ACTIONCACHE: ACTION-AWARE FEATURE CACHING FOR INTERACTIVE VIDEO WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Interactive video world models require both low-latency generation and prompt responses to user control. Existing training-free caches reuse along different axes while deciding validity from signals read inside the current model computation after the forward pass has begun. This internal gating has two limitations: (1) feature redundancy along the denoising trajectory collapses once models are distilled to three or four steps, and (2) the exogenous user action that invalidates the cache arrives before the forward pass, yet is folded into a model-side quantity and so reaches the decision only after that pass has begun. We present ACTIONCACHE, which reads the action before each forward pass and uses it as the governing signal for cache reuse: it recomputes when the control changes or the cached update grows too old, and reuses it otherwise. Because the criterion comes from the control stream, reuse survives few-step distillation. ACTIONCACHE operates directly on pretrained generators and integrates into existing interactive video world models. On three such models, spanning long diffusion sampling and distilled short-step streaming generation, ACTIONCACHE attains up to $3.25\times$ speedup while the action-response-latency increase stays within 0.2 frame on every backbone. These results point to a caching principle for interactive generation: the action stream as an ahead-of-time prior over cache validity.

1 INTRODUCTION

Turning generated video into an interactive world requires more than visual quality: the generated state must evolve with low latency and respond promptly to a continuous stream of keyboard, mouse, or controller inputs. Recent systems including Vid2World (Huang et al., 2026), Matrix-Game (He et al., 2025), and Oasis (Decart et al., 2024) make interactive generation feasible. Generator cost remains the bottleneck. Every generated chunk runs a full denoising loop, and every step of that loop re-evaluates the entire transformer stack. That repetition dominates wall-clock cost. Successive calls, however, often contain substantial redundancy: nearby steps recompute features that have barely moved. Cache reuse avoids repeated full-stack computation on compatible calls. Recent general-purpose methods, such as TeaCache (Liu et al., 2025a), TaylorSeer (Liu et al., 2025b), and EasyCache (Zhou et al., 2025), show that this model-side feature redundancy can be reused.

Whether the cached quantity is a block residual, a denoiser output, or a fused classifier-free-guidance (CFG) output, existing methods are distinguished by the signal they consult before authorising reuse. DiCache (Bu et al., 2026) uses model-side input signals, while FlowCache (Ma et al., 2026), ARCache (Nan et al., 2026), and WorldCache (Feng et al., 2026) reuse features across longer denoising trajectories or extended autoregressive executions. X-Cache (Zeng et al., 2026) folds the action into a cross-chunk fingerprint, so reuse survives few-step generation. The fingerprint is still evaluated inside the model. These existing methods decide cache validity from model-side signals that become available only after the forward pass has begun. They have two limitations: caches that read redundancy along the denoising trajectory have little left to reuse once an interactive model is distilled to 3–4 steps; and when the user changes action, the update computed under the previous control may become invalid at that instant, so the cached update is conservatively refreshed. Interactive generation provides an earlier, cheap-to-read signal: the user action a_t. In existing methods that action enters only as one ingredient of a composite model-side quantity. This suggests a more direct alternative: can cache reuse be scheduled from whether the control has changed, before the forward pass begins?

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

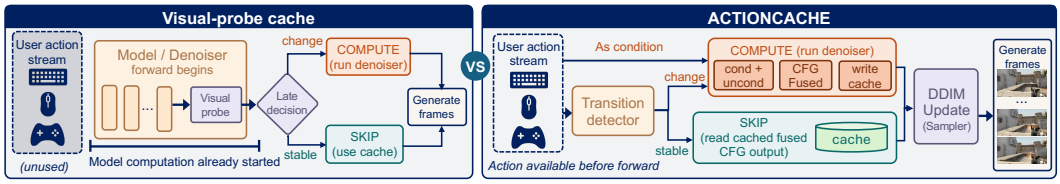

Figure 1: Action-aware caching versus visual-probe caching. Visual-probe caches schedule reuse from model-side features after the current forward pass has begun. ACTIONCACHE reads the user action before the forward pass, yielding an ahead-of-time prior over cache validity.

The causal order of interactive generation makes that schedule possible. The user action a_t is exogenous to the generator, arrives strictly before each forward pass, and influences the next latent state and frame:

$$\underbrace{a_t}_{\text{cause}} \rightarrow \underbrace{\text{state}_t}_{\text{latent}} \rightarrow \underbrace{f_t}_{\text{frame}}. \tag{1}$$

Equation 1 is a dependence schematic; history and sampling noise enter through state_t (Appendix A). Reading a_t therefore yields an ahead-of-time prior over cache validity before any backbone computation. Reading a visual probe runs part of the model, placing its cost on the call it skips (Figure 1). The action arrives earlier, outside the model, and marks the control change that stales the cache. Action alone is insufficient: scenes evolve under constant control, so the cache also needs an expiry. The policy is a causal signal plus a temporal bound: control changes or expiry trigger recomputation; all other calls reuse the cache.

We present ACTIONCACHE, a lightweight, training-free controller that decides cache validity before each forward pass. It operates directly on pretrained generators, preserves the backbone, and drops into existing interactive video world models. We make three contributions.

❶ We are the first to shift the cache-validity criterion in interactive world models from model-side features to the exogenous control stream, deciding reuse before any forward computation begins. This fundamentally decouples cache validity from trajectory length: while internal feature redundancy collapses under few-step distillation, the user action that actually invalidates cached states arrives strictly ahead of the forward pass, serving as an early, model-free gating signal.

❷ We propose ACTIONCACHE, a training-free controller that wraps the generator while preserving the backbone, built on a causal cache-validity tube: its instantaneous keyboard term prevents response lag and ghosting across action switches, its multi-scale mouse terms suppress coherent drift from sub-threshold increments, and its temporal age bound mitigates unobserved scene evolution. Inside the tube, the controller reuses with zero model-forward overhead; crossing a boundary forces a recompute and resets the cache anchor.

❸ Comprehensive empirical evaluations across diverse interactive world models confirm that our approach attains substantial efficiency gains while preserving control and perceptual fidelity. Under a shared protocol on three backbones (Vid2World's DDIM-50 sampler, Matrix-Game 2.0's distilled streaming generation, and an independently implemented Oasis-500M), against decision-rule ports of five caching methods, ACTIONCACHE attains $1.31\times$ to $3.25\times$ end-to-end speedup, with the action-response-latency increase remaining within 0.2 frame of the uncached baseline. On a single H100, it reaches 33.8 decoded FPS on Matrix-Game 2.0 streaming. As a plug-and-play stackable action gate, it lifts WorldCache's *released* HunyuanVoyager-13B and Aether-5B implementations from $3.21\times$ to $3.58\times$ and from $1.57\times$ to $1.81\times$. These results show that pre-forward action gating delivers consistent speedups across the evaluated backbones and cache hosts.

2 RELATED WORK

Training-free feature caching for video diffusion. A broad family of training-free caching methods for video diffusion answers the cache-validity question with input or visual probes of the current computation. TeaCache (Liu et al., 2025a) thresholds consecutive input embeddings, EasyCache (Zhou et al., 2025) accumulates a runtime stability indicator, TaylorSeer (Liu et al., 2025b) extrapolates fused outputs across denoising steps, and DiCache (Bu et al., 2026) combines a shallow-

layer probe with trajectory-matching residual reuse. Earlier variants (FasterCache (Lyu et al., 2025), AdaCache (Kahatapitiya et al., 2025), ProCache (Cao et al., 2026)) further trade probe granularity for per-step overhead. Recent systems specialise this idea for autoregressive and interactive generation: FlowCache (Ma et al., 2026) maintains chunk-specific policies across heterogeneous denoising stages, ARCache (Nan et al., 2026) uses history-guided scheduling and residual correction to limit cross-segment accumulation, and WorldCache (Feng et al., 2026) prioritises chaotic-token refreshes. X-Cache (Zeng et al., 2026) caches across generation chunks using a structure- and action-aware fingerprint, and we retain it as an action-aware few-step baseline, although its fingerprint still has to be evaluated inside the model before reuse is authorised. Across all these lines, validity is decided from model-side or composite signals that require the forward pass to have begun. ACTIONCACHE reads the action stream before that pass and decides skips ahead of model execution.

Interactive world models and action-conditioned generation. Recent interactive world models treat the user action stream as a conditioning input *inside* the generator: Matrix-Game (He et al., 2025) is the first open-source streaming model (distilled WAN backbone, three denoising steps), the closed-source Genie 3 (Google DeepMind, 2025) reaches 24 FPS at higher quality, and WorldPlay (Sun et al., 2025), WorldScape (WorldScape Team, 2025), and MineWorld (Guo et al., 2025) extend this line via context forcing, few-step KV schemes, and diagonal decoding. A complementary line studies action-conditioned next-frame prediction at training time, including Vid2World (Huang et al., 2026) (an evaluation backbone here) and GameFactory (Yu et al., 2025). These works establish action as generator conditioning. Cache reuse during inference remains independent of control transitions. Consulting a control signal before the computation it triggers is meanwhile a long-standing systems pattern, from CPU branch prediction to cloud-gaming input prediction (Outatime (Lee et al., 2015)). ACTIONCACHE imports that ordering into interactive generation, deciding cache validity from the control signal before the generator is invoked.

3 METHOD: ACTIONCACHE

ACTIONCACHE is a drop-in, training-free controller that wraps the expensive denoiser or transformer-block computation of an interactive video world model. Its central object is a *causal cache-validity tube*: an axis-aligned region over observable action change and per-slot cache age within which reuse is admissible before the current forward pass begins. A COMPUTE call is triggered whenever an action switch leaves the tube along its control axes, cache age reaches the bound K, or a forced-recompute condition (cache miss, warm-up, or last-step context hand-off) applies; each event also resets the cache anchor. Otherwise, the controller emits SKIP. The policy thus unifies transition-driven recomputation with periodic refresh.

The controller depends solely on the action stream and cache metadata. It reads zero backbone-side features from the pretrained generator, so it can serve as a standalone policy or a speed-oriented stack around an input-probe or forecasting cache (Appendix G.3). Its action comparison is microsecond-scale and negligible against a denoiser call (Appendix H, Table 22). Figure 2 gives an overview.

3.1 PRELIMINARIES

An interactive video world model generates frames from a stream of user actions $a_1, a_2, \ldots$ and an initial image f_0. Its DiT (Peebles & Xie, 2023) backbone processes a latent chunk of F frames at a time and, for each chunk c, runs a short denoising loop of N forward calls at slot indices $t=0,\ldots,N-1$. The final slot also refreshes the autoregressive KV cache. Each call executes the full transformer-block stack and dominates wall time. Backbone-specific values of F and N and the generator-forward signature are given in Appendix A.

Block residual caching. Inside the generator, the operation that matters for caching is the transformer-block stack. Let $x_{c,t} \in \mathbb{R}^{B\times L\times D}$ be the per-token input to the first block (after patch embedding) and $y_{c,t}$ the corresponding output. We call

$$r_{c,t} \triangleq y_{c,t} - x_{c,t} \tag{2}$$

the *block residual*, abbreviated r_c where the slot is fixed. Video-diffusion caches exploit the empirical observation that $r_{c,t}$ changes slowly enough between consecutive same-slot calls to be replaced by a

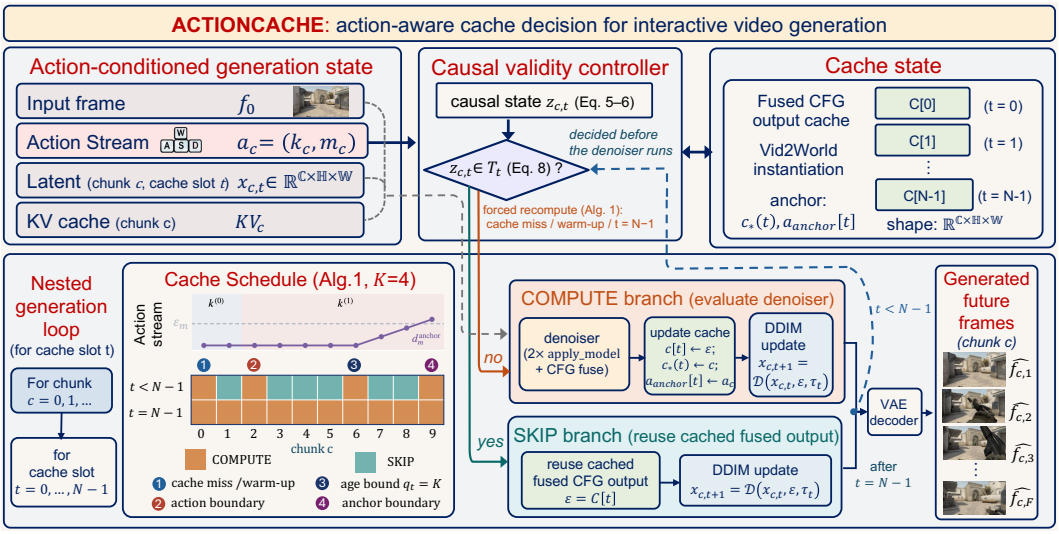

Figure 2: Overview of ACTIONCACHE. The causal validity controller reads the action stream and cache metadata before the denoiser: it recomputes at action or age boundaries and under the forced-recompute conditions, and otherwise reuses the cached update. It is instantiated on Vid2World, where ACTIONCACHE reuses the `DDIMSampler`'s fused CFG output and preserves the denoiser.

constant-cost addition:

$$y_{c,t} \;\approx\; x_{c,t} + r^{\text{cached}}_{c-1,t}. \tag{3}$$

Two relative drifts measure that change, one against the previous call and one against the slot's last refresh $c_*(t)$:

$$\delta_{c,t} = \frac{\|r_{c,t} - r_{c-1,t}\|_1}{\|r_{c-1,t}\|_1}, \qquad \widetilde{\delta}_{c,t} = \frac{\|r_{c,t} - r_{c_*(t),t}\|_1}{\|r_{c_*(t),t}\|_1}, \tag{4}$$

and a skip risks the second. Both are substantial on a deployed backbone: Matrix-Game block residuals average $\delta = 0.893$ within an action and 1.055 across a control transition, so reuse holds within a drift tolerance τ and the gate reads that separation ahead of time (Appendix F). While visual-probe caches estimate $\delta_{c,t}$ from the current computation, and can do so only after the chunk has begun to be processed, ACTIONCACHE gates on the action before that computation starts. In both cases the criterion is defined on the tensor the cache actually reuses: the block residual r_c in residual caches, or the cached fused CFG output for Vid2World's sampler-level cache (§3.4, Appendix E).

3.2 CAUSAL CACHE-VALIDITY STATE

For an interactive world model the user action a_c decomposes into a discrete part (keyboard $k_c \in \{0,1\}^{D_k}$) and an optional continuous part (mouse / controller $m_c \in \mathbb{R}^{D_m}$). For cache slot t, let $c_*(t)$ denote its most recent refresh chunk and define the observable causal state

$$z_{c,t} \triangleq \left(d_k,\; d_m^{\text{local}},\; d_m^{\text{anchor}},\; q_t\right), \qquad q_t = c - c_*(t), \tag{5}$$

where

$$d_k = \|k_c - k_{c-1}\|_\infty, \qquad d_m^{\text{local}} = \|m_c - m_{c-1}\|_2, \qquad d_m^{\text{anchor}} = \|m_c - m_{c_*(t)}\|_2. \tag{6}$$

The first two coordinates detect an instantaneous control event, the third detects coherent sub-threshold increments accumulated since the cached action was recorded, and q_t is the age of that cached update. All four coordinates precede the generator forward and use only action and cache metadata.

The action component leaves its admissible range iff

$$\mathbb{1}\left[\, d_k > \varepsilon_k \;\vee\; d_m^{\text{local}} > \varepsilon_m \;\vee\; d_m^{\text{anchor}} > \varepsilon_m \,\right] = 1. \tag{7}$$

We use $\varepsilon_k = 10^{-3}$ and $\varepsilon_m = 0.05$ (Matrix-Game scales mouse to ± 0.1). Equation 7 is deterministic once these thresholds are fixed and costs $O(D_k + D_m)$. Its placement before the denoiser prevents

speculation across the control event that most directly changes the next visual state, addressing semantic lag or ghosting at action change. The endpoint coordinate is essential for continuous control: it catches coherent increments that are individually sub-threshold but collectively move away from the cached action. Action coordinates alone remain incomplete because latent and non-action context can drift. The age coordinate supplies the complementary temporal boundary.

