WORLDRETAIN: PREDICTIVE RETENTION AND SPATIAL COMPRESSION FOR EFFICIENT AND CONSISTENT VIDEO WORLD MODEL INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive video world models must preserve scene identity across viewpoint revisits while limiting the cost of historical context. Sliding-window inference can discard evidence needed for later returns, and without an external memory bank, this eviction is irreversible. We propose WorldRetain, an inference-time cache policy that combines predictive KV retention with spatial compression. A lightweight predictor learns from future attention and viewpoint reuse to rank historical chunks, using only observed camera geometry and resident cache features at inference. Selected history is compressed into salient tokens and pooled summaries, maintaining a compact native cache without an external bank or changes to generator weights. On 100 RealCam-Vid camera-loop trajectories per backbone, WorldRetain achieves the best scene-identity and appearance consistency among the evaluated methods on LingBot-World-Fast and Matrix-Game 2.0. Compared with sliding-window inference, same-place scores increase from 1.77 to 2.29 and from 1.06 to 2.71, respectively, alongside 1.37–1.43$\times$ throughput speedups and lower peak GPU memory.

1 INTRODUCTION

Autoregressive video world models generate interactive environments by extending a video in response to camera or action controls (Yin et al., 2025; Huang et al., 2026; He et al., 2026; Team et al., 2026). Their usefulness depends on the consistency of the generated world and the cost of maintaining it. Frames should remain visually coherent, previously visited places should remain recognizable, and inference should sustain interactive throughput without accumulating excessive memory. These requirements meet in the KV cache: historical tokens provide visual context, but also contribute to storage and attention cost.

A longer cache does not necessarily provide the most useful context. Recent frames support local continuity, yet a camera can move away from a place long before returning to it. A sliding window may discard that place while keeping many similar recent views. Pose-indexed retrieval can recover historical context from a memory bank (Yi et al., 2026), but reducing the active attention window alone does not bound the storage of that bank or account for retrieval overhead. Whether a memory policy improves efficiency therefore depends on its measured throughput and memory use, as well as the consistency of its generated revisits. Figure 1 illustrates the visual consequences of these history-management choices on LingBot-World-Fast. A corresponding Matrix-Game 2.0 example is reported in Appendix C.

We introduce WorldRetain, an inference optimization framework that allocates a compact KV cache using predicted future-use scores for visual history. Its central principle is to retain useful places while reducing the spatial cost of representing them. This leads to two complementary operations. *Predictive history retention* learns to rank resident chunks by their predicted future-use scores, rather than only their relevance to the current camera. A place outside the current view can therefore receive a high retention score when the observed trajectory and cache suggest that it may be useful again. *Spatial compression* reduces the token footprint of the selected history, so preserving a useful place removes the requirement to retain its full spatial grid.

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.

