PERCEPTION–ACTION CONDITIONED EFFICIENT INFERENCE FOR INTERACTIVE VIDEO WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Interactive video world models enable users and agents to continuously explore generated environments via action-conditioned visual generation, benefiting applications in game simulation, autonomous driving, and embodied intelligence. Nevertheless, the repeated Diffusion Transformer (DiT) denoising conditioned on the long visual history incurs substantial latency, severely hindering responsive interaction. Existing approaches decrease the latency through reducing redundant computations in generated context, attention and denoising, by exploring the relevance between historically generated contents and the one under the current action. However, they generally rely on limited guidance signals for compression and adopt fixed reduction schemes, overlooking perceptual redundancy and dynamiclly varying computation demands, thus suffering significant performance degradation. To address this issue, we propose *Perception–Action Conditioned Efficient Inference* (PACE), a training-free framework that synergizes the perceived visual evidence with action cues to dynamically allocate computation. Specifically, we firstly develop the Block-wise Context Compression method to filter historical blocks based on visual redundancy and target-view relevance. Subsequently, we employ Action-Conditioned Attention Sparsity by optimally distributing attention budgets between past and current tokens. Besides, we design the Interruption-Aware Cache Reuse strategy, by scheduling feature reuse based on visual stability while conservatively adjusting during abrupt action transitions. On HY-WorldPlay and Matrix-Game-3.0, our method achieves $\mathbf{2.77}\times$ and $\mathbf{1.74}\times$ DiT-core speedup, respectively, remarkably promoting both inference latency and peak memory overhead with improved generation fidelity in most cases, compared to the state-of-the-art approaches. We will release source code upon acceptance.

1 INTRODUCTION

Interactive video world models, predominantly built on Diffusion Transformer (DiT) backbones, have recently emerged as a prominent paradigm for modeling complex, dynamic environments. By generating future visual states conditioned on action sequences, they enable users and agents to continuously explore and interact with evolving environments (Yang et al., 2024; Bruce et al., 2024; Valevski et al., 2025), benefiting broad applications such as game simulation and embodied intelligence. However, the long-horizon interaction typically requires each new segment to be denoised over multiple timesteps while conditioning on the increasingly long visual history (Sun et al., 2025; Wang et al., 2026; Yu et al., 2025), incurring substantial computational overhead and inference latency, ultimately undermining real-time interactivity (Peebles & Xie, 2023; Lu et al., 2026).

In order to improve inference efficiency, a variety of works have explored several complementary directions. In general, context compression or pruning explicitly shortens the conditioning sequence (Yu et al., 2025; Zhang et al., 2025b; Yi et al., 2026); sparse attention skips fine-grained query-key dot-product blocks (Zhang et al., 2025d; Xi et al., 2025; Lv et al., 2026); and caching methods reuse intermediate feature updates across denoising timesteps (Zhao et al., 2025; Lyu et al., 2025; Liu et al., 2025; Kahatapitiya et al., 2025). Some recent studies have adapted further these techniques to video world models, explicitly incorporating interaction signals into the optimization process (Yang et al., 2026; Lu et al., 2026). However, they rely solely on the action-induced Field of View (FoV) signal to guide compression and adopt fixed acceleration configurations, risking the removal of critical information and thus yielding degraded world-modeling consistency.

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.

Specifically, there are three key limitations: **(1) Coarse-grained historical context compression limits reference coverage.** Given the accumulated visual context, existing methods solely adopt explicit FoV cues to select a fixed number of previous frames for full retention. This frame-level strategy overlooks the perceptual redundancy across overlapping regions, wasting the limited context capacity that could otherwise accommodate complementary visual content, leading to insufficient evidence for faithful world modeling. **(2) Rigid sparse-attention allocation ignores the differing modeling priorities across interaction modes.** The native multi-head attention involves dot-products over both the current denoising segment and historical context, with different heads exhibiting heterogeneous preferences for the two sources. Meanwhile, interaction modes, *e.g.* exploration and revisitation intuitively prioritize current and historical information differently, where the common scheme of applying a fixed sparsity allocation may discard critical computations. **(3) Stale perception signals introduce unreliable cache reusing decisions.** Existing caching methods determine reuse solely from previously generated visual states, which reflect only past generation dynamics. Such signal becomes unreliable when the upcoming action abruptly changes the target view, as reuse may still by triggered despite the increased generation difficulty. In this case, reusing the unstable feature updates may degrade generation quality.

To address these issues, we propose *Perception–Action Conditioned Efficient Inference* (PACE), a novel training-free framework for accelerating interactive video world models. PACE emphasizes a key principle: both perceived visual evidence and action cues jointly enable more reliable computation allocation. Visual history reveals the existing redundancy, but not whether such redundancy remains exploitable for the upcoming target view, while the action signal acts conversely. By combining these complementary signals, PACE jointly guides compression on three dimensions. Specifically, Block-wise Context Compression combines target-view geometry and visual redundancy to select historical blocks; Action-Conditioned Attention Sparsity Allocation adjusts the attention budget to the interaction state and allocates it between historical and current keys at the layer-head level; and Interruption-aware Cache Reuse plans reuse from recent visual stability while reducing it when the upcoming action signals a large viewpoint change.

In summary, our contributions are three-fold:

- We propose a training-free perception–action conditioned framework PACE by synergizing the perceived visual evidence with action cues to dynamically allocate inference computation.
- We design three novel components for hybridly decreasing latency, including Block-wise Context Compression for redundant historical context removal, Action-Conditioned Attention Sparsity Allocation for attention computation reduction, and Interruption-aware Cache Reuse for denoising acceleration.
- We extensively evaluate PACE on HY-WorldPlay and Matrix-Game 3.0, achieving $2.77\times$ and $1.74\times$ DiT-core speedups, respectively, while decreasing the peak memory overhead and preserving higher fidelity in most cases, compared to the state-of-the-art methods.

2 Related Work

Interactive video world models incur substantial inference cost from repeatedly processing long visual histories during generation. Existing acceleration methods mainly reduce this cost through historical-context compression and selection, sparse attention, and diffusion-cache reuse.

2.1 Historical Context Compression and Selection

Visual-history cost can be reduced by compressing history or retaining it selectively (Yu et al., 2025; Yi et al., 2026). FramePack gives historical frames importance-dependent context lengths, fitting longer history within a fixed context budget (Zhang et al., 2025b). Closer to our interactive setting, Light Interaction uses a camera-aware interaction signal to retain or discard each retrieved spatial-memory group as a whole (Lu et al., 2026).

Such group-level decisions treat retrieved-memory content uniformly, although typically only a subset is relevant to the upcoming target view. Selectors driven mainly by interaction or geometry also underuse perceptual redundancy in visual history.

2.2 Sparse Attention

Sparse attention restricts query–key interactions to reduce full attention's quadratic cost in video Transformers (Yuan et al., 2024; Zhang et al., 2025a). Video diffusion methods include local spatiotemporal tiles in Sliding Tile Attention (Zhang et al., 2025d), structured patterns in Sparse VideoGen and Sparse-vDiT (Xi et al., 2025; Chen et al., 2026), and trainable or adaptive block selection in VSA and VMoBA (Zhang et al., 2025c; Wu et al., 2026). These methods also reveal head heterogeneity.

Autoregressive video diffusion adds growing history with distinct current and past visual keys. Light Forcing varies historical frame and block selection across rollout stages while retaining the current chunk (Lv et al., 2026); Dummy Forcing assigns heads different historical receptive fields—sink, recent, or none—while keeping current keys available (Guo et al., 2026). In interactive world models, Light Interaction selects historical visual blocks per head, retains all current visual keys, and uses a shared historical retention ratio (Lu et al., 2026).

These autoregressive methods primarily sparsify history without adapting current–history allocation to the upcoming interaction state. We instead calibrate the visual-key budget by interaction state and allocate it across current and historical keys for each head, allowing both to be sparsified.

