s)}, \mathbf{V}_{t,l}^{\mathrm{reuse},(s)}) & \text{at } \mathcal{R}_l, \\ (\mathbf{K}_{t,l}^{\mathrm{new},(s)}, \mathbf{V}_{t,l}^{\mathrm{new},(s)}) & \text{at } \bar{\mathcal{R}}_l \end{array}\right)$
    $\mathbf{H}_{t,l}^{(s)} \leftarrow \textsc{TransformerLayer}_l\big(\mathbf{H}_{t,l-1}^{(s)}, \mathbf{Q}_{t,l}^{(s)}, \mathbf{K}_{t,l}^{(s)}, \mathbf{V}_{t,l}^{(s)}, \mathcal{C}_{t-1,l}\big)$
  end for
  $\hat{\mathbf{x}}_t^{0,(S-s+1)} \leftarrow \textsc{PredictClean}(\mathbf{H}_{t,L}^{(s)})$
  $\mathbf{x}_t^{(s-1)} \leftarrow \textsc{SchedulerStep}\big(\mathbf{x}_t^{(s)}, \hat{\mathbf{x}}_t^{0,(S-s+1)}, s\big)$
end for
$\mathbf{x}_t^0 \leftarrow \mathbf{x}_t^{(0)}$
$\mathcal{C}_t \leftarrow \textsc{FullAppend}(\mathbf{x}_t^0, \mathcal{C}_{t-1}, \mathbf{c}^t)$ {Fully recompute and append clean KV states}
return $\mathbf{x}_t^0, \mathcal{C}_t$

gree, aesthetic quality, and imaging quality without requiring the standard VBench prompt suite. All methods are evaluated using identical initial frames and control sequences.

Figure 6a and Table 7 show that ActCache preserves video quality across the evaluated dimensions. Among accelerated methods, it achieves the best subject consistency, background consistency, motion smoothness, and aesthetic quality, while ranking second in imaging quality. Its temporal-consistency metrics remain close to native inference, indicating that fine-grained KV reuse introduces little temporal instability. WorldKV attains the highest imaging quality, whereas X-Cache

Table 7: **VBench evaluation on MatrixGame2.** We use the custom-input evaluation of VBench on videos generated from identical initial frames and control sequences. Higher is better for all metrics. Among accelerated methods, the best and second-best results are highlighted in **bold** and underlined, respectively.

Method	Subject↑	Background↑	Smoothness↑	Dynamic↑	Aesthetic↑	Imaging↑
Native	95.62	96.34	98.35	68.89	65.72	69.61
TeaCache	94.18	95.10	97.62	67.45	63.88	67.10
WorldKV	94.71	<u>95.66</u>	<u>97.91</u>	68.45	<u>65.05</u>	**68.91**
X-Cache	<u>94.82</u>	95.31	97.84	**68.72**	64.66	68.37
ActCache	**95.13**	**95.79**	**97.98**	68.54	**65.28**	<u>68.83</u>



(a) (b)

Figure 6: **VBench evaluation and user preference study on MatrixGame2.** (a) VBench evaluation reports quality retention normalized by native inference. ActCache achieves the best results on four of the six VBench metrics and provides the most balanced quality preservation overall. Raw scores are provided in Table 7. (b) User preference study: fifteen participants select the preferred video among outputs from competing accelerators under identical initial frames, prompts, and control trajectories. Each participant evaluates 20 randomized trajectories.

obtains the highest dynamic degree. Overall, ActCache provides the most balanced visual-quality preservation under acceleration.

A.2.3 User Study

We conduct a blind user study to evaluate perceptual quality under interactive video generation. Fifteen volunteers evaluate 20 randomized trajectories each, resulting in 300 preference annotations. For each trajectory, participants are presented with videos generated by different acceleration methods from the same initial frame, prompt, and control sequence. Video order is randomized and method identities are hidden. Participants select the video that best preserves overall visual quality, temporal consistency, and responsiveness to the given controls. As shown in Fig. 6b, ActCache receives the highest preference rate among the compared accelerators, corroborating the automatic evaluations.