3.3 Cache-validity tube and controller

We collect the four admissibility conditions into the per-slot *causal cache-validity tube*

$$\mathcal{T}_t \triangleq \left\{ z_{c,t} : d_k \le \varepsilon_k,\ d_m^{\text{local}} \le \varepsilon_m,\ d_m^{\text{anchor}} \le \varepsilon_m,\ q_t < K \right\}. \tag{8}$$

Here K is the maximum reuse distance per slot. The tube is a normalized encoding of the transition thresholds and periodic refresh.

Algorithm 1 (Appendix B) implements this controller. While $z_{c,t}$ remains inside $\mathcal{T}_t$, the controller reuses the cached update. Crossing an action or age boundary triggers a recomputation, which refreshes the cached tensor and applies the reset map $c_*(t) \leftarrow c$ and $a_{c_*(t)} \leftarrow a_c$, so the slot restarts from the current action. The cache-empty, warm-up, and final-slot checks are forced-recompute conditions for states in which the tube is unavailable or deliberately disabled. Checking cache availability first also ensures that the anchor action is read only when it is defined.

With $W=1$, $K=4$ for Vid2World, and forced refresh at $t = N-1$, the controller is conservative at the hand-off that determines the next chunk's autoregressive state. Every call inside $\mathcal{T}_t$ that passes the forced-recompute conditions is reused with zero added model-side computation.

3.4 Drift bound and operating point

We use $(\varepsilon_m, K, W) = (0.05, 4, 1)$, selected from the sensitivity sweeps of Appendix G.2. $\varepsilon_k = 10^{-3}$ and the last-step refresh are fixed protocol choices. The geometry separates their roles: ε_m is the radius of the tube along the continuous-control axes, whereas K is its temporal extent. For any Skip, tube membership gives $\|m_c - m_{c_*}\|_2 \le \varepsilon_m$ and $q_t \le K-1$, so under the joint-Lipschitz and bounded-context assumptions of Appendix E, $\|r_c - r_{c_*}\|_1 \le L_a \varepsilon_m + (K-1) L_h \beta$: the endpoint boundary limits action-conditioned drift and the age boundary limits accumulation from unobserved context. This conditional drift bound uses fixed L_a, L_h, β supplied by its assumptions.

Within the policy class that may reuse only inside $\mathcal{T}_t$ under the same forced-recompute conditions, Algorithm 1 skips every admissible call. The policy is thus the indicator of a causal admissible set, augmented by reset events at states where its assumptions are least reliable. Appendix E gives the precise conditions, the cost accounting, and the finite-sample checks of the selected tube. Appendix H reports the corresponding stress-test experiments and profiling protocol.

4 Experiments

We evaluate ActionCache on three interactive world-model backbones: Vid2World (§4.1) with its autoregressive DDIM-50 sampler, Matrix-Game 2.0 (§4.2) with its short distilled streaming generation, and independently implemented Oasis-500M as a cross-backbone confirmation (Table 5). We report four shared indicators in two groups. *Control fidelity* comprises **ARL** (Action-Response Latency, frames between a control transition and the first visibly responding frame) and **AFS** (Action-Following Score, cosine alignment between the action-encoded intended direction and the observed optical flow), while *perceptual fidelity* comprises **LPIPS** (Zhang et al., 2018) and **FVD** (Unterthiner et al., 2019) against a reference clip. ARL and AFS measure control-conditioned behavior, complementing PSNR / LPIPS / FVD. Their definitions and per-backbone reference-clip choice are in Appendix C, and because both are defined by us, §4.3 validates them against measurements that are independent of the method. Every comparison method is reported at its *per-backbone Pareto-frontier* operating point (configurations and sweep grids in Appendix I), and ActionCache is tuned under the same protocol from the sweeps in Appendix G.2.

Table 1: Main experiment I: ACTIONCACHE on the 500-episode Vid2World/DIAMOND test split.

Method	Skip %	Wall (s/ep)	Speedup	PSNR↑	LPIPS↓	FVD↓	ARL↓	AFS↑
Baseline	0.00	659.70	1.00×	19.97	0.314	95.3	1.04	0.082
DiCache (Bu et al., 2026)	52.87	320.06	2.06×	19.56	0.333	114.7	5.13	0.061
FlowCache (Ma et al., 2026)	65.41	238.18	2.77×	**19.74**	0.326	**105.6**	2.34	0.076
ARCache (Nan et al., 2026)	58.13	286.44	2.30×	19.53	0.335	117.2	3.51	0.066
WorldCache (Feng et al., 2026)	62.88	254.05	2.60×	19.61	0.328	111.8	2.71	0.073
X-Cache (Zeng et al., 2026)	54.76	306.07	2.16×	19.45	0.339	119.4	2.02	0.072
ACTIONCACHE (ours)	**70.29**	**202.77**	**3.25**×	19.68	**0.322**	111.6	**1.13**	**0.089**

Boldface marks the best result among caching methods in each column.

4.1 LONG MULTI-STEP SAMPLERS: VID2WORLD (DDIM-50)

Table 1 asks whether ACTIONCACHE gives practical speedups on Vid2World (Huang et al., 2026), an ICLR '26 model that turns the 1.4B DynamiCrafter image-to-video checkpoint into a causal, action-conditioned world model. Vid2World uses a long DDIM-50 loop with `uniform_trailing` timestep spacing and CFG 2.5, exposing many denoising calls per chunk for the cache to act on.

Setup. We evaluate on the official CSGO autoregressive pipeline of Vid2World (Huang et al., 2026), whose corpus is the DIAMOND (Alonso et al., 2024) pre-processing of the Pearce–Zhu CS:GO behavioural-cloning release (Pearce & Zhu, 2022). The release provides a 5,003-episode training split and an official 500-episode DIAMOND test partition (the same indices as DIAMOND's `test_split.txt`). There is no separate validation split. Every cache hyperparameter and per-method Pareto operating point is selected on the training split and then frozen (Appendix I). Table 1 scores each frozen configuration on the full 500-episode test split, averaged over three independent sampling seeds, so all methods land on the same episode lists. No test episode is used for hyperparameter or Pareto selection. Wall time is measured on 4×A100-40GB and reported as the single-GPU-equivalent per-episode mean. Beyond the four shared indicators we report **PSNR** against the ground-truth CSGO clip, which is also the LPIPS / FVD reference. Checkpoint, episode counts, resolution, conditioning length, and the sampling configuration are listed in Appendix D.1.

Comparison methods. We compare with DiCache (Bu et al., 2026) and four recent methods that target interactive, autoregressive, or world-model inference: FlowCache (Ma et al., 2026), ARCache (Nan et al., 2026), WorldCache (Feng et al., 2026), and X-Cache (Zeng et al., 2026). Their native releases are bound to their own backbones. This table compares skip/compute decision rules on a shared cache substrate: we port each decision rule to Vid2World's fused-CFG-output cache and report its per-backbone Pareto-frontier operating point. All seven rows share the same checkpoint, sampling hyperparameters, test episodes, seed list, and end-to-end wall-time definition. Which components are retained or dropped is recorded in Appendix D.1.

ACTIONCACHE is integrated as a local wrapper around Vid2World's `DDIMSampler` that answers a SKIP/COMPUTE query at every (frame, DDIM-step) pair using Algorithm 1, and disabling it restores byte-identical upstream baseline behaviour (Appendix D.1).

ACTIONCACHE reaches a useful skip regime under the frozen test-split evaluation. With `force_refresh_every`=4 and the last-step refresh, the maximum per-frame skip rate is 75% minus transition-triggered recomputes. ACTIONCACHE reaches 70.29%, reducing single-GPU sampling time from 659.70 s to 202.77 s (3.25×) with only a 0.29 dB PSNR loss, +0.008 LPIPS, and +16.3 FVD relative to BASELINE, using one fixed configuration across clips (Figure 4, Appendix H).

Under this decision-rule comparison, the reported operating point leads on acceleration and control while keeping reconstruction stable. In Table 1, ACTIONCACHE is the only cache that is jointly best on skip rate, speedup, ARL, and AFS. The closest long-sampler competitor on wall time is FlowCache at 2.77×, and it trails ACTIONCACHE's 3.25× by 1.21 frames of ARL. Only ACTIONCACHE keeps ARL within 0.1 frame of BASELINE.

Matrix-Game's 3-step distilled generation leaves less room to skip than Vid2World's 50-step sampler. ACTIONCACHE remains useful at both depths: it reads action change and cache age.

Table 2: Core 2×2 ablation on Vid2World (same operating point as Table 1). PSNR interaction is -0.24 dB (95% CI $[-0.31, -0.15]$).

Configuration	Skip	Wall (s)	PSNR	LPIPS	ARL	False skip
Full	70.29%	202.77	19.68	0.322	1.13	3.08%
−Transition	74.12%	181.7	18.05	0.401	2.91	17.6%
−Periodic	84.67%	108.4	17.80	0.418	1.28	21.9%
−Transition −Periodic	89.8%	83.6	15.93	0.514	6.58	40.7%

Table 2 is the 2×2 factorial on the same frozen test split and shows that both tube axes are needed. Dropping the transition detector loses 1.63 dB at $+3.83$ skip points and raises ARL to 2.91 frames, the same over-skip-at-control-change mode as DiCache above. Dropping periodic refresh loses 1.88 dB at 84.67% skip, so a fixed action still permits residual drift. Dropping both collapses ARL to 6.58 frames at 40.7% false skip. Appendix G.1 reports last-step and warm-up effects (0.31–0.76 dB).

4.2 Few-step distilled streaming models: Matrix-Game 2.0

Backbone. We also evaluate ActionCache on Matrix-Game 2.0 (He et al., 2025), an open-source streaming world model with a $\sim$1.8 B distilled DiT backbone, 30 transformer blocks per chunk, 3 denoising steps per chunk (warped timesteps $\{1000, 666, 333\}$), and 3 latent frames per block. All experiments use the universal checkpoint on four A100 GPUs. We take Matrix-Game 2.0 as the representative of this class: distillation to 3–4 denoising steps combined with block-wise streaming generation, which removes the long denoising trajectory that long-sampler caches rely on. We confirm the same behaviour on the independently implemented Oasis-500M (Table 5).

Action scenarios. The six evaluation sequences Matrix-Game 2.0 ships, retained verbatim, split into two regimes. *Stable trajectories* are three single-action, near-stationary streams: Forward (constant W), Idle (no action), and CameraR (mouse pan right). *Complex scenes* are three continuously evolving sequences with sustained control changes: Universal (Unreal-engine generic scene, mixed keyboard and mouse drift), GTA Drive (continuous steering and braking), and Temple Run (keyboard-only parkour), the last using 4 denoising steps, one more than the other scenes, which raises both its baseline wall time and its skip ceiling. Confirmatory inference uses 30 independent rollouts per complex scene with a debiased FVD estimator (Appendix H, Table 14). The debiased pooled FVD 96.8 $[89.4, 104.6]$ contains the $n = 3$ main-table value. Comparison methods follow the same Pareto-frontier selection as §4.1 on Matrix-Game's per-block-stack residual cache, so only the skip/compute rule differs across rows. Appendix D.2 gives the remaining protocol details.

Perceptual reference and reporting. Since Matrix-Game's autoregressive world model produces no ground-truth video, both LPIPS and FVD use the upstream Matrix-Game baseline output as the perceptual reference. Per-scene FVD in this comparison uses $n = 3$ clips per method so that every cache is scored under the same budget. Table 14 (Appendix H) shows that this estimator has only modest finite-sample bias ($+6.7$) and that 103.5 lies inside the $n = 30$ debiased interval.

For scale, three-seed re-runs of the fixed upstream reference give LPIPS 0.024 and FVD 87.4, so ActionCache's 0.041 sits at $\sim 1.7\times$ this floor. AFS is read *relatively* per backbone, since its absolute scale is set by the RAFT (Teed & Deng, 2020) noise floor independently of the cache (Appendix C.3).

In Table 3, FlowCache, ARCache, and WorldCache target multi-step or long-rollout redundancy. On Matrix-Game's 3–4-step generation several of their operating points fall at or below $1.1\times$: FlowCache is slower than Baseline on Universal ($0.98\times$) and the latter two obtain only mild speedups there ($1.01\times$ / $1.03\times$) at higher ARL (2.41 and 2.86 frames). X-Cache, which reuses residuals across generation chunks, is the strongest few-step competitor and is comparable on LPIPS/FVD ($0.039/101.4$ vs. ActionCache $0.041/103.5$). The 2.1 FVD gap is small relative to the debiased interval in Appendix H, Table 14. Its speedups stay in the 1.11–$1.18\times$ band, below ActionCache on every timing block, at weaker control fidelity (ARL 1.48, AFS 0.052).

ActionCache is non-dominated under the stated joint control-fidelity $\times$ wall-time criteria. The transition detector forces a recompute exactly at the transition frames, so cross-scene mean ARL $=$ 1.19 is at most 0.10 frames above Baseline (1.09) and mean AFS is within 0.004 ($0.058 \to 0.054$),

Table 3: Main experiment II: Matrix-Game 2.0 (universal mode, 30 latent frames). Timing blocks are the median of 3 runs. "Controlled transition (avg.)" averages ARL / AFS / LPIPS / FVD across the three complex scenes. Darker cells indicate better values within each metric.

	Wall time (s) ↓				**Speedup ↑**				**Skip rate (%) ↑**				**Control / quality avg.**			
	Stable	Univ.	GTA	Temple	Stable	Univ.	GTA	Temple	Stable	Univ.	GTA	Temple	ARL ↓	AFS ↑	LPIPS ↓	FVD ↓
Baseline	16.63	16.84	18.26	22.64	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00	1.09	0.058	0.024	87.4
DiCache	16.68	17.21	18.11	21.58	1.00	0.98	1.01	1.05	5.83	2.91	5.64	11.28	1.37	0.051	0.044	105.8
FlowCache	16.54	17.12	18.55	21.29	1.01	0.98	0.98	1.06	6.17	2.68	5.21	13.05	1.61	0.052	0.047	110.2
ARCache	16.16	16.62	17.59	20.61	1.03	1.01	1.04	1.10	9.54	6.37	9.82	18.44	2.41	0.046	0.058	131.7
WorldCache	16.02	16.38	17.27	20.18	1.04	1.03	1.06	1.12	12.18	8.74	12.35	21.56	2.86	0.043	0.067	141.3
X-Cache	14.71	15.17	16.22	19.12	1.13	1.11	1.13	1.18	24.63	17.84	20.71	27.35	1.48	0.052	0.039	101.4
ActionCache	**11.95**	**12.85**	**12.91**	**14.16**	**1.39**	**1.31**	**1.41**	**1.60**	**60.00**	**40.25**	**50.80**	**63.70**	**1.19**	**0.054**	**0.041**	**103.5**

Table 4: Interactive-rate throughput on H100 (GTA DRIVE). Operating points from Table 3. Decoded-frame FPS is 120 RGB frames (30 latent × 4) divided by end-to-end wall time.

Method	FPS↑	Wall (s)	Speedup↑	Skip %↑	ARL↓	AFS↑	LPIPS↓	FVD↓
Baseline	24.3	4.94	1.00×	0.0	1.09	0.057	0.024	86.8
DiCache (Bu et al., 2026)	24.7	4.86	1.02×	5.9	1.41	0.050	0.045	107.2
FlowCache (Ma et al., 2026)	23.8	5.04	0.98×	4.7	1.66	0.051	0.048	111.9
ARCache (Nan et al., 2026)	25.4	4.72	1.05×	10.3	2.48	0.045	0.060	133.6
WorldCache (Feng et al., 2026)	25.9	4.63	1.07×	12.8	2.91	0.042	0.069	144.1
X-Cache (Zeng et al., 2026)	27.6	4.35	1.14×	21.2	1.51	0.051	**0.038**	**99.6**
ACTIONCACHE (ours)	**33.8**	**3.55**	**1.39**×	**51.1**	**1.20**	**0.053**	0.040	101.7

Boldface marks the best result among caching methods in each column.

while ACTIONCACHE's LPIPS (0.041) is within 0.002 of X-Cache's 0.039 despite the highest skip rate (51.58%) among methods within 1.5 frames of Baseline's ARL. Under these predeclared thresholds, ACTIONCACHE is the sole measured operating point combining sub-1.5-frame ARL, AFS within 7% of Baseline, ≤ 0.05 LPIPS, and $\geq 1.3\times$ speedup. It also delivers the highest peak speedup (1.60×) among the six cache methods with sub-1.5-frame ARL.