2.3 Diffusion Cache Reuse

Diffusion caching accelerates generation by reusing attention outputs, intermediate features, or model predictions across denoising steps (Ma et al., 2024b; Wimbauer et al., 2024; Zou et al., 2025; Ma et al., 2024a; Qiu et al., 2025). For example, PAB schedules attention reuse (Zhao et al., 2025), FasterCache and D2Cache estimate skipped features or predictions from cached states (Lyu et al., 2025; Liu et al., 2026), and TeaCache, AdaCache, EasyCache, and DiCache adapt reuse using denoising-time signals (Liu et al., 2025; Kahatapitiya et al., 2025; Zhou et al., 2025; Bu et al., 2026).

For autoregressive video and world models, WorldCache uses denoising-time motion and saliency to reuse features (Nawaz et al., 2026), MotionCache updates tokens from intra-chunk frame differences (Xu et al., 2026), and ARCache refreshes caches from inter-step history-token changes (Nan et al., 2026). The distilled interactive backbones HY-WorldPlay and Matrix-Game 3.0 use few-step generation (Sun et al., 2025; Wang et al., 2026), limiting online reuse adjustment. Light Interaction decides reuse from FoV revisitation before denoising (Lu et al., 2026), while DriveCache derives a pre-denoising reuse budget from planned ego motion (Yang et al., 2026). Both rely on individual interaction cues without jointly exploiting completed-history redundancy and the upcoming action, limiting reliability for these few-step interactive backbones. We instead plan reuse from completed-chunk changes and the upcoming action.

3 The Proposed Approach

3.1 Framework Overview

We consider DiT-based interactive video world models. At interaction round t, the model generates a visual segment x_t from the current action a_t and generated visual history $H_{t-1} = \{x_0, \ldots, x_{t-1}\}$:

$$x_t \sim p_\theta(x_t \mid H_{t-1}, a_t, c), \tag{1}$$

where c denotes the remaining conditioning inputs. In the camera-controlled setting, a_t specifies the target camera state P_t and field of view $\mathcal{F}_t$ before x_t is generated (Sun et al., 2025). Both visual history and target-view information are therefore available before generation, without accessing the unseen segment. We keep the generator parameters θ fixed and modify only inference. Within the backbone's historical context, *anchor history*, *retrieved history*, and *recent history* denote persistent reference, retrieved long-term, and most recently generated context, respectively. With model depth and hidden dimension fixed, we use

$$\mathcal{C}_{\mathrm{DiT}} \approx S\left(c_{\mathrm{tok}} n + c_{\mathrm{attn}} A\right), \qquad A = O(n^2) \text{ for dense attention}, \tag{2}$$

as a high-level abstraction of DiT inference cost. Here, S is the number of DiT forward passes actually executed, n is the visual sequence length processed by each pass, A is the number of visual query–key interactions actually evaluated, and c_{tok} and c_{attn} are fixed coefficients.

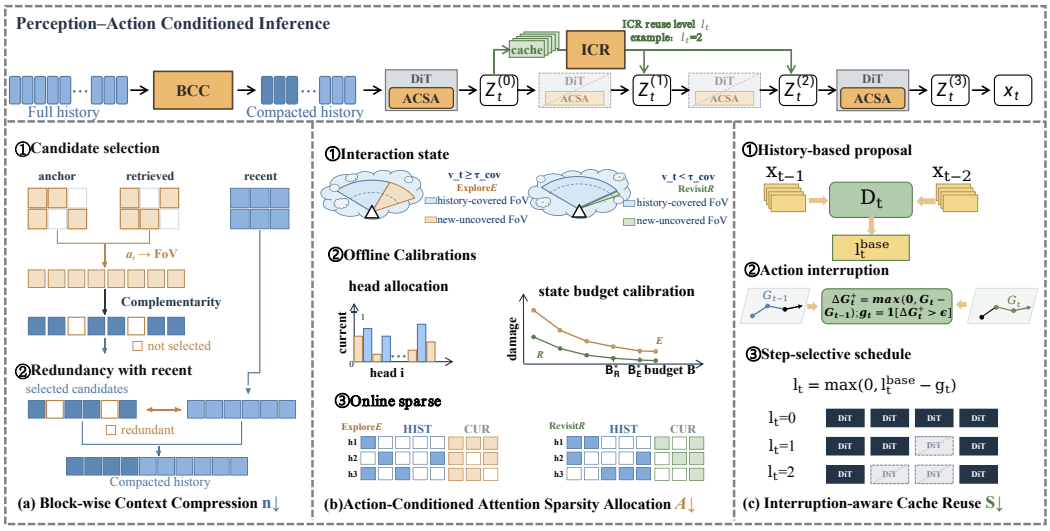


Figure 1: Overview of PACE. The top row shows how BCC, ACSA, and ICR act on historical context, attention computation, and denoising steps, respectively. (a) BCC selects and packs historical blocks using target-view relevance and visual redundancy. (b) ACSA allocates sparse-attention budgets between historical and current keys according to the interaction state. (c) ICR determines denoising reuse from history variation and action-induced FoV change.

The central question is which computation remains useful for the upcoming interaction. PACE follows a perception–action conditioned compute-allocation principle: completed visual evidence identifies potential redundancy, while action-derived target-view information indicates whether the corresponding history remains relevant to the upcoming view.

Figure 1 summarizes the framework. Block-wise Context Compression selects and physically packs historical blocks using target-view relevance and conditional visual redundancy, reducing n. Action-Conditioned Attention Sparsity Allocation uses the upcoming interaction state and calibrated head preferences to decide both how much visual attention to retain and where to allocate it, reducing A. Interruption-aware Cache Reuse derives a base reuse level from recent visual stability and lets the upcoming action conservatively interrupt that plan, reducing S.

3.2 Block-wise Context Compression

Historical context is not uniformly useful across space. Even when a historical frame supports the next segment as a whole, only some of its regions may contribute evidence for the target view. Frame- or history-group-level selection is therefore unnecessarily coarse. We partition the *anchor* and *retrieved history* into spatiotemporal blocks and select them under a prescribed per-round history budget, while retaining the *recent history* in full to preserve local continuity.

Target-conditioned Retention Value. For each candidate block b, we project samples from the upcoming target FoV into historical views and obtain a static relevance score $u(b)$. We first preselect candidates using their marginal target-view coverage and their content complementarity to blocks already selected from the same history role. For each preselected block, we then use geometric correspondence together with latent-content similarity and change to estimate its normalized redundancy $\widetilde{\xi}(b) \in [0,1]$ relative to the fully retained recent history. The final retention value used for role-wise ranking is

$$v(b) = u(b)\big(1 - \widetilde{\xi}(b)\big). \tag{3}$$

Therefore, a block is favored when it is relevant to the target view and not readily represented by the fully retained recent history. The marginal-coverage preselection, geometric correspondence, latent descriptor, and conditional redundancy estimators are detailed in Appendix A.

Selection and Physical Compaction. Anchor and retrieved blocks are ranked by their retention values, and the available capacity is allocated between the two roles while respecting the per-round

total budget; recent blocks remain fully retained. The unselected blocks are then removed at the input to the first transformer block. Surviving tokens are packed while preserving their original spatiotemporal indices for positional and camera conditioning. Thus, Block-wise Context Compression shortens the sequence processed by the transformer. Appendix A gives the role-wise allocation and tensor-packing procedure.

3.3 Action-Conditioned Attention Sparsity Allocation

A fixed sparse-attention policy overlooks two forms of heterogeneity. First, attention heads differ in their reliance on current and historical visual information. Second, the useful amount and distribution of attention change with the upcoming interaction. Action-Conditioned Attention Sparsity Allocation therefore separates allocation into two independently calibrated decisions: how much visual-key computation to retain for the interaction state, and how each layer-head distributes its retained access between current and historical keys.