A.3 More Qualitative Results

This section we show more qualitative results between several methods.

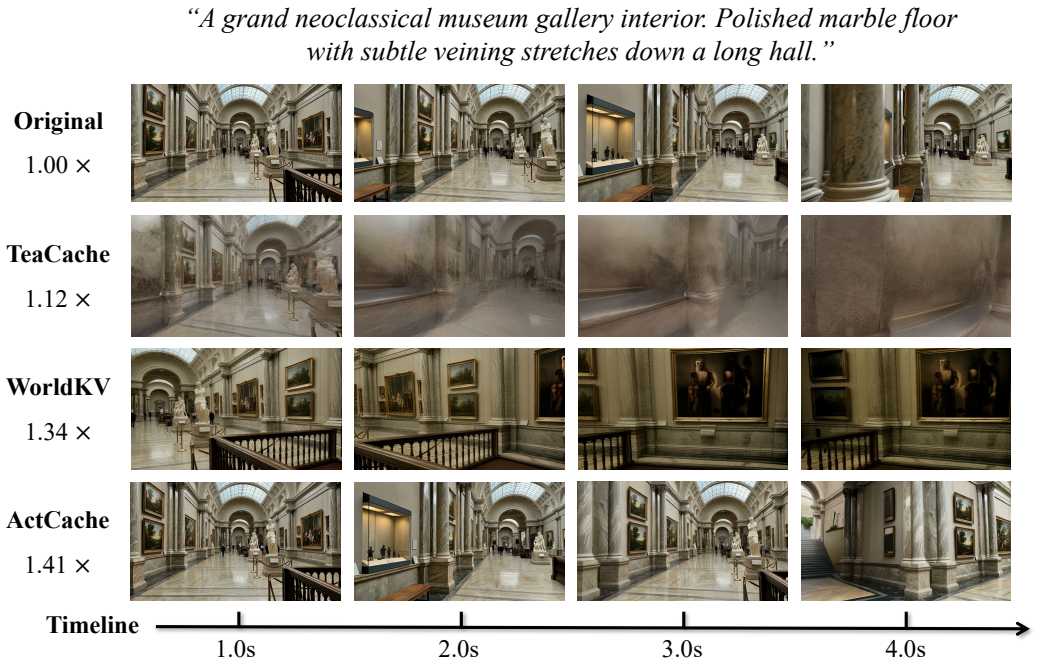


Figure 7: Qualitative examples for ActCache and other methods on Lingbot-World.