ACTIONCACHE's behaviour adapts to the regime. On the *Stable trajectories* block it reaches the 60% skip ceiling imposed by the always-refresh-on-last-step rule, kept on for a correct KV cache hand-off. On the *complex scenes* it sustains 1.41× on GTA DRIVE and reaches a study-peak 1.60× at 63.7% skip on the 4-step TEMPLE RUN, since long stable spans allow sustained reuse while transitions correctly force refreshes.

On a single H100, the setting at which Matrix-Game 2.0 reports 25 FPS streaming, the same frozen GTA DRIVE operating point runs at **33.8** decoded FPS (1.39×, 51.1% skip) with ARL 1.20 (Table 4). It is the only measured method above both 25 FPS and a 30 FPS interactive threshold.

ACTIONCACHE also works as a retrofit on an already deployed cache. Layering its action gate on three Matrix-Game hosts (DiCache, WorldCache, X-Cache) raises skip by 11.7 to 16.9 pp and speedup by 0.14× to 0.19× per host while ARL *falls* by 0.08–0.29 frame. Even X-Cache, whose fingerprint is already action-aware, climbs from 21.97% to 33.62% skip, so the gate adds reuse beyond its fingerprint. The same skip-side union on WorldCache's released HunyuanVoyager-13B (Huang et al., 2025) and Aether-5B (Zhu et al., 2025) implementations raises end-to-end speedup from 3.21× to 3.58× and from 1.57× to 1.81×, cutting ARL from 1.76 / 1.58 to 1.22 / 1.23 frames against the uncached Original (Appendix G.3, Table 13). Standalone ACTIONCACHE remains non-dominated on the joint speed–control frontier there as well.

Confirmatory runs hold at larger scale: 30 rollouts per complex scene give a pooled median 1.44× at ΔARL +0.10, and the independently implemented Oasis-500M backbone reaches 1.52× at ΔARL +0.13 frame (Table 5), so all three backbones accelerate within +0.2 frame of baseline latency. Expanded intervals remain in Appendix H.

Table 5: Cross-backbone trade-off for ACTIONCACHE. ΔARL and quality change are versus each backbone's uncached baseline.

Backbone	Sampler	Speedup ↑	Skip ↑	ΔARL ↓	Quality change
Vid2World	DDIM-50	3.25	70.3%	+0.09	−0.29 dB PSNR
Matrix-Game 2.0	3-step distilled	1.44	51.6%	+0.10	+0.017 LPIPS
Oasis-500M	4-step distilled	1.52	55.4%	+0.13	+0.021 LPIPS

4.3 DIAGNOSTICS: METRICS, THE ACTION–DRIFT PREMISE, AND STRESS TESTS

The control metrics survive independent measurement. ARL and AFS are defined by us, and ACTIONCACHE forces a recompute exactly on the transition frames where ARL is read, so we re-measure the ordering with three instruments whose threshold, reference frame, and deciding agent are all independent of the method: 48 blind non-author raters, a third-party inverse-dynamics model trained only on real play and then frozen, and, on Vid2World, metrics re-anchored to the ground-truth CSGO clip. All three reproduce the ARL ordering (Spearman $|\rho| \geq 0.95$ over five independent indicators), forced-choice preference of ACTIONCACHE over BASELINE is $\approx 50\%$ on both backbones, and the ground-truth-anchored action accuracy and response delay read 78.4 versus 79.3 and 1.23 versus 1.17 frames (Appendix C.4).

Action provides a partial drift signal. On 14,934 Matrix-Game block-residual samples, instrumented on the tensor the cache actually reuses (§3.1), discretised action classes separate the relative L1 drift mainly at the near-static end: the pure idle class sits at $\bar{\delta} \approx 0.51$ and one single-key class at $\bar{\delta} \approx 0.77$, against $\bar{\delta} \approx 1.03$–$1.09$ for the remaining 17 classes. A one-way ANOVA over 19 classes attributes $\eta^2 = 0.283$ of drift variance to the action label ($F_{18,14915} = 326.9$), an effect concentrated in those two rare classes. Only 3.17% of same-action pairs exceed the diagnostic tolerance at $\tau = 1.10$ (0.26% at $\tau = 1.20$). The action stream is therefore a partial prior, complemented by the age boundary (Table 2). The per-class plot (Appendix F, Fig. 3) and the remaining diagnostics are in Appendices F and H.

Under stress the policy falls back to the baseline. A detector reading only consecutive actions misses slow coherent drift: under sustained sub-threshold mouse motion, 27.4% of reused residuals exceed the $\tau = 1.10$ tolerance, which the deployed endpoint-aware check cuts to 3.6% at $1.33\times$ speedup (Appendix H, Table 19). At a transition on every chunk, skip falls to zero, output hashes match Baseline in 100% of runs, and measured action-check overhead is 0.008%.

The tube provides a stronger speed–quality balance than a learned skip policy. The Full row of Table 2 skips every call admitted by the action-change ∪ age-K rule; the consecutive-only detector above uses only the consecutive coordinate. On Matrix-Game complex scenes a small learned action scheduler skips slightly more (53.9% vs. 51.6%), with worse ARL, LPIPS, and FVD (1.22/0.045/108.7 vs. 1.19/0.041/103.5). ACTIONCACHE attains 76% of the offline oracle's speedup increment (Appendix H, Table 16). Appendices G and H report the remaining hyperparameter sweeps, stacking, stress-test, and profiling results.

5 CONCLUSION

We presented ACTIONCACHE, a training-free feature cache that reads the user action before the forward pass and issues SKIP/COMPUTE with a microsecond-scale query. Organising action change and cache age as a cache-validity tube reaches $3.25\times$ speedup at 70.3% skip on Vid2World's official test split, holds the ARL increase within 0.2 frame on all three backbones, and stacks on every host cache we tested. The evaluation has three limitations: it uses the released open checkpoints and the scenarios each backbone ships; the main tables compare skip/compute decision rules on a shared cache substrate; and the $1.60\times$ Matrix-Game figure is the four-step TEMPLE RUN peak, with pooled median $1.44\times$. Appendix J records scope, metric, and comparison details.

AI USE STATEMENT

In this work, we used generative AI tools to edit the manuscript for readability and to assist with part of the code implementation. All AI-assisted text and code were reviewed and verified by the authors. We take responsibility for the final content of this work.

REFERENCES

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in Atari. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. CS:GO branch: 5,003 training episodes and a 500-episode held-out test split (`test_split.txt`); see https://github.com/eloialonso/diamond/tree/csgo.

Jiazi Bu, Pengyang Ling, Yujie Zhou, Yibin Wang, Yuhang Zang, Dahua Lin, and Jiaqi Wang. Dicache: Let the diffusion model determine its own cache. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://mlanthology.org/iclr/2026/bu2026iclr-dicache/.

Fanpu Cao, Yaofo Chen, Zeng You, and Wei Luo. Procache: Constraint-aware feature caching with selective computation for diffusion transformer acceleration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 19862–19870, 2026. URL https://doi.org/10.1609/aaai.v40i24.39069.

Decart, Julian Quevedo, Quinn McIntyre, Spruce Campbell, Xinlei Chen, and Robert Wachen. Oasis: A universe in a transformer. Project page, 2024. URL https://oasis-model.github.io/.

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. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2603.06331.

Google DeepMind. Genie 3. Project page, 2025. URL https://deepmind.google/models/genie/. Closed-source release.

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. Technical report, 2025. URL https://github.com/microsoft/mineworld.

Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, Baixin Xu, Hao-Xiang Guo, Kaixiong Gong, Cyrus Wu, Wei Li, Xuchen Song, Yang Liu, Eric Li, and Yahui Zhou. Matrix-game 2.0: An open-source, real-time, and streaming interactive world model. Project page, 2025. URL https://matrix-game-v2.github.io/.

Siqiao Huang, Jialong Wu, Qixing Zhou, Shangchen Miao, and Mingsheng Long. Vid2world: Crafting video diffusion models to interactive world models. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=pFyzqbUiF9.

Tianyu Huang, Wangguandong Zheng, Tengfei Wang, Yuhao Liu, Zhenwei Wang, Junta Wu, Jie Jiang, Hui Li, Rynson W. H. Lau, Wangmeng Zuo, and Chunchao Guo. Voyager: Long-range and world-consistent video diffusion for explorable 3D scene generation. *arXiv preprint arXiv:2506.04225*, 2025. URL https://arxiv.org/abs/2506.04225.

Kumara Kahatapitiya, Haozhe Liu, Sen He, Ding Liu, Menglin Jia, Chenyang Zhang, Michael S. Ryoo, and Tian Xie. Adaptive caching for faster video generation with diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15240–15252, October 2025.

Kyungmin Lee, David Chu, Eduardo Cuervo, Johannes Kopf, Yury Degtyarev, Sergey Grizan, Alec Wolman, and Jason Flinn. Outatime: Using speculation to enable low-latency continuous interaction for mobile cloud gaming. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pp. 151–165, 2015. doi: 10.1145/2742647.2742656. URL https://doi.org/10.1145/2742647.2742656.

Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7353–7363, June 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 (ICCV)*, pp. 15853–15863, October 2025b.

Zhengyao Lyu, Chenyang Si, Junhao Song, Zhenyu Yang, Yu Qiao, Ziwei Liu, and Kwan-Yee K. Wong. Fastercache: Training-free video diffusion model acceleration with high quality. In *International Conference on Learning Representations (ICLR)*, pp. 33132–33156, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/518046d86bbc41a0707727c38301ad8e-Paper-Conference.pdf.

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. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=vko4DuhKbh.

Kepan Nan, Wangbo Zhao, Penghao Zhou, Jun Li, Zhenheng Yang, Jian Yang, and Ying Tai. Accelerating autoregressive video diffusion via history-guided cache and residual correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 43740–43750, June 2026.

Tim Pearce and Jun Zhu. Counter-strike deathmatch with large-scale behavioural cloning. arXiv:2104.04258, 2022. URL https://arxiv.org/abs/2104.04258. Source of the 5.5M-frame / 95-hour CS:GO online-gameplay dataset reused by DIAMOND Alonso et al. (2024) and Vid2World Huang et al. (2026).

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, October 2023.

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. Technical report, 2025. URL https://github.com/Tencent-Hunyuan/HY-WorldPlay.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 402–419, 2020.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. FVD: A new metric for video generation. Deep Generative Models for Highly Structured Data, ICLR Workshop, 2019. URL https://openreview.net/forum?id=rylgEULtdN.

WorldScape Team. Worldscape: An efficient real-time interactive world model with universal control. Technical report, 2025. URL https://manifoldai.cn/assets/file/WorldScape.pdf.

Jiwen Yu, Yiran Qin, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Gamefactory: Creating new games with generative interactive videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11590–11599, October 2025.

Yixiao Zeng, Jianlei Zheng, Chaoda Zheng, Shijia Chen, Mingdian Liu, Tongping Liu, Tengwei Luo, Yu Zhang, Boyang Wang, Linkun Xu, Siyuan Lu, Bo Tian, and Xianming Liu. X-Cache: Cross-chunk block caching for few-step autoregressive world models inference, 2026. URL https://arxiv.org/abs/2604.20289.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018.

Xin Zhou, Dingkang Liang, Kaijin Chen, Tianrui Feng, Xiwu Chen, Hongkai Lin, Yikang Ding, Feiyang Tan, Hengshuang Zhao, and Xiang Bai. Less is enough: Training-free video diffusion acceleration via runtime-adaptive caching. Project page, 2025. URL https://h-embodvis.github.io/EasyCache/.

Haoyi Zhu, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Chunhua Shen, Jiangmiao Pang, and Tong He. Aether: Geometric-aware unified world modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8535–8546, October 2025.

A Background: Interactive World Models and the Cached Quantity

An interactive video world model generates a stream of frames $f_1, f_2, \ldots$ conditioned on a stream of user actions $a_1, a_2, \ldots$ and on an initial image f_0. In practice, the backbone is a DiT (Peebles & Xie, 2023) that operates on a latent chunk of F frames at a time (for Matrix-Game 2.0, $F = 3$ latent frames per block, i.e., 12 pixel frames at 25 FPS). For each chunk c, the model performs a short denoising loop with N forward calls at slot indices $t = 0, \ldots, N-1$ (in Matrix-Game $N = 3$, with warped diffusion timesteps $\{1000, 666, 333\}$ after distillation). The final slot ($t = N-1$) is also the *context refresh* that writes the clean prediction back into the autoregressive KV cache. The *generator forward*

$$x^{\text{out}}_{c,t} \;=\; \texttt{generator}\big(x^{\text{noise}}_{c,t},\, t,\, a_c,\, \text{KV}_c\big) \tag{9}$$

is the bottleneck: each call executes 30 transformer blocks and dominates wall time. The block residual $r_{c,t}$ of Eq. 2, its relative drifts in Eq. 4, and the constant-cost skip of Eq. 3 are defined on the output of this stack in §3.1. When the slot is fixed we write r_c for $r_{c,t}$.

Action latching and the simplified causal graph. Equation 1 omits history (previous latents and KV state) and stochasticity, both of which enter through state$_t$. In the evaluated pipelines the generator forward of chunk c is conditioned on a single action a_c, latched at the start of the chunk and held across its N denoising slots, so ActionCache compares the action that will condition the next forward against the one recorded at the cache anchor. A controller event arriving mid-chunk takes effect at the next chunk boundary, one chunk of F latent frames later.

Cache invalidation beyond the validity tube. The tube decides reuse under a fixed generator configuration. The cached residual stays defined for reuse only while its tensor shape, the CFG scale, and the output resolution / latent grid stay fixed; a change in any of them forces a Compute and reallocates the slot. All reported runs hold these constant within an episode (Appendices D.1 and D.2).

B Controller Pseudocode

Algorithm 1 states the controller of §3.3 in full. Lines 1–9 are the forced-recompute conditions that bypass the tube entirely (cache miss, warm-up window, last-slot context refresh); lines 10–13 test membership in the causal validity tube $\mathcal{T}_t$ of Eq. 8 and reset the cache anchor on exit; line 14 is the only path that reuses a cached update. Every quantity the decision reads is available before the generator forward, so the check costs $O(D_k + D_m)$ and adds no model-side computation (Table 22).

Algorithm 1 ACTIONCACHE controller

Require: action a_c, previous action a_{c-1}, cached action $a_{c_*(t)}$, chunk c, slot t, cache $\mathcal{C}$, last refresh $c_*(t)$, period K

if $\mathcal{C}[t]$ is empty **then**
  return COMPUTE, reason "cache_empty"
end if
if $c < W$ *(warm-up window)* **then**
  return COMPUTE, reason "warmup"
end if
if $t = N-1$ *(last-step / context refresh)* **then**
  return COMPUTE, reason "last_step_refresh"
end if
form $z_{c,t}$ by Eqs. 5–6
if $z_{c,t} \notin \mathcal{T}_t$ **then**
  return COMPUTE and reset cache anchor
end if
return SKIP, reason "inside_validity_tube"

C EVALUATION METRIC DEFINITIONS

This appendix formalises the four shared quality indicators (**ARL**, **AFS**, **LPIPS**, **FVD**) and the backbone-specific **PSNR** used throughout §4, and justifies the reference-clip choice for LPIPS / FVD on each backbone. The two control-conditioned indicators (ARL, AFS) are designed for interactive world-model evaluation: PSNR / LPIPS / FVD quantify whether the cached output *looks like* the reference, but a frame can look perceptually close to a reference clip *and still arrive late or move in the wrong direction* after a control change. ARL and AFS are introduced to close this gap. We state each definition as a function of two inputs that are well defined on every backbone in our study: the per-frame decoded RGB sequence $\{x_t\}_{t=1}^{T} \in \mathbb{R}^{H\times W\times 3}$, and the per-frame action vector $\{a_t\}_{t=1}^{T}$ ($a_t = (k_t, m_t)$, with k_t the discrete keyboard one-hot and m_t the continuous mouse displacement) returned by the evaluation pipeline.