Interaction State. Before generation, the action determines the target views $\mathcal{Q}_t$ of the next segment. Let $\mathcal{M}_t$ denote the available historical views. We measure the average fraction of the target FoV outside their accumulated coverage:

$$\nu_t = \frac{1}{|\mathcal{Q}_t|} \sum_{q \in \mathcal{Q}_t} \left[1 - \frac{\left| \mathcal{F}_q \cap \left(\bigcup_{j \in \mathcal{M}_t} \mathcal{F}_j \right) \right|}{|\mathcal{F}_q|} \right]. \tag{4}$$

The upcoming interaction is exploration (E) if $\nu_t \geq \tau_{\text{cov}}$ and revisitation (R) otherwise.

State-conditioned Head Allocation. Dense calibration trajectories reveal how each layer-head divides attention mass between current and historical visual keys. For layer l, head h, and state $s \in \{E, R\}$, let $M^{\text{cur}} l, h, s$ and $M^{\text{hist}} l, h, s$ denote the attention mass assigned to current and historical visual keys, respectively, aggregated over dense calibration samples in state s. We summarize this preference by

$$\rho_{l,h}^{(s)} = \frac{M_{l,h,s}^{\text{cur}}}{M_{l,h,s}^{\text{cur}} + M_{l,h,s}^{\text{hist}}}. \tag{5}$$

Although dense head preferences are strongly correlated across states, many layer-heads show consistent shifts toward current or historical keys (Figure 2(a)), motivating state-specific allocation maps.

We calibrate a monotonic mapping from each state's dense head preference to the fraction of available current visual keys retained under sparse attention, $\alpha_{l,h}^{(s)} = f_s(\rho_{l,h}^{(s)}; \theta_s^*)$. For each state, we select θ_s^* by minimizing the dense–sparse x_0 discrepancy after a fixed number of teacher-forced denoising steps on calibration trajectories. The fitted mappings produce two state-specific lookup tables. Appendix B specifies the calibration protocol and mapping family. Figure 2(b) shows how the two mappings convert dense head preferences into current-key allocations.

State Budget and Online Decision. The head mapping sets each layer-head's current-key retention fraction, while the state budget sets the total visual-key tile quota. Holding the mapping fixed, we measure discrepancy curves over visual-key budgets for exploration and revisitation. We then jointly select their operating points to minimize state-frequency-weighted discrepancy on the calibration set, subject to the average attention-compute budget of a uniform sparse policy. These points yield B_E and B_R.

At inference, state s selects B_s and its head-allocation table $\{\alpha_{l,h}^{(s)}\}_{l,h}$. Let N_{cur} denote the number of available current-key tiles, and let $K_{l,h,s}^{\text{cur}}$ and $K_{l,h,s}^{\text{hist}}$ denote the current- and historical-key tile quotas for each layer-head, respectively:

$$K_{l,h,s}^{\text{cur}} = \text{clip}_{[1, B_s - 1]} \left(\text{round} \left(\alpha_{l,h}^{(s)} N_{\text{cur}} \right) \right), \qquad K_{l,h,s}^{\text{hist}} = B_s - K_{l,h,s}^{\text{cur}}. \tag{6}$$

Clipping keeps both quotas nonzero, and separate Top-K searches select tiles from each partition. Calibration is offline; Appendix B details the budget optimization and routing.

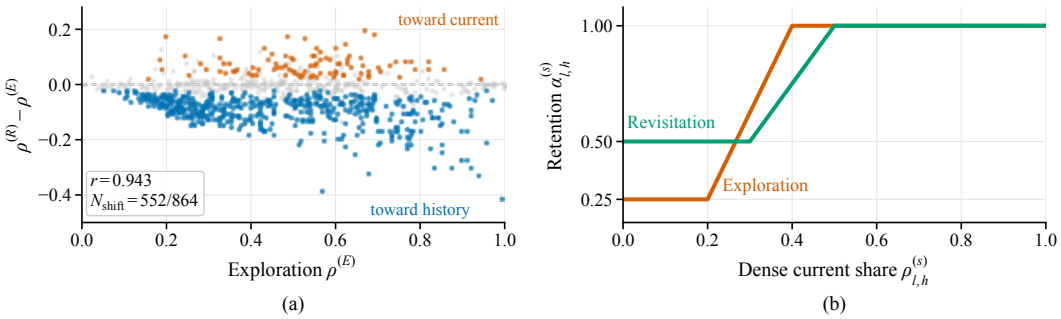


Figure 2: Interaction-state head preferences and state-conditioned allocation. (a) Exploration and revisitation head preferences from Eq. 5; colors indicate shifts toward historical or current keys, with gray denoting the remaining heads. (b) Calibrated mappings from dense head preferences to state-specific current-key retention.

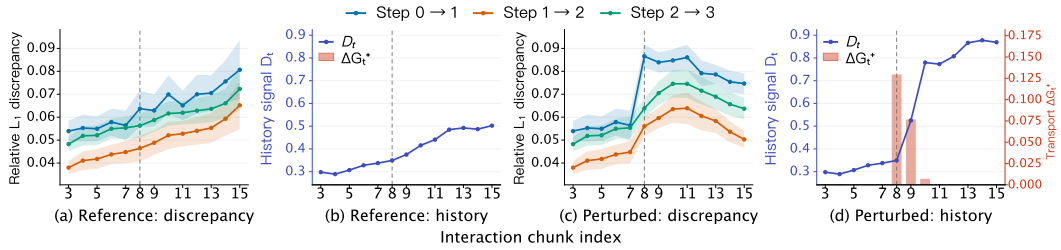


Figure 3: Denoising discrepancy and reuse-planning signals under an action perturbation. The two branches share history through chunk 7 and differ only in the action at chunk 8. Panels (a,c) show adjacent-step relative L_1 discrepancy, while (b,d) show D_t and ΔG_t^+. Curves show means with SEM; the dashed line marks chunk 8.

3.4 Interruption-aware Cache Reuse

Interactive video world models with few-step sampling leave only a small number of intermediate DiT evaluations available for reuse. Once the first and final evaluations are fixed, the remaining reuse opportunities are limited, leaving little room for online reuse adjustment based on the current denoising trajectory. Interruption-aware Cache Reuse therefore determines the reuse plan before denoising: visual history provides the base reuse level, while the next action corrects for changes not yet covered by the history.

History-based Reuse Plan. Before generating x_t, we compute the visual change between the two previously generated segments x_{t-1} and x_{t-2}:

$$D_t = \mathrm{MSE}\left(x_{t-1}, x_{t-2}\right). \tag{7}$$

For a common four-step sampling configuration, the first and final DiT evaluations are always recomputed, leaving two intermediate evaluations available for reuse. We therefore define three reuse levels using two thresholds:

$$\ell_t^{\text{base}} = \begin{cases} 2, & D_t < \tau_1, \\ 1, & \tau_1 \le D_t < \tau_2, \\ 0, & D_t \ge \tau_2, \end{cases} \qquad \tau_1 < \tau_2, \tag{8}$$

Here, $\ell_t^{\text{base}} \in \{0, 1, 2\}$ denotes the number of reusable intermediate DiT evaluations, while the first and final steps are always recomputed. Because D_t describes the preceding transition from x_{t-2} to x_{t-1}, it lags the reuse decision for x_t by one interaction round (Figure 3).

Action Interruption Gate. To correct this lag, we compute a multi-depth FoV transport score G_t between the preceding chunk and the target view, and take its positive increment relative to the

preceding transition:

$$\Delta G_t^+ = \max(0, G_t - G_{t-1}). \tag{9}$$

When this increment exceeds an interruption threshold ϵ, we reduce the base reuse level by one, yielding the final reuse level ℓ_t:

$$\ell_t = \max\left(0, \ell_t^{\text{base}} - \mathbf{1}[\Delta G_t^+ > \epsilon]\right). \tag{10}$$

Otherwise, the original reuse plan remains unchanged. The multi-depth transport and step-level reuse schedule are detailed in Appendix C.