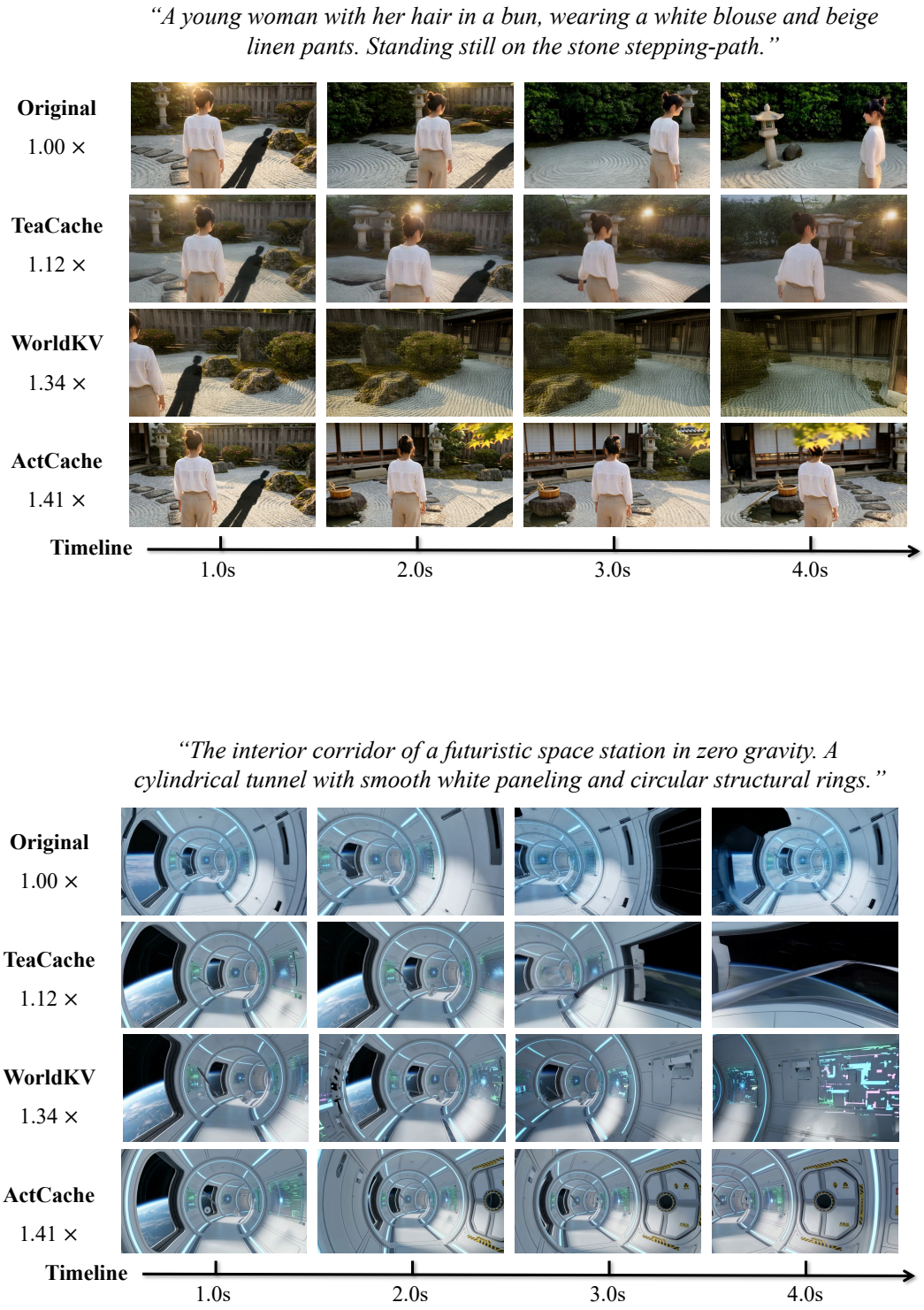
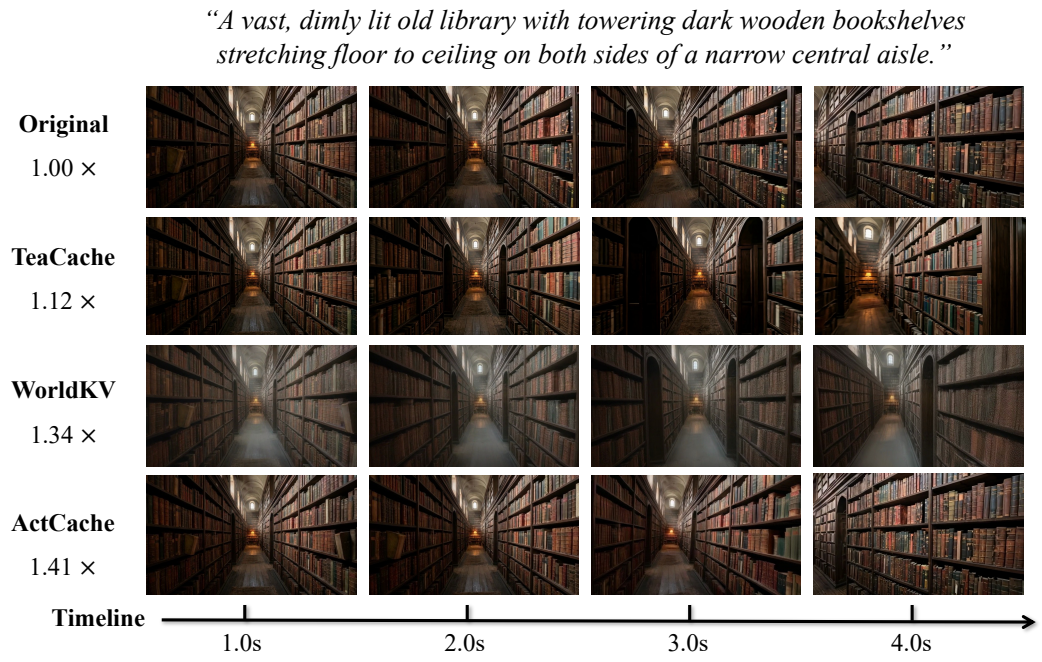