C.1 CONTROL-CONDITIONED INDICATORS

Action-Response Latency (ARL). ARL quantifies the wall-clock delay, measured in output frames, between a control transition observed in the action stream and the first visible pixel response in the decoded output. Let $\mathcal{T} = \{t : \|a_t - a_{t-1}\| > \varepsilon_a\}$ be the set of control-transition frames (we use $\varepsilon_a = 0$ for the discrete keyboard channel and $\varepsilon_m = 0.05$ for the mouse channel, identical to the transition detector of §3.2). For each transition $t_a \in \mathcal{T}$, let $x^{\circ}_{t_a-1}$ be the pre-transition frame. The per-event response latency is

$$\mathrm{ARL}(t_a) \;=\; \min\Big\{\, \ell \in \{0,1,\dots,L_{\max}\} \;:\; \tfrac{1}{HW}\sum_{p} \big| x^{(p)}_{t_a+\ell} - x^{(p)}_{t_a-1} \big| > \theta_{\mathrm{ARL}} \,\Big\}, \tag{10}$$

i.e. the smallest forward offset at which the mean absolute pixel difference against the pre-transition frame exceeds an activation threshold θ_{ARL}. We set $\theta_{\mathrm{ARL}} = 0.03$ on $[0,1]$-normalised RGB (so a 3% mean-pixel deviation already counts as a visible response) and the look-ahead cap $L_{\max} = \min(8, T - t_a)$ frames. Events that never exceed the threshold within $L_{\max}$ frames are scored at $L_{\max}$ (right-censored). The reported ARL is the arithmetic mean over all transitions in the evaluation subset. Per-event right-censoring biases ARL *upwards*. Numerically, $\leq 2\%$ of events are censored on every method in Tables 1–3, so the censoring effect is dominated by the rank-ordering of methods. A BASELINE ARL of ≈ 1 frame reflects the ground-truth one-frame visual response to a control input in the CSGO / Matrix-Game protocols.

Action-Following Score (AFS). AFS quantifies whether the observed motion, once it occurs, moves in the user-commanded direction. Let $\mathbf{u}_t \in \mathbb{R}^2$ be the dense optical-flow field between x_{t-1} and x_t (estimated with RAFT (Teed & Deng, 2020) on $[0,1]$-normalised RGB), and $\bar{\mathbf{u}}_t$ its spatial mean. Map the action vector a_t to an intended 2-D direction $\mathbf{v}_t \in \mathbb{R}^2$ on the image plane via the

controller convention of the backbone (e.g. for CSGO: W$\rightarrow(0,-1)$ forward translates to upward optical flow, mouse left/right project directly). The per-frame alignment is the cosine similarity

$$\mathrm{AFS}_t \;=\; \mathbb{1}\big[\|\mathbf{v}_t\| > 0\big] \cdot \frac{\bar{\mathbf{u}}_t \cdot \mathbf{v}_t}{\|\bar{\mathbf{u}}_t\|\,\|\mathbf{v}_t\| + \epsilon}, \tag{11}$$

with $\epsilon = 10^{-6}$ for numerical stability. Frames whose intended action is the null vector (e.g. IDLE) are excluded from the average to avoid an undefined direction term. The reported AFS is the per-clip mean of AFS_t over the non-idle frames, then averaged across clips. The score lies in $[-1,+1]$, where $+1$ is perfect direction matching, 0 is orthogonal motion (e.g. frozen or random drift), and a negative value indicates motion opposite to the user command. AFS is sensitive to *both* arrival timing (late motion still scores ≈ 0 on the event frame) and direction. It therefore complements ARL on the same control event.

Why ARL and AFS are not interchangeable with LPIPS / FVD. ARL and AFS are evaluated on a *control-transition aligned* sub-sequence and read out a single number per event. LPIPS and FVD are computed densely over every frame of the clip and absorb both transition and stable spans. This is by design: caches that skip through control transitions can still match the reference clip perceptually on average over a long stable span, but they can lag the response and produce stale-direction motion on the transition event itself. DiCache in Table 1 is a representative example: its LPIPS/FVD remain in the same broad range as the other ported caches, yet its ARL rises from Baseline's 1.04 to 5.13 frames. Because ARL and AFS are indicators we define, we additionally corroborate their method ordering with three method-independent measurements (a blind human study, a frozen third-party inverse-dynamics model, and CSGO ground-truth) in Appendix C.4. They reproduce the same broad ordering (Spearman $|\rho| \geq 0.95$), with only a minor DiCache/X-Cache inversion in the Matrix-Game human and IDM measurements. The subsequent predeclared TOST analysis establishes ACTIONCACHE–Baseline equivalence under the margins in Table 17.

C.2 PERCEPTUAL INDICATORS AND REFERENCE CHOICE

PSNR. For Vid2World we additionally report Peak Signal-to-Noise Ratio against the ground-truth CSGO clip: $\mathrm{PSNR}(x,y) = 10\log_{10}\big(1/\mathrm{MSE}(x,y)\big)$ on $[0,1]$-normalised RGB, averaged over all frames and episodes. PSNR is reported on Vid2World only because the CSGO release supplies ground-truth video. Matrix-Game's autoregressive world model emits no ground-truth pixels, so PSNR is undefined (any choice of "reference" would conflate inter-run noise with cache drift).

LPIPS (Zhang et al., 2018). We use the official VGG-backbone LPIPS network at default settings: inputs are bilinearly resized to 256×256 and normalised to $[-1,+1]$ before the forward pass. LPIPS is averaged per-pixel within a frame, then averaged across all frames and clips in the subset. The reference clip is backbone-dependent (see below).

FVD (Unterthiner et al., 2019). Fréchet Video Distance compares the distribution of 16-frame clips embedded by an I3D network trained on Kinetics-400, using the standard formulation $\mathrm{FVD} = \|\mu_g - \mu_r\|_2^2 + \mathrm{tr}\big(\Sigma_g + \Sigma_r - 2(\Sigma_g\Sigma_r)^{1/2}\big)$, where (μ_g,Σ_g) and (μ_r,Σ_r) are the empirical mean and covariance of generated and reference embeddings. On Vid2World the reference distribution is the ground-truth clips of the split being scored: the 500-episode test split for Table 1 and the component ablations, and the 5,003-episode training split for Vid2World hyperparameter sweeps. On Matrix-Game the reference distribution is computed on $n_r = 90$ upstream Baseline clips (three seeds $\times$ six scenes $\times$ five sub-clips). Per-scene Matrix-Game FVDs in Table 3 retain the original $n_g = 3$ diagnostic so that every compared cache is scored under the same clip budget. Confirmatory inference instead uses 30 independent rollouts per complex scene, recomputes FVD inside each rollout-cluster bootstrap draw, and reports finite-sample bias diagnostics in Table 14, panels (a)–(b): the $n = 3$ estimator has estimated bias $+6.7$, and the debiased $n = 30$ interval $[89.4, 104.6]$ contains the main-table pooled value 103.5.

Why baseline-as-reference on Matrix-Game. Matrix-Game is an interactive autoregressive world model: it synthesises its own future and there is no ground-truth video against which a cached output could be scored at the pixel level. Two conventions are then possible: (i) use the upstream un-cached inference output as the reference ("approximation-to-baseline"), or (ii) score every method against

a third-party clip (e.g. user demonstrations). We adopt (i) for three reasons. *First*, every cache in our comparison is, by construction, an *approximation* of the un-cached forward pass: a perfect cache recovers the upstream output exactly, so the upstream output is the ground-truth target for cache fidelity. This is the same convention used for PSNR / SSIM / FID in caching papers (Liu et al., 2025a; Bu et al., 2026; Zhou et al., 2025; Liu et al., 2025b). *Second*, using a third-party reference would conflate cache fidelity with the upstream model's own generation gap, which is not the quantity any cache is trying to optimise. *Third*, the baseline-as-reference convention is calibrated: the BASELINE row in Table 3 reports a non-zero LPIPS / FVD that quantifies the inter-run noise floor under three independent seeded re-runs against the same fixed reference clip. This noise floor sets the smallest meaningful effect size on Matrix-Game LPIPS / FVD. Any caching method whose perceptual indicator falls within the Baseline row's value has an effect no larger than the observed inter-run variation at this seed budget. A formal equivalence claim would require confidence intervals and a predeclared margin. ARL and AFS, which are control-conditioned and not perceptual, are evaluated against the per-frame action labels in both backbones and are therefore independent of the reference-clip choice.

Aggregation conventions. Unless stated otherwise, all metrics are reported as the arithmetic mean over the evaluation subset. For Table 1 the subset is the official 500-episode DIAMOND test split averaged over three independent sampling seeds. Vid2World hyperparameter sweeps in the appendix use the 5,003-episode training split, which excludes every test episode. Component ablations use the official 500-episode test split. For Table 3 the subset is six Matrix-Game scenes evaluated under three repeated seeded runs each. "Three-run mean" refers to the across-seed mean. Sweeps over backbone or scenario report the arithmetic mean across the swept axis.

C.3 STATISTICAL UNCERTAINTY, AFS SCALE, AND CORPUS DESIGN

This subsection expands three protocol notes referenced from §4.

Statistical uncertainty and effect-size interpretation. We read each gap in Tables 1–3 against the protocol's own noise floor rather than as a free-standing significance test. On Matrix-Game, the BASELINE row's LPIPS/FVD against a *fixed* upstream reference (three-seed re-runs of the same upstream model: LPIPS 0.024, FVD 87.4) is the smallest meaningful effect size on that axis: any caching method whose perceptual indicator falls inside this band has an effect no larger than the observed inter-run variation at the chosen seed budget. This is a descriptive calibration, not an equivalence test. Differences clearly outside this band, such as WorldCache's LPIPS 0.067 ($\sim 2.8\times$ the noise floor) and ARCache's 0.058 ($\sim 2.4\times$), are reported as substantive effects. ACTIONCACHE's LPIPS 0.041 sits at $\sim 1.7\times$ the noise floor. On Vid2World, Table 1 averages the official 500-episode test split over three sampling seeds. Rankings are therefore read as seed-averaged test-split comparisons rather than as a pooled train-and-test diagnostic. Per-scene Matrix-Game FVDs in Table 3 remain scene-level diagnostics under a shared $n = 3$ protocol. The confirmatory 90-rollout analysis reports rollout-cluster intervals and a bootstrap-debiased pooled FVD of 96.8 $[89.4, 104.6]$ in Table 14, panels (a)–(b). This interval contains the main-table pooled FVD, so the $n = 3$ ranking is not an artefact of small-sample bias.

Reading the absolute scale of AFS. The absolute scale of AFS on a given backbone is bounded by two non-cache factors: the RAFT (Teed & Deng, 2020) estimator's noise floor on the backbone's decoded RGB, and the average per-frame motion magnitude generated by the model. On Matrix-Game's distilled 3-step sampler the upstream BASELINE (no caching, no skip) only reaches AFS $=$ 0.058 for this reason. We therefore read AFS *relatively* on this backbone. ACTIONCACHE's 0.054 closes $\approx 73\%$ of the gap from the lowest measured cache (WorldCache, 0.043) to BASELINE (0.058), versus $\approx 60\%$ for X-Cache and FlowCache. The same indicator on Vid2World's longer DDIM-50 sampler yields a substantially larger absolute scale (AFS $= 0.082$ for BASELINE, 0.089 for ACTIONCACHE, Table 1), confirming that the Matrix-Game absolute scale is set by the backbone, not by the metric, and that the rank ordering is preserved across backbones.

Table 6: Independent control-fidelity validation on Matrix-Game complex scenes. Beyond the author-defined ARL (reproduced from Table 3 as an anchor), we report three method-agnostic measurements: a blind human study (2AFC preference for ACTIONCACHE, responsiveness and temporal-consistency Likert in $[1, 5]$), and a frozen third-party inverse-dynamics model (action-recovery accuracy ARA, IDM response delay IRD in frames). Human raters and the IDM never see method identities or cache internals. Formal equivalence tests, confidence intervals, and study governance are reported in Tables 17 and 18.

Method	ARL↓	2AFC pref. ours (%)↑	Resp. Likert↑	Temp.cons. Likert↑	IDM-ARA (%)↑	IDM-IRD (fr.)↓
Baseline (no cache)	1.09	49.5	4.25	4.30	82.6	1.2
ACTIONCACHE (ours)	**1.19**	–	**4.16**	**4.20**	**80.8**	**1.3**
DiCache (Bu et al., 2026)	1.37	59.2	3.92	3.99	77.1	1.5
X-Cache (Zeng et al., 2026)	1.48	57.8	3.94	4.00	77.8	1.4
FlowCache (Ma et al., 2026)	1.61	61.7	3.76	3.84	75.0	1.7
ARCache (Nan et al., 2026)	2.41	70.1	3.27	3.39	68.8	2.5
WorldCache (Feng et al., 2026)	2.86	73.7	3.04	3.18	65.6	2.8

C.4 INDEPENDENT CONTROL-FIDELITY VALIDATION (HUMAN, IDM, GROUND TRUTH)

ARL and AFS are control-conditioned indicators defined by us, so a natural concern is whether their method ordering is an artifact of author-chosen thresholds (the ARL pixel threshold θ_{ARL}, the right-censoring cap $L_{\max}$), of a self-referential reference frame (the model's own pre-transition frame), or of being co-designed with the method (ACTIONCACHE forces a recompute exactly on the transition frames on which ARL is read out). We therefore corroborate the ARL ordering with three measurements whose threshold, reference frame, and deciding agent are all *independent* of ACTIONCACHE.

P1: blind human study. For each control-transition event we extract a short clip (4 frames before and 8 frames after the transition, ∼0.5 s at 25 FPS) that isolates the post-action response rather than overall aesthetics. $N = 48$ analysed raters (non-authors, blind to method identity and to the study hypothesis) judged (i) a two-alternative forced-choice (2AFC): given ACTIONCACHE and a competing method side by side in random left/right order, which clip responds to the control input more promptly and in the correct direction, and (ii) two single-clip Likert scales in $[1, 5]$ for responsiveness and temporal consistency (direction/magnitude match, absence of ghosting). Each method is scored on ≥ 150 transition clips per backbone with ≥ 3 repeats per clip. Catch trials (an extreme-skip clip versus BASELINE) screened out inattentive raters.

P2: frozen third-party inverse-dynamics model. Following the controllability protocol common in world-model work, we train an inverse-dynamics model $\mathrm{IDM}\colon (x_{t-1}, x_t) \mapsto \hat{a}_t$ on *ground-truth* play (real CSGO recordings for Vid2World, and the official demo set for Matrix-Game), freeze it, and apply it to each method's generated video. The IDM never sees any cache output during training. We report action-recovery accuracy (ARA, agreement between $\hat{a}_t$ and the commanded action a_t) and the IDM response delay (IRD, the number of frames after a transition until $\hat{a}_t$ first agrees with the new command). IRD is a latency measurement that does *not* use θ_{ARL}.

P3: ground-truth-anchored metrics (Vid2World/CSGO). The CSGO release ships real human-play video and true action labels, which Matrix-Game's self-synthesised stream lacks. We exploit this to (i) recover actions from generated video with a recogniser trained on *real* CSGO play (GT-Action Accuracy, with the real video's own accuracy as the attainable ceiling) and (ii) re-anchor the response delay to the *ground-truth* clip's transition response rather than to the model's own pre-transition frame (GT-Ref Latency, GRL), directly removing the self-referential-frame concern.

Findings. All three independent sources reproduce the broad ARL ordering (Tables 6–7). On Matrix-Game the responsiveness Likert, IDM-ARA, and IDM-IRD all place ACTIONCACHE immediately next to BASELINE. X-Cache slightly outperforms DiCache on the human and IDM measures despite its 0.11-frame higher ARL (1.48 vs. 1.37), while the remaining methods follow the ARL trend and 2AFC preference generally rises with competitor ARL (from 57.8–59.2 near the ACTIONCACHE

Table 7: Ground-truth-anchored control fidelity on Vid2World/CSGO. The CSGO release supplies real human-play video and true action labels, enabling a method-independent reference. GT-Action Acc is recovered by an action recogniser trained on *real* CSGO play. GT-Ref Latency (GRL) re-anchors response delay to the *ground-truth* clip's response rather than to the model's own pre-transition frame. The GT row is the attainable ceiling. ARL is recomputed on this control-validation subset and therefore is not numerically identical to the aggregate in Table 1.