4 Experimental Results and Analysis

4.1 Experimental setup

Models. We evaluate PACE on two representative interactive video world models with different architectures, including HY-WorldPlay (Sun et al., 2025) and Matrix-Game 3.0 (Wang et al., 2026), which differ in how they organize historical context and perform generation.

Datasets and Evaluation Metrics. Following the evaluation protocols of Light Interaction (Lu et al., 2026), we use 200 image–text pairs derived from VBench (Huang et al., 2024). Each pair is evaluated under two predefined camera trajectories, forward–backward and left–right (Sun et al., 2025; Lu et al., 2026), yielding 400 runs in total. We report results averaged over both trajectories, including the agreement with original generation using PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018), the revisit consistency using Self-Comparison (Lu et al., 2026) on pose-matched frames, and the overall video quality using the VBench (Huang et al., 2024) score. For inference efficiency, we report DiT-core latency, speedup ratio, and peak GPU memory.

Implementation Details. For HY-WorldPlay with fixed-trajectories, the sparse-attention selector uses a precomputed exploration/revisitation schedule derived from the planned target views. Following Light Interaction (Lu et al., 2026), the DiT-core latency excludes VAE decoding (Lu et al., 2026). We run 400 generations across four NVIDIA H100 GPUs and report the averaged latency. Additional model settings and component-specific hyperparameters are provided in Appendix.

4.2 Comparison with the State-of-the-Art Approaches

Compared Methods. We compare PACE against the base model (Original) and the state-of-the-art training-free accelerators: 1) the sparse attention methods including Sparse VideoGen (SVG) (Xi et al., 2025) and BSA (Team et al., 2025; Lu et al., 2026), 2) the denoising reuse method including TeaCache (Liu et al., 2025), and 3) the hybrid method including Light Interaction (Lu et al., 2026) that combines context management, sparse attention, and denoising reuse.

Inference Efficiency. As shown in Table 1, PACE achieves DiT-core speedups of $2.77\times$ and $1.74\times$ on HY-WorldPlay and Matrix-Game 3.0, respectively. On HY-WorldPlay, it also reduces peak memory from 76.57 GB to 51.31 GB. In contrast, single-dimension baselines achieve at most a $1.29\times$ speedup. On Matrix-Game 3.0, SVG and BSA reduce attention computation without lowering latency, showing that compute reduction does not necessarily translate into wall-clock speedup. Compared with Light Interaction, PACE achieves comparable or lower latency on both backbones while improving both generation fidelity and revisit consistency.

Generation Quality. Quality scores for Original and the baselines are quoted from Lu et al. (2026), while those for PACE are obtained from our evaluation. Compared with Light Interaction, the system-level baseline with the closest latency, our method improves PSNR, SSIM, and LPIPS on both backbones. On HY-WorldPlay, the two methods have comparable latency (41.38 s versus 42.17 s), while our method improves PSNR from 24.81 to 26.21 dB. On Matrix-Game 3.0, it likewise achieves higher fidelity at lower latency. For revisit consistency, our method improves all three Self-Comparison metrics on both backbones and also outperforms Original. This suggests that prioritizing target-view-relevant history while reducing weakly relevant historical interactions may limit interference from irrelevant history and improve scene consistency during revisits.

Table 1: Comparison on HY-WorldPlay and Matrix-Game 3.0. Metrics under *vs. Original* measure fidelity to full-computation outputs, while *Self-Comparison* measures revisit consistency between pose-matched frames. The 1st and 2nd highest results are marked in **bold** and underlined.

Method	vs. Original			Self-Comparison			VBench↑	Latency↓ (s)	Speedup↑	Mem.↓ (GB)
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓				
HY-WorldPlay										
Original	–	–	–	18.60	0.5678	0.2051	0.8190	114.67	1.00×	76.57
SVG	19.48	0.6028	0.2209	17.75	0.5299	0.2187	0.8082	141.64	0.81×	77.86
BSA	15.94	0.4639	0.3755	15.44	0.4205	0.3720	0.7943	88.58	1.29×	75.03
TeaCache	20.90	0.6588	0.1892	18.86	0.5743	0.2054	0.8150	101.13	1.13×	76.64
Light Interaction	24.81	0.6500	0.1788	18.85	0.5854	0.1963	0.8220	42.17	2.72×	54.66
PACE (Ours)	**26.21**	**0.6936**	**0.1430**	**20.02**	**0.6182**	**0.1584**	**0.8278**	**41.38**	**2.77×**	**51.31**
Matrix-Game 3.0										
Original	–	–	–	15.49	0.4685	0.4048	0.7432	25.64	1.00×	**40.36**
SVG	12.98	0.4170	0.5587	14.48	0.4949	0.4406	**0.7511**	114.89	0.22×	**40.36**
BSA	13.34	0.4228	0.5795	16.66	0.5326	0.4094	0.7336	173.76	0.15×	**40.36**
TeaCache	19.03	0.5619	0.3818	**18.84**	**0.5765**	**0.3602**	0.7146	26.32	0.97×	**40.36**
Light Interaction	17.76	0.5306	0.3692	14.63	0.4570	0.4424	0.7350	15.25	1.68×	**40.36**
PACE (Ours)	**20.10**	**0.6133**	**0.2824**	16.95	0.5196	0.3610	0.7315	**14.73**	**1.74×**	**40.36**

End-to-end Latency. Figure 4 profiles the complete HY-WorldPlay pipeline as components are progressively added. The method reduces total latency from 136.66 to 63.47 s, yielding a 2.15× end-to-end speedup. Since VAE decoding and other operations take around 22 s, the end-to-end speedup is lower than the 2.77× DiT-core speedup. The breakdown shows that context selection primarily reduces KV reconstruction time, whereas cache reuse primarily reduces denoising time.

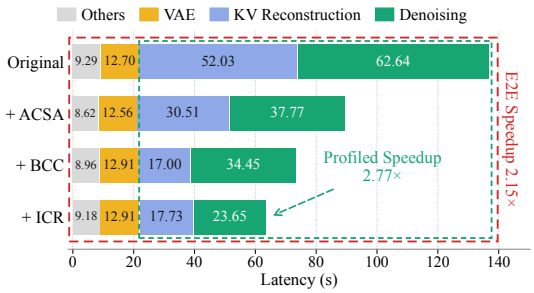


Figure 4: End-to-end latency reduction of PACE on HY-WorldPlay, decomposed by cumulatively enabling the main modules.

4.3 Ablation Study

In this section, we evaluate the effect of the proposed main components, including Action-Conditioned Attention Sparsity Allocation (ACSA), Block-wise Context Compression (BCC), and Interruption-aware Cache Reuse (ICR). We further analyze the design choices within each component on a fixed 40-clip subset containing 20 forward–backward and 20 left–right trajectories, in order to control the computational cost of these ablations.

On Main Components. We progressively add ACSA, BCC, and ICR to evaluate their contributions to overall efficiency. Since BCC relies on a block-sparse attention backend, we first enable sparse attention and then add context compression and cache reuse sequentially. As shown in Table 2, the overall speedup is promoted from 1.00× to 1.68×, 2.23×, and 2.77×. Memory savings come primarily from context compression, which reduces peak memory from 76.62 GB to 51.30 GB. Moreover, the generation quality metrics under the *Self-Comparison* setting together with the VBench score are also remarkably improved, with slight fluctuations under the *vs. Original* setting. These results clearly show that the proposed components progressively reduce inference cost while preserving generation fidelity and improving long-range scene consistency.