Figure 8: Qualitative examples for ActCache and other methods on Lingbot-World.



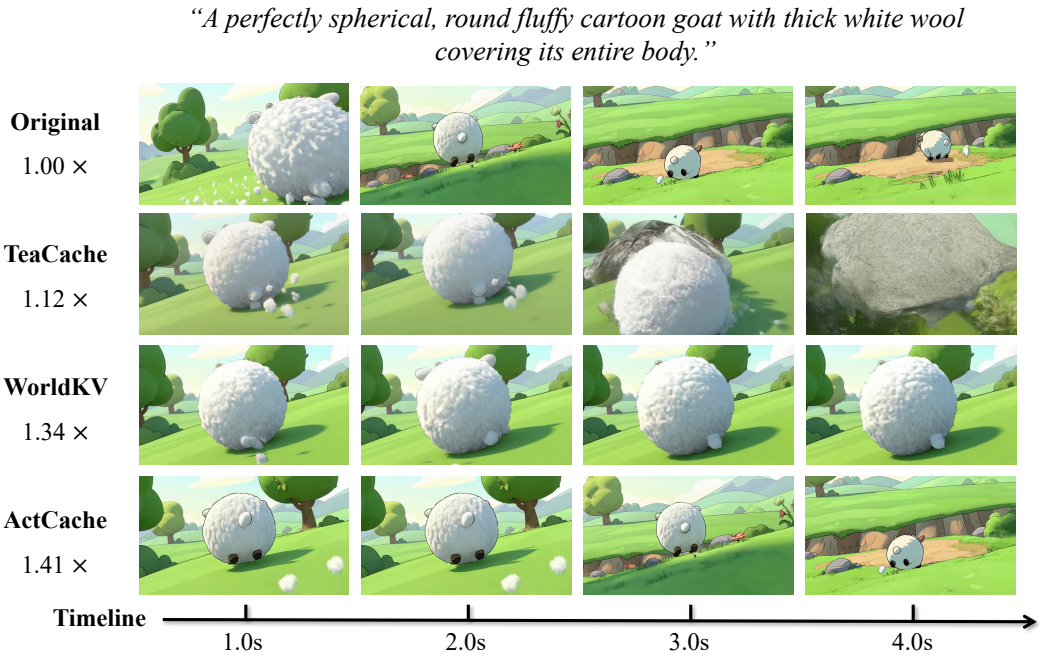
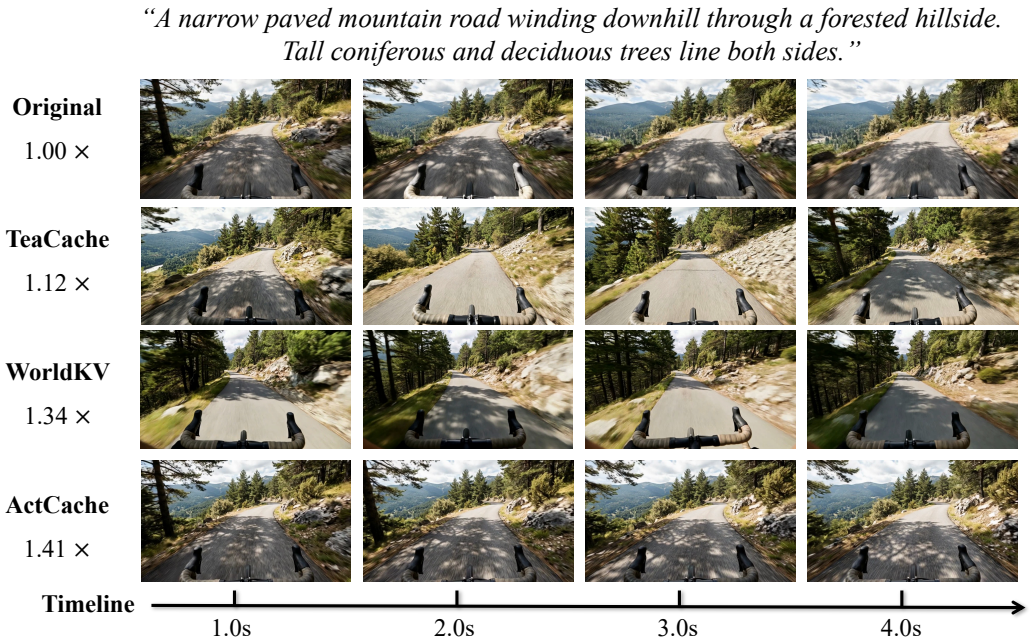