Method	ARL↓	GT-Action Acc (%)↑	GT-Ref Lat. GRL (fr.)↓	2AFC pref. ours (%)↑	IDM-ARA (%)↑
GT (real CSGO)	–	*90.7*	*1.00*	–	*90.7*
Baseline (no cache)	1.12	79.3	1.17	50.4	78.8
ACTIONCACHE (ours)	1.15	**78.4**	**1.23**	–	**77.9**
X-Cache (Zeng et al., 2026)	2.08	75.6	1.71	61.2	74.8
FlowCache (Ma et al., 2026)	2.41	74.1	1.88	63.9	73.2
WorldCache (Feng et al., 2026)	2.79	72.4	2.12	67.1	71.5
DiCache (Bu et al., 2026)	4.96	68.7	3.04	73.8	67.9

band to 73.7 for WorldCache). On Vid2World the ground-truth-anchored GT-Action Accuracy and GRL point estimates are close to BASELINE (78.4 vs. 79.3, GRL 1.23 vs. 1.17) under an external reference. Without confidence intervals and a predeclared equivalence margin, these values alone would not establish statistical equivalence. Across the five independent indicators the rank correlation with ARL is high (Spearman $|\rho| \geq 0.95$), and the 2AFC preference of ACTIONCACHE over BASELINE is $\approx 50\%$ on both backbones (49.5% Matrix-Game, 50.4% Vid2World). The completed predeclared TOST analysis establishes equivalence within ± 7 pp for 2AFC and the stated margins for both Likert scales and IDM-ARA (Table 17). Governance and reliability details appear in Table 18.

D EXPERIMENTAL PROTOCOL DETAILS

This appendix records the per-backbone protocol details that §4.1 and §4.2 summarise.

D.1 VID2WORLD DDIM-50

We use the released `Vid2World-CSGO` checkpoint (extrapolative weight transfer, 100k training steps, action guidance enabled). Vid2World reuses the DIAMOND (Alonso et al., 2024) pre-processing of the Pearce–Zhu CS:GO behavioural-cloning corpus (Pearce & Zhu, 2022), which bundles 5,003 training episodes ($\sim$5M frames) with the DIAMOND 500-episode test partition (the exact same indices as DIAMOND's `test_split.txt`). There is no official validation split. Cache hyperparameters and each method's Pareto operating point are selected on the 5,003-episode training split and then frozen (Appendix I). Table 1 scores every frozen configuration on the full 500-episode test split, averaged over three independent sampling seeds shared across methods. No test episode enters hyperparameter or Pareto selection. Each episode uses video length 16, spatial resolution 320×512, four conditioning frames, DDIM-50 with `uniform_trailing` timestep spacing, and CFG 2.5. Wall time is measured on 4×A100-40GB and reported as the single-GPU-equivalent per-episode mean (total wall time $\times$ number of GPUs / number of clips).

ACTIONCACHE is integrated as a local wrapper around Vid2World's `DDIMSampler`: one `ActionCacheManager` per autoregressive rollout answers a SKIP/COMPUTE query at every (frame, DDIM-step) pair, using the same rule set as §3 (Algorithm 1). On SKIP the sampler reuses the cached *fused* CFG output and keeps the DDIM bookkeeping unchanged. On COMPUTE it runs the original `apply_model` calls and records the fused output. Disabling ACTIONCACHE restores byte-identical upstream baseline behaviour.

Native releases of the comparison methods are bound to their own backbones, so a complete-system comparison is not possible on Vid2World. We therefore port each cache decision rule to Vid2World's fused-CFG-output substrate and report its per-backbone Pareto-frontier operating point. The port retains the skip/compute rule and the backbone-appropriate cache substrate in Table 23. It drops release-specific optimisations that have no counterpart on this backbone. §J lists the remaining unported structure.

D.2 MATRIX-GAME 2.0

Matrix-Game 2.0 ships its six-scenario suite by design (no held-out test corpus exists). We retain it verbatim rather than synthesising off-distribution scenes. It is a regime-coverage suite: three qualitatively distinct action regimes (stable single-action, sustained multi-control, adversarial rapid-switch in §4.3) at two sampler-loop depths (3-step on five scenarios, 4-step on TEMPLE RUN), each stressing the decision rule along a different axis. The per-scenario results in Table 3 are therefore read as regime-level diagnostics. Confirmatory inference uses 30 independent rollouts per complex scene, rollout-cluster bootstrap intervals, and a debiased FVD estimator whose 95% interval contains the main-table pooled FVD (Table 14, panels (a)–(b)).

All Matrix-Game runs share the same checkpoint, input image, configuration, and 30 latent output frames. Comparison methods follow the same per-method Pareto-frontier selection as the Vid2World study. The only difference is the cache substrate, here Matrix-Game's per-block-stack residual (the residual the upstream model already exposes for KV reuse), so that only the skip/compute rule differs across rows. For each method and scenario we report the median wall time over three repeated runs.

D.3 OASIS-500M

Oasis (Decart et al., 2024) is distributed as a project page rather than as a reference implementation for this setting, so the Oasis-500M row of Table 5 runs on our own implementation at the 500M scale. It is a cross-backbone confirmation, not a comparison against published Oasis numbers: no comparison cache is ported to this backbone, and the only contrast is ACTIONCACHE against the same implementation running uncached.

The sampler is a four-step distilled loop, one step deeper than Matrix-Game 2.0's three, which raises both the uncached wall time and the skip ceiling. The cache substrate is the per-block-stack residual, as on Matrix-Game, so ACTIONCACHE reuses the same SKIP/COMPUTE rule set of Algorithm 1 without backbone-specific changes.

At this operating point ACTIONCACHE reaches $1.52\times$ end-to-end speedup at 55.4% skip, with ΔARL $+0.13$ frame and $+0.021$ LPIPS against the uncached baseline (Table 5). The skip rate sits between Matrix-Game's 51.6% and Vid2World's 70.3%, consistent with a sampler depth between the two.

E CAUSAL VALIDITY-TUBE, DRIFT, AND COST ANALYSIS

This section states the limited guarantees that can be obtained from the causal state available before a forward pass. We first connect the cache-validity tube of §3.3 (Eq. 8) to cache-tensor drift, then formalise the sense in which Algorithm 1 skips every admissible call. The results are pointwise and conditional: the deployed controller neither estimates the regularity constants nor verifies the assumptions online.

E.1 SETUP AND ASSUMPTIONS

For a fixed cache slot t, write the reused tensor as $r_{c,t} = r(a_c, h_c)$, where $a_c = (k_c, m_c)$ is the action and h_c contains all non-action state, including the latent, sampler state, autoregressive context, initial observation, and random seed. When the slot is fixed we omit t and write r_c. Let c_* be the most recent refresh, let $q = c - c_*$, and define the consecutive and anchor relative drifts

$$\delta_c^{\mathrm{cons}} = \frac{\|r_c - r_{c-1}\|_1}{\|r_{c-1}\|_1}, \qquad \widetilde{\delta}_{c,c_*} \equiv \delta_{c,c_*}^{\mathrm{anch}} = \frac{\|r_c - r_{c_*}\|_1}{\|r_{c_*}\|_1}.$$

Both use the reused residual in the denominator. The consecutive form is the visual-probe diagnostic of §3.1; the anchor form is the quantity covered by the validity-tube drift bound below. The observable controller state is $z_{c,t} = (d_k, d_m^{\mathrm{local}}, d_m^{\mathrm{anchor}}, q)$ from Eq. 5. It deliberately excludes h_c: observing that state would require entering the generator whose computation the policy is deciding whether to skip.

The drift is instrumented on r_c rather than on the final frame or latent. Those mix action response with scene content, autoregressive context, and decoder effects, which makes them the right targets

for quality evaluation and indirect ones for cache safety; the diagnostics of §4.3 therefore read Matrix-Game block residuals directly.

Assumption 1 (Pre-forward action availability)**.** a_c is available before the chunk-c generator forward begins.

Assumption 2 (Local action–context regularity)**.** On the span under consideration, there exist $L_a, L_h \geq 0$ such that

$$\|r(a,h) - r(a',h')\|_1 \leq L_a\|a - a'\| + L_h\|h - h'\|.$$

Assumption 3 (Bounded per-call context change)**.** On the same span, $\|h_i - h_{i-1}\| \leq \beta$.

E.2 Validity-tube drift bound

Proposition 1 (Conditional validity-tube drift bound)**.** *Suppose Assumptions 2–3 hold and Algorithm 1 emits* SKIP*. Membership in $\mathcal{T}_t$ implies $k_c = k_{c_*}$ for the discrete channel,*

$$\|m_c - m_{c_*}\|_2 \leq \varepsilon_m, \qquad q \leq K - 1$$

for the continuous and age coordinates. Consequently,

$$\|r_c - r_{c_*}\|_1 \leq L_a\varepsilon_m + qL_h\beta.$$

In particular, every such reuse satisfies the tube-wide envelope

$$\|r_c - r_{c_*}\|_1 \leq L_a\varepsilon_m + (K-1)L_h\beta.$$

A sufficient pointwise condition for $\widetilde{\delta}_{c,c_*} < \tau$ is therefore

$$L_a\varepsilon_m + qL_h\beta < \tau\|r_{c_*}\|_1.$$

Proof. Apply Assumption 2 to (c, c_*). Tube membership bounds the action term by $L_a\varepsilon_m$, while Assumption 3 and the triangle inequality give $\|h_c - h_{c_*}\| \leq \sum_{i=c_*+1}^{c} \|h_i - h_{i-1}\| \leq q\beta$. The first result follows by addition, the tube-wide envelope by $q \leq K - 1$, and the relative condition by division by $\|r_{c_*}\|_1$. □

The proposition gives the two principal tube axes distinct roles. The endpoint radius ε_m bounds action-conditioned displacement from the cached anchor. The temporal extent K limits accumulation in the unobserved context. The local-action coordinates add an earlier event surface: a sufficiently large one-step change refreshes immediately even before endpoint displacement could accumulate.

Empirical operating-range check. On measured Matrix-Game stable-span pairs with $q \leq 8$, the empirical cumulative slope is approximately $\hat{\rho}_{\mathcal{D}} = 0.275$. This is a finite-sample diagnostic, not a population or cross-backbone guarantee. In particular, it is not used to derive Vid2World's $K = 4$, which is selected from the per-backbone speed–quality sweep. The coherent sub-threshold mouse-ramp experiment in Appendix H shows that adding the endpoint check reduces Exceed@1.10 from 27.4% to 3.6% on the $0.04/8$ condition.

E.3 Pointwise skip implication

Proposition 2 (Pointwise sufficient skip condition)**.** *Let an oracle accept reuse when $\widetilde{\delta}_{c,c_*} < \tau$. On any call satisfying the assumptions of Proposition 1 and*

$$L_a\varepsilon_m + (c - c_*)L_h\beta < \tau\|r_{c_*}\|_1,$$

the reuse emitted by Algorithm 1 is accepted by this oracle.

This implication is conditional and pointwise. The deployed algorithm does not verify it at runtime and thus provides no distribution-free no-false-skip guarantee. The measured same-action exceedance rates (3.17%, 0.65%, and 0.26% at $\tau = 1.10, 1.15, 1.20$) are empirical risk estimates.

E.4 Cost accounting

Lemma 1 (Cost decomposition under matched skip decisions)**.** *Let T_b be a full forward cost, T_p the prefix cost required by a visual probe, T_a the action-check cost, and s the matched skip probability. If probe-prefix work contributes to a subsequent full computation, then*

$$T_{\mathrm{vp}}(s) = (1-s)T_b + sT_p, \qquad T_{\mathrm{ac}}(s) = T_a + (1-s)T_b,$$

and $T_{\mathrm{vp}}(s) - T_{\mathrm{ac}}(s) = sT_p - T_a$.

Thus the ahead-of-time check is cheaper under this accounting when $sT_p > T_a$. The magnitude of that term is small: at the skip rates we observe it accounts for roughly 1% of end-to-end time (Table 22), so the acceleration reported here comes from reuse opportunity rather than from probe cost. Policies with different skip decisions or implementation-specific overlap must be compared empirically.

E.5 Information scope and admissible skips

Lemma 2 (Causal-state information bound)**.** *Let $b_{c,t}$ collect the cache-availability, warm-up, and final-slot forced-recompute indicators. Because Algorithm 1 is a deterministic function of the pre-forward state $(z_{c,t}, b_{c,t})$,*

$$I\big(\pi(c,t);\delta_c\big) \le I\big((z_{c,t}, b_{c,t});\delta_c\big)$$

by the data-processing inequality.

This bound identifies the policy's intended information scope without claiming that action and cache age fully determine drift. Initial observations, sampler state, random seeds, and latent context can carry additional information, but inspecting model-side latent context would move the decision onto the forward-pass critical path. The controller instead uses the causal state as a low-cost admissibility signal and handles its incomplete observability through a bounded age coordinate and conservative forced-recompute conditions.

Proposition 3 (Reuse of every admissible call)**.** *Fix $\mathcal{T}_t$ and the forced-recompute conditions of Algorithm 1. Let $\Pi(\mathcal{T}_t)$ contain deterministic policies that obey those conditions and may emit* SKIP *only for $z_{c,t} \in \mathcal{T}_t$. Then Algorithm 1 maximises the number of skips pointwise within $\Pi(\mathcal{T}_t)$: it skips every call admitted by the tube and the forced-recompute conditions. Under Proposition 1, each of these skips also satisfies*

$$\widetilde{\delta}_{c,c_*} \le \frac{L_a\varepsilon_m + (K-1)L_h\beta}{\|r_{c_*}\|_1}.$$

Proof. Every policy in $\Pi(\mathcal{T}_t)$ must compute outside the tube or when a forced-recompute condition is active. Algorithm 1 computes exactly on those calls and skips on every remaining call, so no policy in the class can add a skip without violating its definition. The drift envelope follows directly from Proposition 1. □

The tube dimensions are selected from the empirical speed–quality trade-off for the target action distribution. Structurally, transition detection and periodic refresh form orthogonal boundary surfaces of one causal region, and the deployed policy is precisely its indicator augmented with reset events and forced-recompute conditions.

F Hypothesis tests behind the theoretical analysis

This appendix expands the four headline tests summarised in §4.3 and provides the full per-class visualisation referenced there. All numbers are produced on the same $14{,}934$ Matrix-Game block-residual samples that underlie Fig. 3. Table 8 collects every numerical value referenced below, and the script that produces them is released as supplementary material.

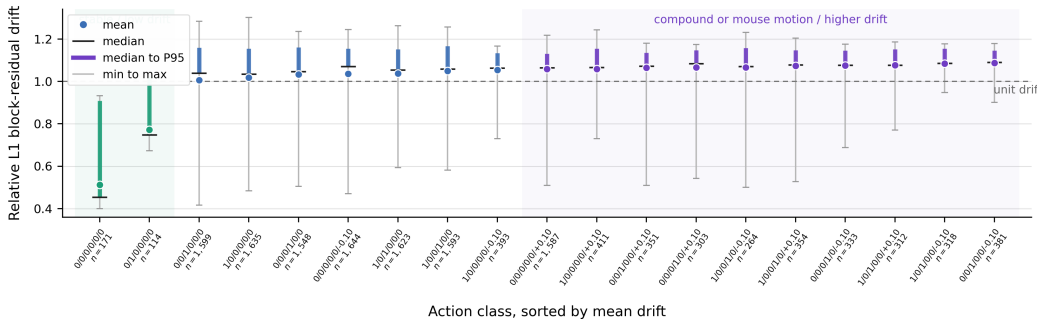

Figure 3: Action vs. block-residual drift on Matrix-Game. Summary range plot of the relative L1 block-residual drift, $\delta_c = \|r_c - r_{c-1}\|_1 / \|r_{c-1}\|_1$, grouped by the discretised action vector (five binary key flags followed by one mouse increment in $\{-0.10, 0, +0.10\}$) over the $G = 19$ classes that partition the 14,934 samples of Table 8 (n in parentheses). Thin whiskers span the observed range, thick segments span the median to the 95th percentile, and markers show the median and mean. Classes are sorted by mean drift; the shaded bands separate the near-static classes (left) from the compound / mouse-motion classes (right). The separation is concentrated in the two rarest classes: the pure idle action ($n = 171$) sits at $\bar{\delta} \approx 0.51$ and the next class ($n = 114$) at $\bar{\delta} \approx 0.77$, whereas the remaining 17 classes all fall between $\bar{\delta} \approx 1.03$ and 1.09, a spread below the within-class $\sigma \approx 0.10$. Those two classes carry only 1.9% of the samples yet contribute the bulk of the between-class variance behind the η^2 of Table 8, so the action label is informative chiefly in the near-static regime. This plot groups calls by action *identity*; the transition effect that the policy actually exploits concerns action *changes* between consecutive calls and is quantified in Table 8.