On Action-Conditioned Attention Sparsity Allocation. We first examine two design choices in ACSA: per-head history–current allocation (PHA) and interaction-state attention budgeting (IAB). We start from the uniform sparse baseline and keep the average attention budget approximately

Table 2: Ablation results of the proposed approach on HY-WorldPlay.

ACSA	BCC	ICR	vs. Original			Self-Comparison			VBench↑	Latency↓ (s)	Speedup↑	Mem.↓ (GB)
			PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓				
			–	–	–	18.60	0.5678	0.2051	0.8190	114.67	1.00×	76.57
✓			**26.81**	**0.7200**	**0.1211**	18.91	0.5797	0.1937	0.8276	68.28	1.68×	76.62
✓	✓		26.41	0.7034	0.1337	19.66	0.6084	0.1710	0.8274	51.45	2.23×	**51.30**
✓	✓	✓	26.21	0.6936	0.1430	**20.02**	**0.6182**	**0.1584**	**0.8278**	**41.38**	**2.77×**	51.31

Table 3: Ablation of interaction-state attention allocation at matched DiT-core latency. PHA and IAB denote per-head history–current allocation and state-aware budgeting, respectively.

PHA	IAB	Mutual			Latency (s)↓
		PSNR↑	SSIM↑	LPIPS↓	
		26.19	0.6889	0.1238	68.46
✓		26.26	0.6937	0.1182	**68.37**
✓	✓	**26.44**	**0.6982**	**0.1148**	68.59

Table 4: Effect of the FoV interruption gate on reuse fidelity and DiT-core latency. HV and FG denote historical-variation-based reuse and the FoV interruption guard, respectively.

HV	FG	Mutual			Latency (s)↓
		PSNR↑	SSIM↑	LPIPS↓	
		–	–	–	113.46
✓		28.08	0.7442	0.1070	**87.79**
✓	✓	**28.25**	**0.7524**	**0.1048**	89.23

matched across all sparse variants. As shown in Table 3, PHA improves the fidelity metrics by assigning different per-head history–current ratios for exploration and revisitation. IAB further improves performance by assigning different total attention budgets to the two interaction states. Meanwhile, the DiT-core latency retains around 68.5 s, indicating that the gains arise primarily from more effective allocation of attention computation rather than increased overall computation.

On Block-wise Context Compression. Under the same sparse-attention policy, we compare distinct historical-context selection methods. The frame-level baseline retains 10 complete frames, while all block-level variants use the same average block budget as 10 full frames. As shown in Table 5, block-wise similarity selection (BSS) based on cosine similarity and MAE reduces DiT-core latency and improves fidelity. Adding target-FoV guidance (TFG) further improves all agreement metrics. Finally, interaction-state context budgeting (ICB) for exploration and revisitation provides further fidelity gains without increasing the average budget.

Table 5: Ablation of historical-context selection strategies on HY-WorldPlay. BSS, TFG, and ICB denote block-wise similarity selection, target-FoV guidance, and interaction-state context budgeting, respectively.

BSS	TFG	ICB	Mutual			Latency (s)↓
			PSNR↑	SSIM↑	LPIPS↓	
			25.15	0.6493	0.1484	58.05
✓			25.94	0.6716	0.1361	**56.17**
✓	✓		26.01	0.6788	0.1296	56.28
✓	✓	✓	**26.10**	**0.6825**	**0.1268**	56.81

On Interruption-aware Cache Reuse. We examine historical-variation-based reuse (HV) and the FoV interruption guard (FG). As shown in Table 4, HV alone reduces DiT-core latency from 113.46 s to 87.79 s. Adding FG increases latency by only 1.44 s while improving PSNR, SSIM, and LPIPS, showing that upcoming viewpoint changes can interrupt stale reuse plans with negligible additional cost.

5 Conclusion

This work proposes PACE, a training-free inference framework for interactive video world models that uses visual history and upcoming actions to guide context compression, sparse-attention allocation, and denoising reuse. On HY-WorldPlay and Matrix-Game 3.0, it achieves 2.77× and 1.74× DiT-core speedups, respectively, while maintaining higher fidelity to full-computation outputs than existing accelerators at comparable or lower latency. Ablations show that these mechanisms reduce inference cost while preserving generation quality and improving revisit consistency.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the preparation of calibration data and with translation. Specifically, generative AI was used to rewrite captions for the calibration set used in per-head sparse-attention allocation calibration. The corresponding images and source captions were drawn from VBench samples that do not overlap with the evaluation set, and the caption-rewriting procedure followed the style of caption rewriting used in Light Interaction. Generative AI tools were also used to assist with translating related papers during literature reading. We did not use generative AI tools to develop theoretical models or conceptual frameworks, propose or refine research hypotheses, design or provide feedback on the research methodology or experiments, implement the proposed methods, clean or reformat datasets, or interpret experimental results. Tasks involving the formulation or proof of mathematical claims, as well as qualitative or thematic data analysis, were not applicable to this work. Additionally, we used generative AI tools for limited manuscript editing to improve readability and for software-code assistance. The latter was limited to engineering tasks such as data loading, evaluation-pipeline construction, logging, script integration, code refactoring, interface adaptation, debugging, and integrating author-developed algorithmic logic into the complete experimental pipeline; generative AI was not used to design or implement the core algorithms of the proposed method. We reviewed and verified all AI-assisted work, including manually reviewing and testing AI-assisted code. We take responsibility for the final content of this work, including any text, claims, code, or other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies inference efficiency for interactive video generation. Its direct experiments do not require personal data unless a future human evaluation is added. The broader risks of generative video systems, including misleading media and representational bias, remain applicable. Any human study, new dataset, or user-generated trace will require a separate review of consent, privacy, licensing, and data handling before inclusion.

REFERENCES

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C.Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando De Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning*. PMLR, 2024.

Jiazi Bu, Pengyang Ling, Yujie Zhou, Yibin Wang, Yuhang Zang, Dahua Lin, and Jiaqi Wang. Dicache: Let diffusion model determine its own cache. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 73778–73803, 2026.

Pengtao Chen, Xianfang Zeng, Maosen Zhao, Mingzhu Shen, Wei Cheng, Gang Yu, and Tao Chen. Sparse-vdit: Unleashing the power of sparse attention to accelerate video diffusion transformers. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 2957–2965. AAAI Press, 2026.

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

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818, June 2024.

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.

Enhuai Liu, Yunke Wang, Changming Sun, and Chang Xu. D2cache: Second-order delta caching for higher video diffusion acceleration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 43589–43599, June 2026.

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

Jiacheng Lu, Haoyi Zhu, Sipei Yi, Enze Xie, Yu Li, and Cheng Zhuo. Light interaction: Training-free inference acceleration for interactive video world models, 2026.

Chengtao Lv, Yumeng Shi, Yushi Huang, Ruihao Gong, Shen Ren, and Wenya Wang. Light forcing: Accelerating autoregressive video diffusion via sparse attention. *arXiv preprint arXiv:2602.04789*, 2026.

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 Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 33132–33156, 2025.

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

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

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.

Umair Nawaz, Ahmed Heakl, Ufaq Khan, Abdelrahman Shaker, Salman Khan, and Fahad Shahbaz Khan. Worldcache: Content-aware caching for accelerated video world models, 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.

Junxiang Qiu, Lin Liu, Shuo Wang, Jinda Lu, Kezhou Chen, and Yanbin Hao. Accelerating diffusion transformer via gradient-optimized cache. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17608–17617, October 2025.

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 model. *arXiv preprint*, 2025.