Figure 9: Qualitative examples for ActCache and other methods on Lingbot-World.



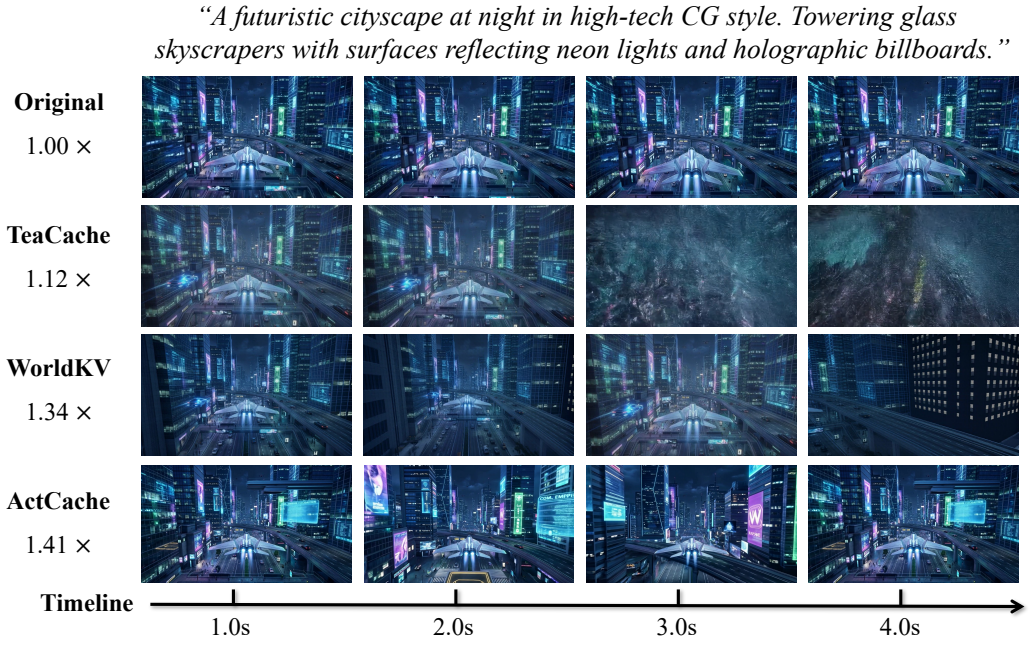


Figure 10: Qualitative examples for ActCache and other methods on Lingbot-World.

A.4 Limitation and Future Work

ActCache relies on explicit controls that can be converted into meaningful spatial displacement priors. Its benefits may therefore diminish under weakly structured controls, abrupt scene changes, or complex object motion that is poorly explained by camera dynamics. Moreover, the first denoising evaluation and the final memory-appending pass remain fully computed, while cross-attention and FFN modules stay dense, limiting the maximum achievable speedup. Future work could learn adaptive motion priors from controls and visual context, extend correspondence search across multiple historical chunks, and jointly sparsify other Transformer components. Evaluating ActCache on broader architectures, control modalities, and hardware platforms is another important direction.