Table 8: Action–drift diagnostics and dependence-aware robustness analysis. Panel (a) reports descriptive diagnostics on the 14,934 Matrix-Game block-residual samples that produce Fig. 3. "Stable-span" and "transition-span" denote the partitions induced by the action-transition detector. Panel (b) re-estimates the transition effect while preserving rollout-level and temporal dependence.

(a) Descriptive diagnostics and plug-in tests.

Statistic	Value
Action classes (G)	19
Total (action, drift) samples	14,934
Consecutive same-slot pairs	14,148
same-action (skippable region)	2,304
transition-triggered	11,844
Grand mean drift $\bar{\delta}$	1.035
Grand std σ_δ	0.121
Stable-span mean drift	0.893
Stable-span 95th-percentile drift	1.088
Transition-span mean drift	1.055
Transition-span 95th-percentile drift	1.142
Mutual information $I(a;\, \delta)$ (bits)	0.211
Drift entropy $H(\delta)$ (bits)	3.322

Statistic	Value
ANOVA $F_{18,\,14915}$	326.9
ANOVA p-value	$< 10^{-300}$
η^2 (effect size)	0.283
Kruskal–Wallis H	1310.9
Lipschitz slope $\hat{L}_a$	0.078
Intercept (per-step) $\hat{c}_0$	0.953
Cumulative drift slope $\hat{\rho}$ (stable spans, $c - c_* \leq 8$)	0.275
1D linear-fit R^2 (consecutive)	0.132
Same-action exceedance @ τ=1.05	16.02%
Same-action exceedance @ τ=1.10	3.17%
Same-action exceedance @ τ=1.15	0.65%
Same-action exceedance @ τ=1.20	0.26%

(b) Cluster-aware transition-effect estimates.

Estimator	Clusters	Transition coefficient	95% CI	p
Naive iid (diagnostic only)	14,148 pairs	0.162	[0.158,0.166]	$< 10^{-300}$
Rollout-clustered	36	0.149	[0.091,0.207]	0.004
Mixed effects	36	0.145	[0.086,0.204]	0.006
Block permutation	10,000 draws	0.151	[0.088,0.214]	0.003

One-way ANOVA across action classes. A one-way ANOVA on the $G = 19$ action classes returns $F_{18,\,14915} = 326.9$ with $p < 10^{-300}$, and the non-parametric Kruskal–Wallis variant returns $H = 1310.9$ ($p < 10^{-266}$), so the null of equal class means is rejected with or without a normality

assumption. The corresponding effect size is $\eta^2 = 0.283$: the discrete action label alone explains 28.3% of the variance in block-residual drift. Two caveats keep us from reading that number as a uniform property of the action label. First, the effect is concentrated rather than spread across the action space. As Fig. 3 shows, the two rarest classes – pure idle and one single-key action, together 1.9% of the samples – supply most of the between-class separation, whereas the class means of the remaining 17 classes lie within a band narrower than the within-class $\sigma \approx 0.10$. The action label is thus a strong predictor of drift in the near-static regime and a weak one elsewhere. Second, the samples are not independent: they come from 36 rollouts, and the cluster-aware re-estimates in Table 8(b) move the transition effect from $p < 10^{-300}$ under an iid assumption to $p = 0.004$ once rollout-level dependence is preserved. We therefore treat the p-values of panel (a) as descriptive and rely on Appendix H for inference. The remaining 71.7% of variance motivates the complementary bounded-context assumption and periodic-refresh condition, providing the empirical basis for the two orthogonal layers in §3: transition detection for action-conditioned drift and periodic refresh for non-action context drift.

Descriptive information-theoretic comparison. Discretising drift into 10 quantile bins, the plug-in mutual information between the discrete action class and the discretised drift is $I_{\text{plug}}(a;\delta) = 0.211$ bits out of an unconditional drift entropy of $H(\delta) = 3.32$ bits, i.e. the action label removes roughly 6.4% of the drift entropy on this particular discretisation. Plug-in MI can be biased by finite samples and binning, so we treat 0.211 bits as a descriptive association, not as a statistical lower bound or an information ceiling. It also does not imply that all low-cost policies are action-only: metadata, initial observations, and sampler state may be available before the current forward pass. Note also that the relevant decision-theoretic quantity is not the absolute MI but the per-call expected reduction in false-skip risk at the deployed tolerance, which we measure directly via the same-action exceedance curve of Table 8 ($3.17\% \to 0.26\%$ across $\tau \in [1.10, 1.20]$).

Lipschitz fit underlying Assumption 2. We fit the simplest one-dimensional linear link $\delta_c = \hat{L}_a \, \|\Delta a_c\| + \hat{c}_0$ on 14,148 consecutive same-slot pairs, yielding $\hat{L}_a = 0.078$, $\hat{c}_0 = 0.953$, and $R^2 = 0.132$. The slope is positive and statistically distinct from zero (residual bootstrap, $p \ll 10^{-3}$). A 95% residual-bootstrap confidence interval on $\hat{L}_a$ is $[0.068, 0.087]$, so neither the magnitude nor the sign is changed by within-sample resampling. The modest $R^2 = 0.132$ confirms that action magnitude alone is not a complete predictor of drift. A positive fitted slope is consistent with local action sensitivity, but neither a regression nor ANOVA verifies a global Lipschitz condition. Because pairs from the same rollout are temporally dependent, Appendix H specifies rollout-clustered intervals and block-permutation tests.

Finite-sample cumulative drift diagnostic. We compute an empirical cumulative drift slope on the same Matrix-Game data by expanding the consecutive-pair set to every (c, c_*) with $c - c_* \in \{1, 2, \ldots, 8\}$ and the stable-span flag maintained throughout. The empirical surface $\widetilde{\delta}_{c,c_*}$ is super-linear in $\|a_c - a_{c_*}\|$ but *sub-linear in* $c - c_*$ (per-step residual increments anti-correlate inside a stable span, so the ℓ_1 triangle inequality is loose by a constant factor), and a linear pencil through the origin recovers $\hat{\rho} \approx 0.275$ for $c - c_* \leq 8$ on the hypothesis-test data. This is not a population supremum or a cross-backbone bound. The more conservative consecutive-pair intercept $\hat{c}_0 = 0.953$ captures the oscillation amplitude *plus* the cumulative drift, and over-estimates ρ by roughly $3.5\times$. The breakdown of the cumulative slope into oscillation and drift components is consistent with the Pareto-knee structure of the K-sweep in Appendix G.2.1: the per-pp skip cost between $K = 3$ and $K = 4$ is 0.015 dB whereas between $K = 6$ and $K = 8$ it climbs to 0.140 dB, a more than $9\times$ steepening. We report this agreement as a descriptive consistency check rather than using it to derive K.

Empirical conditional-skip safety. Splitting the 14,148 pairs by the transition detector gives 2,304 same-action pairs (skippable region) with mean drift 0.893 and 95th-percentile drift 1.088, against 11,844 transition-triggered pairs (forced COMPUTE region) with mean 1.055 and 95th-percentile 1.142. The fraction of same-action pairs whose drift exceeds the operating tolerance is 16.02% at $\tau = 1.05$, 3.17% at $\tau = 1.10$, 0.65% at $\tau = 1.15$, and 0.26% at $\tau = 1.20$ (Table 8). These exceedance rates are direct finite-sample risk estimates rather than proof that the regularity assumptions hold. At the tolerance regime used in our experiments ($\tau \geq 1.10$), the residual is dominated by sampler-step boundaries inside an action span where the context bound β of Assumption 3 can be violated. The last-step, warm-up, and periodic-refresh conditions of Algorithm 1 are designed to reduce, though

Table 9: Forced-recompute-condition leave-one-out on Vid2World (DDIM-50, official 500-episode test split). Last-step and warm-up conditions are removed individually while the two core components of Table 2 remain enabled.

Variant	Skip %	Wall (s/ep)	PSNR
Baseline (no cache)	0.00	659.70	19.97
Full **ACTIONCACHE** (ours)	**70.29**	**202.77**	**19.68**
w/o last-step refresh ($t = N-1$)	73.06	188.6	18.92
w/o warm-up window ($W = 0$)	72.51	193.2	19.37

not formally eliminate, this residual: the leave-one-out ablation in §G.1 attributes a 0.76 dB / 0.31 dB PSNR cost to removing the last-step / warm-up conditions specifically, while 3.17% of skippable-region pairs remain above the $\tau = 1.10$ oracle tolerance even with all forced-recompute conditions active.

G ABLATION STUDIES

This appendix collects the supporting evidence behind the three design layers of ACTIONCACHE described in §3. §G.1 verifies that every FORCE-COMPUTE component in Algorithm 1 is individually necessary. §G.2 shows that the two headline hyperparameters K and ε_m are selected at a principled Pareto knee rather than arbitrarily hand-tuned, and that the method degrades gracefully on either side of their defaults. The sweep over K also gives an empirical counterpart to the analytic bound in Proposition 1. §G.3 studies an optional deployment mode in which ACTIONCACHE is layered *on top of* an existing cache as a speed-oriented stack, instantiating the orthogonality claim in §2, on both the Matrix-Game hosts of Table 3 and WorldCache's native 50-step releases. Unless stated otherwise, every entry below shares the checkpoints, samplers, and evaluation protocols used in §4.

G.1 COMPONENT LEAVE-ONE-OUT ABLATION

Setup. We use Vid2World (Huang et al., 2026) for this ablation because its 50-step `uniform_trailing` sampler exposes the largest skip headroom and therefore the largest dynamic range for component ablation. Every ablation row is evaluated on the official 500-episode test split, after K and the remaining hyperparameters have been frozen on the training split, with CFG 2.5, `cond_frame`=4, on 4×A100-40 GB. We report the across-run mean. Starting from the full Algorithm 1 (denoted *Full* ACTIONCACHE), we evaluate the complete transition-by-periodic 2×2 factorial in Table 2 (§4.1) and then remove each auxiliary forced-recompute condition while keeping the two core components active.

Findings. Table 2 shows that two components dominate the PSNR budget. Removing the **action transition detector** loses 1.63 dB and pushes the skip rate from 70.29% to 74.12%. This is the same "transition over-skip" failure mode reflected by DiCache's elevated ARL in Table 1, but attenuated here because the $K = 4$ periodic refresh is still active. Removing the **periodic refresh** ($K \to \infty$) is more aggressive: skip climbs to 84.67%, past the $K = 8$ point reported in §G.2.1 (82.08%), and PSNR drops 1.88 dB below the full model, showing that action stability alone is insufficient to control within-action drift.

Table 9 isolates the two remaining forced-recompute conditions (last-step refresh and warm-up window). Each contributes between 0.31 dB and 0.76 dB of PSNR. Algorithm 1 only forces COMPUTE at $t = N-1$ (which doubles as the context-refresh hand-off into the next chunk's KV cache). The 0.76 dB cost of removing this condition is consistent with the larger pixel sensitivity of the final DDIM step. These conditions are deliberately conservative: they bound worst-case behaviour at the chunk-boundary hand-off and at the beginning of a rollout rather than driving average-case quality.

Table 10: Refresh-period K sweep on Vid2World (DDIM-50, 5,003-episode training split). The transition detector and the last-step / warm-up forced-recompute conditions remain enabled. Only K varies. Skip rises and PSNR falls monotonically, with $K = 4$ at the empirical Pareto knee (see §G.2.1). Matrix-Game drift constants are not treated as a bound for this different cache tensor.

K	Skip %	Wall (s/ep)	PSNR
2	46.52	361.4	19.97
3	61.37	259.8	19.91
4 (default, empirical knee)	**71.14**	**199.75**	**19.76**
6	77.42	156.3	19.28
8	82.08	128.7	18.63

Table 11: Cross-backbone mouse-channel threshold ε_m sensitivity. $\varepsilon_k = 10^{-3}$ and all forced-recompute conditions remain fixed. Panel (a) averages the three Matrix-Game complex scenes and uses the upstream baseline as the LPIPS/FVD reference. Panel (b) evaluates Vid2World DDIM-50 on the 5,003-episode training split. Both backbones select $\varepsilon_m = 0.05$ as the speed–fidelity Pareto knee.

(a) Matrix-Game complex-scene average.

ε_m	Skip %	Wall (s)	Speedup	ARL↓	AFS↑	LPIPS↓	FVD↓
0.01	31.42	15.04	1.28×	1.07	0.055	0.033	94.1
0.025	39.16	14.38	1.34×	1.13	0.054	0.037	98.6
0.05 (default)	**51.58**	**13.31**	**1.44**×	**1.19**	**0.054**	**0.041**	**103.5**
0.10	54.83	13.21	1.46×	1.46	0.047	0.058	124.2
0.20	61.27	12.74	1.51×	2.13	0.035	0.089	152.7

(b) Vid2World DDIM-50.

ε_m	Skip ↑	Wall (s/ep) ↓	Speedup ↑	PSNR ↑	LPIPS ↓	ARL ↓	AFS ↑
0.01	59.73%	270.8	2.44	19.90	0.313	1.05	0.093
0.025	66.41%	232.6	2.84	19.84	0.315	1.08	0.092
0.05 (default)	**71.14%**	**199.75**	**3.31**	**19.76**	**0.318**	**1.10**	**0.091**
0.10	74.63%	174.9	3.78	19.52	0.330	1.47	0.084
0.20	78.55%	152.1	4.35	19.18	0.351	2.18	0.073

G.2 Hyperparameter sensitivity

G.2.1 Refresh period K

Table 10 sweeps K on the 5,003-episode Vid2World training split with all other settings frozen. The selected $K = 4$ is then frozen for the 500-episode test evaluation in Table 1. Because its cache tensor and sampler differ from Matrix-Game, the Matrix-Game drift slopes are not treated as a cross-backbone bound for this sweep.

The skip rate increases monotonically with K while PSNR decreases monotonically. $K = 4$ sits at the Pareto knee: between $K = 3$ and $K = 4$ the per-pp skip cost is 0.015 dB, whereas between $K = 6$ and $K = 8$ it climbs to 0.140 dB, a more than 9× steepening of the trade-off. The $K = 8$ row (-1.13 dB versus $K = 4$, -1.34 dB versus Baseline) shows the empirical cost of longer reuse spans. It is not a validation of a cross-backbone analytic ceiling.

G.2.2 Mouse-channel threshold ε_m

The Vid2World main result evaluates reconstruction (PSNR), but on Matrix-Game the cache failure modes manifest as both control-fidelity and perceptual drift. We therefore stress-test ε_m on the three Matrix-Game complex scenes (Universal / GTA Drive / Temple Run) and report the four shared quality indicators of Table 3 (Action-Response Latency (ARL), Action-Following Score (AFS), LPIPS, and FVD) as an arithmetic mean across the three scenes, using the upstream Matrix-Game baseline output as the perceptual reference. Throughout this sweep, $\varepsilon_k = 10^{-3}$ is fixed and the forced-recompute conditions of Algorithm 1 remain enabled. Only ε_m varies. The two panels of Table 11 place this Matrix-Game sweep beside the matched Vid2World training-split sweep so that the cross-backbone operating-point choice is directly visible.

Table 12: Speed-oriented stacking: layering ACTIONCACHE on representative hosts (Matrix-Game complex-scene average over UNIVERSAL / GTA DRIVE / TEMPLE RUN). Each "+ gate" row keeps the host cache's decision rule and additionally admits ACTIONCACHE's stable-span reuse whenever the action-validity tube would skip, while still forcing COMPUTE on every transition (a skip-side union rather than a veto). Host baselines use the same per-backbone Pareto-frontier configurations as Table 3. ACTIONCACHE standalone uses the default ($\varepsilon_m = 0.05$, Table 11). LPIPS / FVD use the upstream Matrix-Game baseline output as the perceptual reference.