Meituan LongCat Team, Xunliang Cai, Qilong Huang, Zhuoliang Kang, Hongyu Li, Shijun Liang, Liya Ma, Siyu Ren, Xiaoming Wei, Rixu Xie, and Tong Zhang. Longcat-video technical report, 2025.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 73754–73776, 2025.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Zile Wang, Zexiang Liu, Jiaxing Li, Kaichen Huang, Baixin Xu, Fei Kang, Mengyin An, Peiyu Wang, Biao Jiang, Yichen Wei, Yidan Xietian, Jiangbo Pei, Liang Hu, Boyi Jiang, Hua Xue, Zidong Wang, Haofeng Sun, Wei Li, Wanli Ouyang, Xianglong He, Yang Liu, Yangguang Li, and Yahui Zhou. Matrix-game 3.0: Real-time and streaming interactive world model with long-horizon memory, 2026.

Felix Wimbauer, Bichen Wu, Edgar Schoenfeld, Xiaoliang Dai, Ji Hou, Zijian He, Artsiom Sanakoyeu, Peizhao Zhang, Sam Tsai, Jonas Kohler, Christian Rupprecht, Daniel Cremers, Peter Vajda, and Jialiang Wang. Cache me if you can: Accelerating diffusion models through block caching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6211–6220, June 2024.

Jianzong Wu, Liang Hou, Haotian Yang, Ye Tian, Pengfei Wan, Di ZHANG, and Yunhai Tong. Vmoba: Mixture-of-block attention for video diffusion models. In *International Conference on Learning Representations*, volume 2026, pp. 132534–132548, 2026.

Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, Jianfei Chen, Ion Stoica, Kurt Keutzer, and Song Han. Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 68208–68224. PMLR, 2025.

Jing Xu, Yuexiao Ma, Xuzhe Zheng, Xing Wang, Shiwei Liu, Chenqian Yan, Xiawu Zheng, Rongrong Ji, Fei Chao, and Songwei Liu. Motion-aware caching for efficient autoregressive video generation, 2026.

Jianchun Yang, Jian Liang, Xianda Guo, Pinhan Fu, Yanlun Peng, Conglang Zhang, Wenke Huang, and Mang Ye. Drivecache: Action-aware caching for driving world model inference, 2026.

Sherry Yang, Yilun Du, Seyed Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 45210–45234, 2024.

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

Jiwen Yu, Jianhong Bai, Yiran Qin, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Context as memory: Scene-consistent interactive long video generation with memory retrieval. *arXiv preprint arXiv:2506.03141*, 2025.

Zhihang Yuan, Hanling Zhang, Pu Lu, Xuefei Ning, Linfeng Zhang, Tianchen Zhao, Shengen Yan, Guohao Dai, and Yu Wang. Ditfastattn: Attention compression for diffusion transformer models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 1196–1219. Curran Associates, Inc., 2024.

Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. SpargeAttention: Accurate and training-free sparse attention accelerating any model inference. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 76397–76413. PMLR, 13–19 Jul 2025a.

Lvmin Zhang, Shengqu Cai, Muyang Li, Gordon Wetzstein, and Maneesh Agrawala. Frame context packing and drift prevention in next-frame-prediction video diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 30546–30566. Curran Associates, Inc., 2025b. doi: 10.52202/085713-1024.

Peiyuan Zhang, Yongqi Chen, Haofeng Huang, Will Lin, Zhengzhong Liu, Ion Stoica, Eric Xing, and Hao Zhang. Faster video diffusion with trainable sparse attention. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 152509–152534. Curran Associates, Inc., 2025c. doi: 10.52202/085713-5099.

Peiyuan Zhang, Yongqi Chen, Runlong Su, Hangliang Ding, Ion Stoica, Zhengzhong Liu, and Hao Zhang. Fast video generation with sliding tile attention. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 74714–74731. PMLR, 13–19 Jul 2025d.

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 Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Xuanlei Zhao, Xiaolong Jin, Kai Wang, and Yang You. Real-time video generation with pyramid attention broadcast. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 3296–3319, 2025.

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, 2025.

Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 75467–75488, 2025.

A Block-wise Context Compression Details

This section expands the selection rule underlying Block-wise Context Compression. Target-view geometry identifies relevant historical blocks and establishes correspondence for content comparison. Selection then accounts for complementarity within a history role and redundancy with the fully retained recent history before allocating capacity across roles.

A.1 Target-view geometry and block correspondence

At interaction round t, let $\mathcal{B}_t = \mathcal{B}_t^{\mathrm{anc}} \cup \mathcal{B}_t^{\mathrm{ret}} \cup \mathcal{B}_t^{\mathrm{rec}}$ denote candidate blocks from anchor, retrieved, and recent history. The current noisy latent is not part of this pool. Let $\mathcal{S}_t$ be samples drawn from the planned target views $\mathcal{Q}_t$. For a historical block b, $C_t(b) \subseteq \mathcal{S}_t$ contains the target samples whose projections fall inside b. Its normalized target-view relevance is

$$u(b) = \frac{|C_t(b)|}{\max_j |C_t(j)| + \varepsilon}. \tag{11}$$

Projection uses hypothetical depths to establish geometric support; it does not require scene depth or occlusion estimates.

Anchor and retrieved blocks are greedily preselected within their respective roles. For a role r, let K be the blocks already selected from that role, and let $\mathcal{P}_r$ contain the blocks in the other available roles that are temporarily treated as retained support. Write $U_r(K)$ for the target samples covered by K, and let $s_{K\cup\mathcal{P}_r}(q)$ count the distinct history tiles in $K \cup \mathcal{P}_r$ that cover sample q. The geometric marginal gain is

$$g(b \mid K) = \sum_{q \in C_t(b) \setminus U_r(K)} \log \frac{s_{K\cup\mathcal{P}_r}(q) + 2}{s_{K\cup\mathcal{P}_r}(q) + 1}. \tag{12}$$

The exclusion prevents repeated coverage within the role being preselected; the support count still reflects the other retained roles. Each role is preselected separately, so this stage does not jointly optimize the final anchor and retrieved sets.

Content is compared only after establishing target-view correspondence. We dilate each block's coverage set within a neighborhood on its history-tile grid, giving $\widetilde{C}_t(b)$. The directed overlap from b to j is

$$\omega_{bj} = \frac{|\widetilde{C}_t(b) \cap \widetilde{C}_t(j)|}{|\widetilde{C}_t(b)| + \varepsilon}. \tag{13}$$

The geometrically matched recent block for b is

$$r^*(b) = \arg\max_{r \in \mathcal{B}_t^{\mathrm{rec}}} \omega_{br}. \tag{14}$$

Dilation affects correspondence only; relevance and geometric preselection use the original coverage sets.

A.2 Content complementarity and conditional redundancy

For the valid latent positions of block b, channel-wise mean and standard-deviation pooling produces a descriptor

$$d_b = [\mu_b; \sigma_b] \in \mathbb{R}^{2C_{\mathrm{lat}}}, \tag{15}$$

where C_{lat} is the number of selected latent channels. Descriptors are centered by the mean over historical blocks and normalized with a norm floor, yielding $\widehat{d}_b$. The unnormalized channel means also give the magnitude difference

$$m_{bj} = \frac{1}{C_{\mathrm{lat}}} \sum_{c=1}^{C_{\mathrm{lat}}} |\mu_b[c] - \mu_j[c]|. \tag{16}$$

Within-role preselection discounts a candidate when a selected block covers corresponding target samples and has similar latent content. For blocks in the same role, define pairwise replaceability as

$$R_{bj} = \omega_{bj} [\widehat{d}_b^{\top} \widehat{d}_j]_+ \left(1 - \mathrm{clip}_{[0,1]}\left(\frac{m_{bj}}{\tau_{\mathrm{intra}}\ \mathrm{median}_{p\neq q \in \mathrm{role}(b)}\, m_{pq} + \varepsilon}\right)\right), \qquad R_{bb} = 0. \tag{17}$$

The complementarity of b to a set K already selected from its role is

$$\eta_b(K) = 1 - \max_{j \in K} R_{bj}, \qquad \eta_b(\emptyset) = 1. \tag{18}$$

Anchor and retrieved candidates are preselected using

$$\Delta(b \mid K) = \widehat{g}(b \mid K) + \lambda u(b)\eta_b(K), \tag{19}$$

where $\widehat{g}$ is the current-round maximum-normalized geometric marginal gain. Each preselection retains a candidate margin beyond its base quota so that later role allocation can transfer capacity.

After preselection, each anchor or retrieved block is compared with its highest-overlap block $r^*(b)$ in the fully retained recent history. Define

$$c_b = \mathrm{clip}_{[0,1]}\left(\frac{m_{b,r^*(b)}}{\tau_{\mathrm{mae}}\ \mathrm{median}_j\, m_{j,r^*(j)} + \varepsilon}\right) \tag{20}$$

and the estimated redundancy relative to recent history as

$$\xi_b = \omega_{b,r^*(b)}[\widehat{d}_b^{\top}\, \widehat{d}_{r^*(b)}]_+(1 - c_b). \tag{21}$$

Because the anchor and retrieved roles can have different score distributions, redundancy is normalized within each role:

$$\widetilde{\xi}(b) = \mathrm{clip}_{[0,1]}\left(\frac{\xi_b}{p_\kappa(\{\xi_j : j \in \mathrm{role}(b)\}) + \varepsilon}\right), \qquad v(b) = u(b)\big(1 - \widetilde{\xi}(b)\big). \tag{22}$$

Here p_κ denotes a within-role percentile. When its denominator is numerically zero, normalized redundancy for that role is set to zero. Equation 22 instantiates the retention value in Eq. 3: a candidate must be relevant to the target view and provide content not readily represented by recent history.

A.3 Role-wise allocation and physical packing

Let $v_{(i)}^{\mathrm{anc}}$ and $v_{(i)}^{\mathrm{ret}}$ be descending retention values of the preselected anchor and retrieved blocks. Starting from base quotas K_{anc}^0 and K_{ret}^0, Block-wise Context Compression permits a bounded transfer of capacity:

$$\delta^* = \arg\max_{|\delta| \le \delta_{\max}} \left[\sum_{i=1}^{K_{\mathrm{anc}}^0+\delta} v_{(i)}^{\mathrm{anc}} + \sum_{i=1}^{K_{\mathrm{ret}}^0-\delta} v_{(i)}^{\mathrm{ret}}\right]. \tag{23}$$

The search respects each role's capacity and the available candidate counts; ties favor the smaller $|\delta|$. Recent blocks remain fully retained. The selected historical tokens are gathered into a shorter sequence while their original spatiotemporal indices and aligned conditioning are preserved. Unselected history is removed before transformer-block processing and historical key–value construction.

B ACSA Calibration and Sparse Routing

Action-Conditioned Attention Sparsity Allocation uses two independently calibrated quantities: a mapping that allocates current-key access at each layer-head and an interaction-state budget for total visual-key access. At inference, the selected mapping and budget jointly determine current and historical key quotas.

B.1 Interaction-state computation

At each interaction round, the received action specifies the target views $\mathcal{Q}_t$ before generation. The online state rule computes ν_t from those views and the currently available historical views $\mathcal{M}_t$ by Eq. 4, using deterministic geometric samples to approximate FoV coverage. It assigns exploration when $\nu_t \ge \tau_{\mathrm{cov}}$ and revisitation otherwise. The decision uses only the current action and completed history, not future actions or the unseen segment x_t.

B.2 STATE-SPECIFIC HEAD MAPPING

Dense calibration runs aggregate attention mass assigned to current and historical visual keys at each layer-head. Historical mass combines anchor, retrieved, and recent roles. Equation 5 gives the state-specific dense preference $\rho_{l,h}^{(s)}$; it is a calibration coordinate, not a sparse retention ratio.

For each state, we fit a monotonic piecewise-linear mapping

$$f_s(\rho;\ell_s,\mu_s,h_s,W)=\begin{cases}\ell_s, & \rho\le\mu_s-W,\\ \ell_s+\dfrac{\rho-(\mu_s-W)}{2W}(h_s-\ell_s), & \mu_s-W<\rho<\mu_s+W,\\ h_s, & \rho\ge\mu_s+W.\end{cases} \tag{24}$$

Let $\mathcal{V}_s$ be calibration trajectories assigned to state s. For a candidate parameter vector θ_s, dense and sparse branches start from the same noisy latent and receive the same reference latent at each of K_{cal} teacher-forced denoising steps. Their final clean-latent prediction discrepancy is

$$d_{x_0}^{(K_{\mathrm{cal}})}(v;\theta_s)=\mathrm{MSE}\left(x_{0,v,\mathrm{sparse}}^{(K_{\mathrm{cal}})}(\theta_s),x_{0,v,\mathrm{dense}}^{(K_{\mathrm{cal}})}\right). \tag{25}$$

We select

$$\theta_s^*=\arg\min_{\theta_s}\frac{1}{|\mathcal{V}_s|}\sum_{v\in\mathcal{V}_s}d_{x_0}^{(K_{\mathrm{cal}})}(v;\theta_s). \tag{26}$$

Applying $f_s(\rho;\theta_s^*)$ to all layer-head coordinates yields the state-specific tables $\{\alpha_{l,h}^{(E)}\}_{l,h}$ and $\{\alpha_{l,h}^{(R)}\}_{l,h}$. Attention profiling and mapping search are offline; inference only looks up the selected table.

B.3 STATE-DEPENDENT BUDGET CALIBRATION

Holding the head-allocation mapping fixed, we scan the retained visual-attention fraction T separately in the two states. Let $\mathcal{L}_E(T)$ and $\mathcal{L}_R(T)$ be their dense–sparse discrepancy curves, π_E and π_R their calibration frequencies, $C(T)$ the measured attention-compute cost, and T_{ref} the matched uniform operating point. The state-specific operating pair solves

$$(T_E^*,T_R^*)=\arg\min_{T_E,T_R}\left[\pi_E\mathcal{L}_E(T_E)+\pi_R\mathcal{L}_R(T_R)\right] \tag{27}$$

subject to

$$\pi_E C(T_E)+\pi_R C(T_R)\le C(T_{\mathrm{ref}}). \tag{28}$$

The calibrated operating points are converted to visual-key tile budgets B_E and B_R. This budget calibration and the head mapping use separate searches; they are combined at inference.

B.4 PARTITIONED TOP-K EXECUTION

For query tile q and candidate key tile b, we mean-pool their features and compute

$$a_{q,b}^{l,h}=\frac{\langle\bar{q}_q^{l,h},\bar{k}_b^{l,h}\rangle}{\sqrt{d}}. \tag{29}$$

Each current query tile selects the $K_{l,h,s}^{\mathrm{cur}}$ highest-scoring current tiles and the $K_{l,h,s}^{\mathrm{hist}}$ highest-scoring historical tiles according to Eq. 6. Quotas are clipped to available tile counts while preserving access to both visual partitions. The selected visual keys are combined with all text keys; every current visual query continues to produce an output.

C INTERRUPTION-AWARE CACHE REUSE DETAILS

The history signal D_t in Eq. 7 is evaluated on the completed latent frames of x_{t-1} and x_{t-2} after their final scheduler updates. It does not inspect an intermediate denoising state of the segment being generated. This completed-history signal determines the base reuse level, while the known action can only reduce that level before generating x_t.