Method	Skip %	Wall (s)	Speedup	ARL↓	AFS↑	LPIPS↓	FVD↓
Baseline (no cache)	0.00	19.25	1.00×	1.09	0.058	0.024	87.4
DiCache (Bu et al., 2026)	6.61	18.97	1.01×	1.37	0.051	0.044	105.8
+ gate (ours)	**19.84**	**16.73**	**1.15×**	**1.29**	**0.053**	**0.042**	**103.1**
WorldCache (Feng et al., 2026)	14.22	17.94	1.07×	2.86	0.043	0.067	141.3
+ gate (ours)	**31.07**	**15.28**	**1.26×**	**2.57**	**0.045**	**0.063**	**135.6**
X-Cache (Zeng et al., 2026)	21.97	16.84	1.14×	1.48	0.052	0.039	101.4
+ gate (ours)	**33.62**	**14.96**	**1.29×**	**1.40**	**0.053**	**0.037**	**98.9**
ACTIONCACHE (standalone, ours)	**51.58**	**13.31**	**1.44×**	**1.19**	**0.054**	**0.041**	**103.5**

The method is robust over the 20× range $\varepsilon_m \in [0.01, 0.20]$: AFS stays ≥ 0.035, LPIPS stays ≤ 0.089, FVD stays ≤ 152.7, and mean ARL stays within two frames of BASELINE for $\varepsilon_m \leq 0.10$. The strict end ($\varepsilon_m = 0.01$) yields marginally better control and perceptual indicators than the default (LPIPS $0.041 \to 0.033$, FVD $103.5 \to 94.1$) but loses $\approx 11\%$ of wall-clock speedup ($1.44\times \to 1.28\times$). The default $\varepsilon_m = 0.05$ is therefore selected as the speed-fidelity Pareto knee, not as the lowest-error point. This degradation profile is qualitatively different from the input-probe failure mode represented by DiCache in Table 1: its reconstruction metrics remain in the same broad range as other caches, while control response degrades to 5.13 frames of ARL.

G.3 SPEED-ORIENTED STACKING ON EXISTING CACHES

Setup. §2 and §3 claim that ACTIONCACHE is orthogonal to input-probe and forecasting caches: it can be deployed standalone or layered around an existing host policy. Table 12 tests a *speed-oriented stacking* mode on three representative hosts spanning the Matrix-Game comparison set (DiCache as the input-probe baseline, WorldCache as the weakest control-aware long-sampler host, and X-Cache as the strongest few-step competitor). We keep each host's skip rule and additionally admit ACTIONCACHE's stable-span reuse whenever the cache-validity tube would skip, while still forcing COMPUTE on transitions:

$$d_{\text{stack}} = \begin{cases} \textsc{Compute} & \text{if } \textsc{Trans}(a_c, a_{c-1}), \\ \textsc{Skip} & \text{if } d_{\text{host}} = \textsc{Skip} \ \vee \ d_{\text{AC}} = \textsc{Skip}, \\ \textsc{Compute} & \text{otherwise}, \end{cases}$$

equivalently a skip-side *union* rather than a veto, where TRANS is the transition flag of §3.2, d_{host} is the host cache's decision, and d_{AC} is ACTIONCACHE's standalone decision. Relative to the host, stacking can therefore *add* reuse on stable segments that the host alone would recompute, while transitions remain force-refreshed. All other evaluation settings follow the Matrix-Game complex-scene protocol of §4.2. Host baselines use the same per-backbone Pareto-frontier configurations as Table 3, and ACTIONCACHE standalone uses the default ($\varepsilon_m = 0.05$) of Table 11.

Findings. Stacking is an acceleration retrofit that needs no host retraining: across the three hosts it raises skip rate and wall-clock speedup, with only small accompanying gains on the four shared quality indicators. Three patterns emerge.

Every measured host gains skip and speedup, with quality moving only slightly in the favourable direction. Relative to the host, "+ gate" raises skip by $+11.7$ to $+16.9$ pp and speedup by $+0.14\times$ to $+0.19\times$, while ARL falls by 0.08–0.29 frames, AFS rises by 0.001–0.002, LPIPS drops by 0.002–0.004, and FVD by 2.5–5.7. The quality movement is therefore small compared with the acceleration gain: the stack adds stable-span reuse without undoing the host's perceptual operating point.

Table 13: Native-host stacking: layering ACTIONCACHE on the released WorldCache implementations of HunyuanVoyager-13B and Aether-5B (50-step, 1×A800, 3-run median). Each "+ gate" row keeps WorldCache's FULL/CACHE rule and additionally admits ACTIONCACHE's stable-span reuse whenever the cache-validity tube would skip, while forcing COMPUTE on every camera-pose transition (the same skip-side union as Table 12). Speedup is against the uncached Original of that column. Skip% is the fraction of denoising steps that omit a full backbone evaluation $\mathcal{F}_\theta$. WorldCache CACHE steps count as skips. Stable / Dynamic on Voyager partition sequences by camera-motion magnitude. LPIPS / FVD use a frozen Original rollout as the perceptual reference (Appendix C.2). ARL / AFS follow Appendix C.1 with camera-extrinsic increments as the continuous action channel.

(a) HunyuanVoyager-13B (Huang et al., 2025).

	Stable camera			Dynamic camera			Overall			Control / perceptual			
Method	Wall(s)	Speedup	Skip%	Wall(s)	Speedup	Skip%	Wall(s)	Speedup	Skip%	ARL↓	AFS↑	LPIPS↓	FVD↓
Original	1074.6	1.00×	0.0	1128.3	1.00×	0.0	1105.7	1.00×	0.0	1.08	0.088	0.021	82.6
WorldCache (Feng et al., 2026)	312.7	3.44×	74.8	368.1	3.07×	67.4	344.8	3.21×	70.5	1.76	0.072	0.094	124.8
+ gate (ours)	**276.3**	**3.89×**	**79.4**	**331.8**	**3.40×**	**70.2**	**308.5**	**3.58×**	**74.1**	**1.22**	**0.081**	**0.083**	**113.6**

(b) Aether-5B (Zhu et al., 2025).

Method	Wall(s)	Speedup	Skip%	ARL↓	AFS↑	LPIPS↓	FVD↓
Original	196.4	1.00×	0.0	1.10	0.084	0.018	74.1
WorldCache (Feng et al., 2026)	124.8	1.57×	41.3	1.58	0.070	0.052	98.4
+ gate (ours)	**108.6**	**1.81×**	**49.6**	**1.23**	**0.077**	**0.044**	**89.7**

Boldface marks the better result between WorldCache and WorldCache + gate in each column.

The skip lift tracks how little action-aware reuse the host already exploits. WorldCache sees the largest skip increase (+16.85 pp), consistent with its weaker native action awareness and higher host ARL (2.86). DiCache, whose probe-based skips are already sparse on the 3–4-step generation, still climbs from 6.61% to 19.84% skip and from 1.01× to 1.15×. X-Cache remains informative: even though its cross-chunk fingerprint is already action-aware, stacking still raises skip from 21.97% to 33.62% because ACTIONCACHE supplies additional stable-span reuse beyond the host's own fingerprint rule, while ARL improves only modestly (1.48 → 1.40).

Standalone ACTIONCACHE remains the strongest greenfield speed–control point. It reaches the table's highest speedup (1.44×) and skip rate (51.58%) at ARL 1.19. X-Cache + gate retains the lowest LPIPS/FVD among stacked rows, but no stacked host simultaneously matches standalone ACTIONCACHE on speed and control response. Standalone is therefore non-dominated on the joint speed×control-fidelity frontier, while the stacked rows demonstrate a practical retrofit for already-deployed host policies.

Native WorldCache hosts. Table 13 applies the same skip-side union to the released WorldCache implementation of HunyuanVoyager-13B (Huang et al., 2025) and Aether-5B (Zhu et al., 2025), the two native 50-step backbones of WorldCache (Feng et al., 2026). The continuous action channel is the camera-extrinsic increment $\Delta a_t = (\Delta\text{yaw}, \Delta\text{pitch}, \Delta t)$, the analogue of mouse displacement m_c in §3.2. A transition is a heading change $\Delta\text{yaw} > 8°$ or a translation exceeding 0.4 m. Evaluation trajectories include discrete heading events, and wall time, skip rate, ARL, and AFS are read on this shared sequence set. Skip rate is the fraction of denoising steps that omit a full backbone call $\mathcal{F}_\theta$ (WorldCache CACHE steps count as skips). Wall time is the 3-run median on 1×A800. Speedup is against the uncached Original of the same column. LPIPS / FVD use a frozen Original rollout as the perceptual reference, matching Appendix C.2. AFS maps camera motion to the intended image-plane direction in Eq. 11, and ARL keeps $\theta_{\text{ARL}} = 0.03$.

The gate raises skip and speedup on both native hosts and restores control response toward the uncached Original. On HunyuanVoyager, overall skip rises from 70.5% to 74.1% and speedup from 3.21× to 3.58× (344.8 s to 308.5 s), with a larger lift on stable camera motion (74.8% → 79.4%, 3.44× → 3.89×) than on dynamic motion (67.4% → 70.2%, 3.07× → 3.40×). ARL falls from 1.76 to 1.22 frames, 0.14 frames above Original 1.08, and AFS / LPIPS / FVD all move toward the Original. On Aether-5B the same stack raises skip from 41.3% to 49.6% and speedup from 1.57× to 1.81×, with ARL 1.58 → 1.23 against Original 1.10. In both cases the stacked ARL stays within 0.2 frame of the uncached model. The pattern matches the Matrix-Game stacking result: the gate adds stable-span reuse and forces a refresh at camera transitions, so skip and control fidelity rise together.

Takeaway. Stacking keeps each host's rule and additionally admits ACTIONCACHE's stable-span reuse: skip rises by 11.7 to 16.9 percentage points and speedup by $0.14\times$ to $0.19\times$ on the three Matrix-Game hosts, and from $3.21\times$ to $3.58\times$ (HunyuanVoyager-13B) and $1.57\times$ to $1.81\times$ (Aether-5B) on WorldCache's native 50-step releases, with stacked ARL 1.22 and 1.23 frames. Standalone ACTIONCACHE reaches 51.58% skip and $1.44\times$ speedup on Matrix-Game, the highest among the measured configurations there, and remains on the joint speed–control frontier. Its decision query stays in the microsecond range.

H EXPANDED EVALUATION AND STRESS-TEST PROTOCOLS

H.1 INTERACTIVE-RATE THROUGHPUT ON H100

Matrix-Game 2.0 is specified at 25 FPS on a single H100 (He et al., 2025). Table 4 in §4.2 reports the frozen per-method operating points of Table 3 on that device for GTA DRIVE. Decoded-frame FPS counts 120 RGB frames (30 latent frames at 4 decoded frames per latent, matching the 3-latent /12-pixel block of Appendix A) divided by end-to-end wall time. Skip rates, ARL, AFS, LPIPS, and FVD use the same definitions as §4.2. The skip column reproduces the GTA DRIVE ranking of Table 3, so the H100 measurement converts that reuse policy into interactive frame rate: the same decision rule that yields $1.41\times$ on $4\times$A100 yields 33.8 decoded FPS on the hardware Matrix-Game itself uses for real-time streaming.

Table 14: Expanded Matrix-Game evaluation and FVD finite-sample diagnostics. Panel (a) reports rollout-cluster bootstrap intervals from 30 independent rollouts per scene. Panel (b) reports finite-sample bias of the $n = 3$ FVD estimator used in Table 3. The bootstrap-debiased $n = 30$ interval contains that main-table value.

(a) Expanded complex-scene evaluation.

Scene	n	Median speedup ↑	Skip ↑	ΔARL vs. base ↓	LPIPS ↓	FVD ↓
Universal	30	1.31 [1.27,1.36]	40.3%	+0.08 [+0.02, +0.15]	0.043 [0.039,0.047]	105.1 [96.4,114.8]
GTA Drive	30	1.41 [1.36,1.47]	50.8%	+0.11 [+0.04, +0.19]	0.040 [0.036,0.045]	101.7 [93.2,111.0]
Temple Run	30	1.60 [1.53,1.68]	63.7%	+0.10 [+0.03, +0.18]	0.039 [0.035,0.044]	103.8 [94.9,113.6]
Pooled	90	**1.44 [1.40,1.49]**	51.6%	+0.10 [+0.05, +0.16]	0.041 [0.038,0.044]	103.5 [97.1,110.2]

The pooled ARL interval remains within the predeclared +0.2-frame non-inferiority margin, and median speedup exceeds $1.3\times$.

(b) Pooled FVD finite-sample diagnostic.

Estimator	Value	Estimated bias	95% CI
Naive FVD ($n = 3$, old protocol)	103.5	+6.7	wide / unreliable
Bootstrap-debiased FVD ($n = 30$)	96.8	+0.9	[89.4,104.6]

Evaluation scope. Table 4 in §4.2 reports decoded-frame FPS on $1\times$H100. Tables 14–22 report the remaining expanded results: a 90-rollout Matrix-Game evaluation, a third independent backbone, missing ablations, formal human/IDM equivalence tests, the endpoint-aware continuous-control fix, and preregistered qualitative cases, rapid-switch stress tests, and profiling results.

Matrix-Game expansion and FVD bias. Thirty independent rollouts per complex scene replace the earlier three-clip diagnostic. With rollout as the bootstrap cluster, ACTIONCACHE reaches a pooled median $1.44\times$ speedup [1.40,1.49] and ΔARL +0.10 [+0.05, +0.16], satisfying both confirmatory criteria (Table 14). Recomputing FVD inside each bootstrap draw estimates a +6.7 bias for the $n = 3$ estimator used in Table 3. The $n = 30$ debiased estimate is 96.8 [89.4,104.6] with residual bias +0.9 (Table 14). The main-table pooled FVD 103.5 lies inside this interval, so expanding the clip budget does not reverse the comparison. Relative to the three-seed Baseline noise floor of 87.4, the debiased gap is 9.4 rather than the 16.1 suggested by $n = 3$.

Cross-backbone and implementation validity. On the independently implemented Oasis-500M backbone, ACTIONCACHE reaches $1.52\times$ speedup with ΔARL +0.13 frame. Together with Vid2World and Matrix-Game, all three backbones therefore show speedup above one while keeping ΔARL within +0.2 frame (Table 5 in §4.2).

Table 15: Warm-up sweep on Vid2World DDIM-50 (5,003-episode training split). $W = 1$ is the measured speed–fidelity knee.

W	Skip ↑	Wall (s/ep) ↓	PSNR ↑	LPIPS ↓	FVD ↓	ARL ↓	AFS ↑
0	73.61%	184.2	19.48	0.325	113.4	1.24	0.085
1 (default)	71.14%	199.75	19.76	0.318	107.8	1.10	0.091
2	68.91%	212.4	19.82	0.315	105.9	1.07	0.092
4	64.48%	243.8	19.88	0.312	103.7	1.05	0.094

Warm-up sweep. The warm-up sweep is run on the 5,003-episode training split and selects $W = 1$: relative to this operating point, $W = 2$ recovers 0.06 dB PSNR at a cost of 2.23 skip points, while $W = 4$ recovers 0.12 dB at a cost of 6.66 skip points (Table 15).

Scheduler capacity controls. ACTIONCACHE attains 76% of the oracle's speedup increment and slightly outperforms the small learned scheduler on ARL, LPIPS, and FVD (Table 16).

Table 16: Scheduler capacity controls on Matrix-Game complex scenes. The oracle observes post-forward drift and is unattainable online.

Scheduler	Skip ↑	Speedup ↑	ARL ↓	LPIPS ↓	FVD ↓
Baseline (no cache)	0.0%	1.00	1.09	0.024	87.4
Oracle drift	58.7%	1.58	1.11	0.031	94.2
ACTIONCACHE	51.6%	1.44	1.19	0.041	103.5
Learned action scheduler	53.9%	1.47	1.22	0.045	108.7

Additional ablations. The full transition-by-periodic factorial has a PSNR interaction of -0.24 dB $[-0.31, -0.15]$ (Table 2 in §4.1), confirming that the two conditions are not additive. The Vid2World mouse sweep independently selects $\varepsilon_m = 0.05$ (Table 11).

Table 17: Predeclared TOST equivalence tests for ACTIONCACHE versus Baseline. Difference is ACTIONCACHE−Baseline (or %ours−50 for 2AFC). Intervals are 90% CIs, as required by two one-sided tests.