C.1 MULTI-DEPTH FOV TRANSPORT AND INTERRUPTION

Let $\mathcal{Q}_t$ and $\mathcal{Q}_{t-1}$ contain the planned camera views of the target and preceding segments. For target view $q \in \mathcal{Q}_t$, we sample a ray u and instantiate a point at hypothetical depth d. Projecting that point into source view $s \in \mathcal{Q}_{t-1}$ gives normalized angular coordinates $\bar{\mathbf{p}}_s(q, u, d)$; the target ray has normalized coordinates $\bar{\mathbf{p}}_q(u)$. The transport cost is

$$c(q, s, u, d) = \begin{cases} \min\{1, \|\bar{\mathbf{p}}_s(q, u, d) - \bar{\mathbf{p}}_q(u)\|_2\}, & \text{valid source projection,} \\ 1, & \text{otherwise.} \end{cases} \tag{30}$$

The transport score is

$$G_t = \mathbb{E}_{q,u,d}\left[\min_{s \in \mathcal{Q}_{t-1}} c(q, s, u, d)\right]. \tag{31}$$

Taking the nearest source view measures whether the preceding segment offers a geometrically close correspondence for each planned sample. Multiple hypothetical depths capture translation-induced parallax without predicted scene depth. The controller compares G_t with G_{t-1} because D_t describes the preceding transition. Consistent with Eqs. 9 and 10 and Figure 1, the interruption indicator is

$$g_t = \mathbf{1}[\Delta G_t^+ > \epsilon], \qquad \ell_t = \max(0, \ell_t^{\text{base}} - g_t). \tag{32}$$

C.2 STEP-SELECTIVE REUSE

The final level ℓ_t selects which eligible intermediate DiT evaluations are reused. The first evaluation establishes the current denoising trajectory, and the final evaluation supplies a correction; both are recomputed. A reused step takes a model prediction cached from a preceding *computed step of the same segment*, while the scheduler still advances at that step. The action interruption can only replace a planned reuse with a computation. Exact eligible steps and cache updates for the evaluated samplers are given below.

D IMPLEMENTATION DETAILS

D.1 CONTEXT COMPRESSION IMPLEMENTATION

HY-WorldPlay. In HY-WorldPlay, anchor, retrieved, and recent history each supply a four-frame tile. A 30×52 latent grid partitioned into $4 \times 8 \times 4$ blocks gives 52 blocks per tile and 156 in total. Bottom-boundary blocks have 96 valid positions rather than 128; padding is excluded from descriptor pooling. Four target views, 8×26 rays per view, and depths $\{2, 4, 7\}$ yield 2,496 samples. The correspondence radius is one block.

From the 65-channel image-projection input, descriptors use only the first 32 VAE-latent channels, excluding 32 conditioning channels and the task mask. Thus $C_{\text{lat}} = 32$ and $d_b \in \mathbb{R}^{64}$ (Table 6).

Table 6: HY-WorldPlay Block-wise Context Compression implementation settings.

Setting	Value
Block shape	$4 \times 8 \times 4$
Target ray grid	8×26 per target view
Hypothetical depths	$\{2, 4, 7\}$
Matching neighborhood radius	1 block
Descriptor	32-channel latent mean/std
Redundancy percentile κ	90
$\tau_{\text{intra}}, \tau_{\text{mae}}$	$2, 2$
$\lambda, \delta_{\max}$	$1, 6$
Geometric-gain normalization	value normalization
Fully retained role	recent history
Selector placement	first-block pre-hook

The full system uses anchor/retrieved/recent base quotas of $26/39/52$ for chunks 4–7 and $26/52/52$ for chunks 8–15, or 117 and 130 historical blocks. The matched-average selector study in Table 5

uses its stated state-conditioned budgets instead. Preselection keeps at most $K_r^0 + \delta_{\max}$ candidates per compressible role, subject to the 52-block role capacity.

At the first-block pre-hook, retained tokens and their rotary/positional indices, camera parameters, and conditioning tensors are gathered by original dense identifiers. Discarded history misses later transformer blocks and historical KV construction, though image-input projection still runs. Reported latency includes projection, descriptors, selection, and host/device transfer.

Matrix-Game 3.0. Matrix-Game 3.0 conditions each generation segment of 10 frames on nine historical frames, organized as one anchor, four retrieved, and four recent frames. Unlike the four-frame tiles in HY-WorldPlay, these frames do not admit the same semantically aligned block-wise partitioning. We therefore apply context compression at the frame level, retaining one anchor, two retrieved, and three recent frames, reducing the historical context from nine to six frames.

D.2 SPARSE ATTENTION

HY-WorldPlay. The interaction-state threshold is $\tau_{\text{cov}} = 0.02$. For fixed-trajectory runs, ν_t is computed from the planned target views according to Eq. (4), and the interaction state is classified as exploration or revisitation using $\tau_{\text{cov}} = 0.02$. Dense calibration yields 864 layer-head coordinates per state. Mapping calibration uses four teacher-forced steps ($K_{\text{cal}} = 4$), with $W = 0.10$ and

$$(\ell_E, \mu_E, h_E, W) = (0.25, 0.30, 1.0, 0.10), \qquad (\ell_R, \mu_R, h_R, W) = (0.50, 0.40, 1.0, 0.10). \quad (33)$$

The resulting current-retention tables are $\{\alpha_{l,h}^{(E)}\}_{l,h}$ and $\{\alpha_{l,h}^{(R)}\}_{l,h}$.

The separate budget scan uses four-step teacher-forced dense–sparse x_0 MSE. Against $T_{\text{ref}} = 0.37$, it selects $T_E = 0.40$ and $T_R = 0.34$: measured loads of 1269.4 and 1100.3 visual blocks per layer. Dividing by 16 heads gives 79.3 and 68.8 blocks per head, rounded to deployed budgets

$$B_E = 79, \qquad B_R = 69. \quad (34)$$

These are matched-calibration operating points. Attention tiles have shape $4 \times 8 \times 4$ and 128 regular visual tokens. Runtime quotas are bounded by available tiles; text keys remain fully accessible.

Matrix-Game 3.0. Matrix-Game 3.0 follows the same state-conditioned sparse-attention procedure. At each interaction round, the exploration/revisitation state is determined from the action-derived target views using Eq. (4), and the corresponding state-specific attention policy is applied. The layer-head retention mappings and sparse-attention budgets are calibrated separately for Matrix-Game 3.0 to account for its model-specific attention structure and context organization, rather than reusing the HY-WorldPlay calibration.

D.3 CACHE REUSE IMPLEMENTATION

HY-WorldPlay. The history MSE uses the last four latent frames after the final scheduler updates of the two preceding chunks. The first three chunks are computed in full. The reported configuration uses $\tau_1 = 0.75$ and $\tau_2 = 1.6968797445297241$ in Eq. 8.

FoV transport uses horizontal/vertical angles of $105°/75°$, a 24×14 angular grid spanning 95% of each FoV half-angle, and depths $\{2, 4, 8, 16\}$. Out-of-view or behind-camera projections cost one. Positive transport increments no larger than 10^{-9} are set to zero; thus the interruption in Eq. 10 uses $\epsilon = 10^{-9}$.

For the four-step sampler, the final level selects the schedule in Table 7.

Table 7: Four-step Interruption-aware Cache Reuse schedule. Reuse skips the DiT evaluation but retains the scheduler update.

Level	$s = 0$	$s = 1$	$s = 2$	$s = 3$	DiT calls
$\ell = 0$	Compute	Compute	Compute	Compute	4
$\ell = 1$	Compute	Compute	Reuse	Compute	3
$\ell = 2$	Compute	Reuse	Reuse	Compute	2

At $\ell = 2$, steps $s = 1, 2$ reuse the computed $s = 0$ prediction. At $\ell = 1$, step $s = 2$ reuses the computed $s = 1$ prediction. The cached quantity is the completed transformer prediction; skipped steps still update the scheduler. The final step is recomputed, and interruption can only lower the reuse level by one.

Matrix-Game 3.0. In the three-step Matrix-Game 3.0 implementation, only the middle step ($s = 1$) can be reused; the first and last DiT evaluations are recomputed. Once two completed chunks are available, their final latent variation controls this binary choice. The earliest eligible chunk falls back to the backbone's FoV criterion. This adaptation does not use the multi-depth ΔG_t^+ interruption rule.