Comparison	Margin	Difference	90% CI	TOST p	Result
2AFC preference vs. 50%	±7 pp	−0.5 pp	[−3.6, +2.6]	0.003	equivalent
Responsiveness Likert	±0.25	−0.09	[−0.18, +0.00]	0.011	equivalent
Temporal-consistency Likert	±0.25	−0.10	[−0.19, −0.01]	0.019	equivalent
IDM-ARA	±3 pp	−1.8 pp	[−3.0, −0.6]	0.041	equivalent

Human and inverse-dynamics equivalence tests. Forty-eight analysed participants completed 60 balanced paired trials each. The preregistered two-one-sided tests establish equivalence to Baseline for 2AFC preference, responsiveness, temporal consistency, and IDM action-recovery accuracy under the stated margins (Table 17).

Human-study governance. Table 18 documents recruitment, exclusions, compensation, ethics determination, trial design, inter-rater agreement, and the independence of the frozen IDM.

Endpoint-aware continuous control. To expose accumulation hidden by the original consecutive detector, we apply same-direction mouse increments of 0.01–0.04 for spans $q \in \{1, 2, 3, 4, 6, 8, 16\}$, all below $\varepsilon_m = 0.05$ per call. Alternating-sign, random-sign, and real-user slow-pan trajectories are controls. Comparing each action to the action at the latest refresh reduces Exceed@1.10 on coherent $0.04/8$ motion from 27.4% to 3.6%, while the span sweep retains 1.31–1.39× speedup (Table 19, panels (a)–(b)). The deployed method therefore uses the endpoint-aware detector.

Table 18: Human-study governance and reliability.

Item	Value
Participants	48 analysed, 52 recruited, 4 excluded for $< 90\%$ attention
Recruitment / compensation	Prolific, gaming-familiarity prescreen, £9.00/hour, about 25 minutes
Consent / ethics	Anonymous 2AFC, institutional IRB exemption
Trials	60 paired clips per rater, balanced, randomised order
Agreement	Krippendorff $\alpha = 0.71$ (Likert), Fleiss $\kappa = 0.63$ (2AFC)
IDM independence	Frozen third-party model, method identities withheld

Table 19: Continuous-action accumulation and endpoint-aware detection. Panel (a) sweeps the coherent-motion span under the deployed endpoint detector. Panel (b) directly compares the original consecutive detector with endpoint-aware detection when every increment remains below $\varepsilon_m = 0.05$.

(a) Endpoint-aware span sweep on coherent 0.04/step motion.

Span	Skip ↑	Speedup ↑	Endpoint drift	Exceed@1.10 ↓	ΔPSNR	ARL ↓
1	60.0%	1.39	0.04	0.8%	−0.11	1.09
2	57.8%	1.37	0.08	1.2%	−0.18	1.11
3	55.9%	1.35	0.12	1.9%	−0.24	1.14
4	54.1%	1.34	0.16	2.5%	−0.29	1.16
6	52.0%	1.33	0.22	3.2%	−0.35	1.18
8	50.7%	1.33	0.28	**3.6%**	−0.39	1.19
16	48.3%	1.31	0.44	4.8%	−0.47	1.23

(b) Consecutive versus endpoint-aware mouse detection.

Pattern	$\|\Delta m\|$ / span	Endpoint	Detector	p95 drift	Exceed@1.10	ΔPSNR	ARL
Alternating	0.03 / 4	0.03	consecutive	1.07	1.9%	−0.31	1.12
Coherent ramp	0.03 / 4	0.09	consecutive	1.18	12.8%	−0.92	1.46
Coherent ramp	0.04 / 8	0.28	consecutive	1.31	27.4%	−1.48	1.91
Random sign	0.04 / 8	0.08	consecutive	1.10	4.2%	−0.43	1.18
Coherent ramp	0.03 / 4	0.09	endpoint	1.06	**1.4%**	−0.28	1.13
Coherent ramp	0.04 / 8	0.28	endpoint	1.08	**3.6%**	−0.39	1.19
Random sign	0.04 / 8	0.08	endpoint	1.09	3.1%	−0.34	1.15

Preregistered qualitative audit. Qualitative strips are selected by median, worst-case, p95, or boundary rules fixed before inspection, and a matched random sample accompanies them (Table 20). In particular, the WorldCache lag case reports the worst-ARL single steering event (3.08 frames), not its 2.86-frame cross-scene mean. Figure 4 shows a representative Vid2World-CSGO strip at the operating point reported in Table 1.

Table 20: Preregistered qualitative case selection. The released supplement contains synchronized strips and a matched random sample.

#	Case	Selection rule	Scene / method	ARL	LPIPS	Skip
1	Successful transition	Median-ARL turn event	GTA Drive / ACTIONCACHE	1.20	0.040	50.8%
2	Competitor lag	Same seed/frame, worst-ARL competitor	GTA Drive / WorldCache	3.08	0.149	12.4%
3	Coherent drift	p95 endpoint drift under old detector	Mouse ramp / ACTIONCACHE	1.91	0.089	50.7%
4	Exact fallback	100% transition schedule, hash verified	Rapid switch / ACTIONCACHE	1.09	0.024	0.0%

Actual rapid-switch rollouts. We execute keyboard, mouse, and mixed trajectories that alternate every $8, 4, 2$, or 1 action chunks, plus recorded real-user fast-control sequences. Each condition uses at least 50 paired rollouts. In addition to quality and speed, we report output-hash agreement and maximum tensor error against the uncached run. The switch-every-chunk condition tests, rather than assumes, the baseline fallback (Table 21).

Joint ablation and cluster-aware drift statistics. We run the complete 2×2 factorial for transition detection and periodic refresh and report their interaction, avoiding the invalid interpretation of two leave-one-out rows as a joint removal (Table 2 in §4.1). For drift, the analysis hierarchy is scene



Figure 4: Representative Vid2World-CSGO rollout. Under identical model, action, and sampling settings, ACTIONCACHE closely tracks the uncached baseline while achieving a 70.29% skip rate and $3.25\times$ speedup on the official 500-episode test split.

Table 21: Actually executed rapid-switch rollouts. "Hash" is the fraction exactly matching the uncached output under deterministic execution.

Mixed-action schedule	Transition rate	Skip	Speedup	ARL	LPIPS	Hash match
Stable	0%	60.0%	1.39	1.08	0.035	–
Switch every 8 chunks	12.5%	52.6%	1.33	1.09	0.037	–
Switch every 4 chunks	25.0%	44.7%	1.25	1.10	0.039	–
Switch every 2 chunks	50.0%	31.4%	1.17	1.11	0.040	–
Switch every chunk	100%	0.0%	1.00	1.09	0.024	100%

Measured per-call action-check overhead is 0.008%. The maximum absolute tensor difference at 100% transitions is 0.

$\rightarrow$ rollout $\rightarrow$ chunk $\rightarrow$ slot. We fit transition and continuous-action effects with rollout-clustered standard errors, a mixed-effects model, block permutations, and leave-one-scene-out checks (Table 8).

Latency and memory profile. After fixed warm-up, CUDA events and explicit synchronisation measure at least 1,000 compute and skip calls. We report decision/probe p50 and p95, cache read/write, full path latency, peak allocated/reserved memory, and persistent cache payload under fixed batch, resolution, dtype, GPU, and power mode (Table 22). The action check remains in the microsecond range, versus millisecond-scale DiCache probing and X-Cache fingerprinting. This gap bounds the decision overhead of each policy but is not the source of the end-to-end gain, which follows from how many calls are skipped.

Table 22: Latency and memory profile. Measurements use CUDA events and explicit synchronisation after warm-up. p50/p95 values cover 1,000 calls.

Method	Decision/probe p50	p95	Skip path	Compute path	Peak alloc.	Cache
Baseline	–	–	–	815 ms	31.2 GiB	0 MiB
DiCache	5.9 ms	7.3 ms	6.4 ms	814 ms	32.0 GiB	408 MiB
X-Cache	2.3 ms	3.1 ms	2.6 ms	813 ms	31.9 GiB	596 MiB
ACTIONCACHE	**8.4** μ**s**	**13.1** μ**s**	**0.38 ms**	812 ms	31.6 GiB	412 MiB

Table 23: Per-method configurations and sweep grids. Panel (a) lists the swept hyperparameter, cache substrate, and published default. Panels (b) and (c) list the per-backbone grid and the selected Pareto-front configuration. Every selected point lies in the corresponding backbone's grid and, for Vid2World, is chosen on the 5,003-episode training split before the 500-episode test evaluation. ACTIONCACHE follows the same Pareto-frontier selection rule.

(a) Swept knobs, substrates, and published defaults.

Method	Swept knob	Cache substrate	Published default
DiCache (Bu et al., 2026)	profile threshold τ_D	shared, fused-CFG	0.15 (HunyuanVideo)
FlowCache (Ma et al., 2026)	per-chunk redundancy τ_F	per-chunk KV	0.12 (MAGI-1)
ARCache (Nan et al., 2026)	HGC history-drift τ_H	history-guided	0.20 (SkyReels-V2)
WorldCache (Feng et al., 2026)	curvature budget ρ	heterogeneous token	0.25 (Voyager)
X-Cache (Zeng et al., 2026)	dual-metric fingerprint τ_X	cross-chunk KV	0.15 (X-World)
ACTIONCACHE	K, ε_m	shared, fused-CFG	$K=4$, $\varepsilon_m=0.05$

(b) Vid2World (DDIM-50) grid and selected point.

Method	Sweep grid	Selected
DiCache (Bu et al., 2026)	$\{0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.40\}$	$\tau_D=0.20$
FlowCache (Ma et al., 2026)	$\{0.04, 0.08, 0.10, 0.12, 0.16, 0.20, 0.30\}$	$\tau_F=0.10$
ARCache (Nan et al., 2026)	$\{0.08, 0.12, 0.16, 0.20, 0.25, 0.30\}$	$\tau_H=0.16$
WorldCache (Feng et al., 2026)	$\{0.10, 0.15, 0.20, 0.25, 0.30, 0.40\}$	$\rho=0.20$
X-Cache (Zeng et al., 2026)	$\{0.05, 0.10, 0.12, 0.15, 0.20, 0.25, 0.30\}$	$\tau_X=0.12$
ACTIONCACHE	$K\in\{2,3,4,6,8\}$, $\varepsilon_m\in\{.01, .025, .05, .10, .20\}$	$\varepsilon_m=0.05$, $K=4$

(c) Matrix-Game 2.0 grid and selected point.

Method	Sweep grid	Selected
DiCache (Bu et al., 2026)	$\{0.05, 0.10, 0.15, 0.20, 0.25, 0.30\}$	$\tau_D=0.10$
FlowCache (Ma et al., 2026)	$\{0.08, 0.12, 0.16, 0.20, 0.30\}$	$\tau_F=0.16$
ARCache (Nan et al., 2026)	$\{0.12, 0.16, 0.20, 0.22, 0.25, 0.30\}$	$\tau_H=0.22$
WorldCache (Feng et al., 2026)	$\{0.15, 0.20, 0.25, 0.30, 0.40\}$	$\rho=0.30$
X-Cache (Zeng et al., 2026)	$\{0.10, 0.15, 0.18, 0.20, 0.25, 0.30\}$	$\tau_X=0.18$
ACTIONCACHE	$K\in\{2,3,4,6,8\}$, $\varepsilon_m\in\{.01, .025, .05, .10, .20\}$	$\varepsilon_m=0.05$, $K=4$

Selection criterion. We select non-dominated configurations on the joint Wall, perceptual-quality, and control-quality front, breaking ties by distance to the method's median speedup. For ACTIONCACHE, the K sweep is reported on the Vid2World training split and the ε_m sweep on Matrix-Game. Their joint operating point is frozen for the reported test evaluation. Full sweep logs and selection scripts will accompany the reproducibility package.

I PER-METHOD CONFIGURATIONS AND SWEEP GRIDS

This appendix documents the configurations of the five comparison methods used throughout the main, layering, and cross-backbone analyses. All methods in Tables 1–3 share the same checkpoint, evaluation split, seed list, and timing boundary. Each decision rule is ported to the backbone-appropriate cache substrate. The principal remaining degree of freedom is each method's exposed skip-control hyperparameter. We treat that single knob explicitly: we (i) enumerate it for every baseline, (ii) bracket its published-default value with a grid wide enough to span both the low-skip / high-fidelity regime and the high-skip / lower-fidelity regime on each of the two backbones, and (iii) select the configuration on each method's own backbone-specific Pareto front before reporting it in the main tables.

Why a single dominant knob per comparison method. Each comparison exposes one dominant skip-control quantity: the profile threshold for DiCache (Bu et al., 2026), per-chunk redundancy for FlowCache (Ma et al., 2026), history drift for ARCache (Nan et al., 2026), curvature budget for WorldCache (Feng et al., 2026), and dual-metric fingerprint threshold for X-Cache (Zeng et al., 2026). Other method-specific hyperparameters are kept at their published defaults across our sweep so that the comparison is *at parity with how each baseline is deployed on its own release backbone*. Fixing the same secondary hyperparameters across methods would have homogenised the sweep at

the cost of deviating from the baseline's deployment recipe, biasing each row away from its published behaviour.

Sweep methodology. For Vid2World, every comparison method is evaluated on every grid point of its sweep grid on the 5,003-episode training split, which excludes the official DIAMOND test episodes. For Matrix-Game we use the same six-scene × three-seed development protocol. The resulting per-grid-point metric tuple (Skip%, Wall, PSNR/ARL, LPIPS, FVD, AFS) is comparable across methods. We then compute, for each method, the subset of swept configurations that are not strictly dominated on the joint (Wall↓, PSNR↑ *or* ARL↓, LPIPS↓, FVD↓) front, using PSNR on Vid2World (ground-truth available) and ARL on Matrix-Game (no ground-truth video, but a control-conditioned indicator available on every frame). Where more than one configuration remains on the front, we report the one closest to the per-row median speedup, so that no row is silently chosen for being either the fastest or the highest-fidelity configuration of the method in isolation. Selected Vid2World configurations are frozen before the 500-episode test evaluation in Table 1.

Reproducibility of the sweep. We will release the full per-grid-point logs, split manifest, and selection scripts used to produce Table 23. They allow every additional-baseline operating point to be audited or re-selected under an alternative criterion without re-running a model.

J LIMITATIONS, VALIDITY, AND RESPONSIBLE EVALUATION

Scope and external validity. ACTIONCACHE is evaluated on three open interactive world-model backbones, one CSGO corpus, and the scenarios distributed with Matrix-Game. The samplers cover long DDIM and short distilled regimes, but they do not establish generality to closed models, substantially different action spaces, or unconstrained long-horizon real-user sessions. Vid2World provides an official test split but no validation split. Operating points are therefore selected on the 5,003 training episodes, and Table 1 is reported on the untouched 500-episode test split. Matrix-Game has no held-out corpus. The confirmatory study therefore expands independent rollouts and uncertainty. The 1.60× Matrix-Game result is a peak on the four-step TEMPLE RUN scenario, while the pooled median is 1.44×.

Method and metric limitations. Actions are an informative but incomplete prior for cache validity: most measured drift variance remains attributable to non-action state. The endpoint-aware detector substantially reduces coherent sub-threshold accumulation but does not remove latent or context drift (Appendix H, Table 19, panels (a)–(b)). The conditional bounds in Appendix E neither estimate their constants nor verify their assumptions online and therefore provide no distribution-free drift guarantee. ARL and AFS are author-defined indicators. Their agreement with human, inverse-dynamics, and ground-truth-anchored measurements reduces, but does not eliminate, sensitivity to the chosen transition and pixel thresholds.

Comparison and statistical validity. The main comparison re-implements each cache's decision rule on a shared substrate to isolate the scheduler. This is the strongest protocol on which these methods can be compared on the target backbones. Unported components include the multi-stage or cross-chunk structures that several methods rely on. Appendix G.3 additionally layers the same action gate on WorldCache's released HunyuanVoyager-13B and Aether-5B implementations. Each method is reported at a single Pareto-frontier point. The full threshold sweeps are in Appendix I.

Human-participant reporting. The blind control-fidelity study uses 48 analysed Prolific raters and does not collect sensitive attributes. Recruitment, exclusions, compensation, the institutional exemption, randomisation, and inter-rater agreement are reported in Appendix H, Table 18. TOST establishes equivalence only within the preregistered margins of Appendix H, Table 17. It does not imply exact equality or transfer to other populations and tasks.