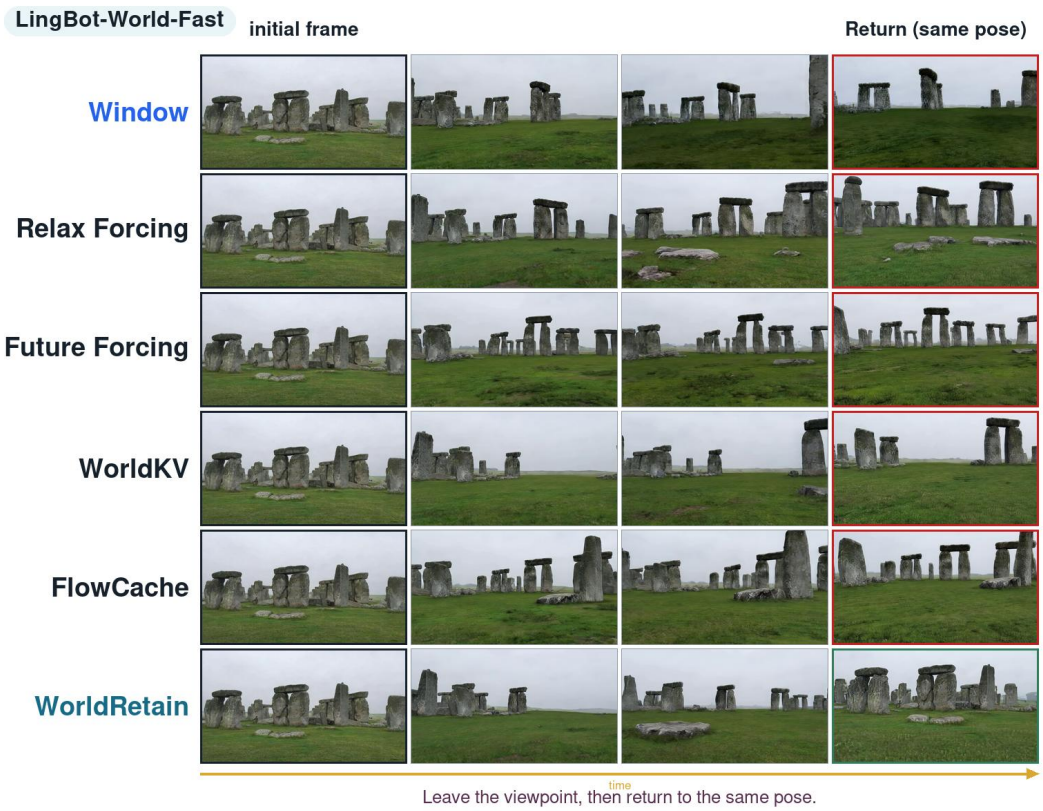


Figure 1: Camera-loop generation on LingBot-World-Fast. Each row follows the same camera trajectory from an initial view through departure to a return at the same pose. In this example, WorldRetain better preserves the initial scene layout at the return. Quantitative consistency and inference costs are reported in Figure 3 and Table 1.

The predictor is trained offline using future camera reuse and attention as supervision; at inference, it uses only observed poses and resident cache features. The selected and compressed history forms the context for subsequent attention, connecting retention decisions to inference cost. Evaluation on LingBot-World-Fast and Matrix-Game 2.0 tests applicability across different model sizes and cache configurations.

Our contributions are threefold. (i) We introduce a shared inference optimization framework for pose-aware autoregressive video world models, targeting revisit consistency and inference efficiency through native KV management. (ii) Predictive history retention coupled with spatial compression separates the decision of which history remains useful from the cost of representing that history. (iii) On 100 LingBot and 100 Matrix-Game 2.0 RealCam-Vid loops, the method improves same-place consistency, PSNR, and FID over sliding-window inference on both backbones while achieving throughput speedups of $1.37\times$ and $1.43\times$, respectively. Figure 3 reports the measured consistency–speed and consistency–memory trade-offs; LingBot ablations examine the supervision target and archive representation (Section 4.3).

2 Related Work

2.1 Autoregressive video world models

Causal video generators stream frames with a KV cache instead of jointly denoising a clip (Yin et al., 2025; Huang et al., 2026; Liu et al., 2026; Yang et al., 2025). Interactive systems attach camera or action controls to that backbone (He et al., 2026; Li et al., 2025; Mao et al., 2026; Team et al., 2026).

The question is which native KV blocks should remain when a bounded cache can no longer hold the full history, so that a later camera return still depicts the same place.

2.2 Retrieval memory

Several world-model memories outlive a sliding window by storing the past outside the native attention window. WorldMem and WorldPlay retrieve past frames as conditioning (Xiao et al., 2026; Sun et al., 2025). VMem uses geometry-grounded spatial memory (Wu et al., 2025), while latent spatial memory stores scene information in a persistent 3D latent cache (Wang et al., 2026). RELIC compresses historical latents inside the cache (Hong et al., 2025); WorldTrace studies whether compressed KV remains addressable under rotary embeddings (Wu et al., 2026).

WorldKV banks evicted KV and retrieves pose-matched chunks at revisit time (Yi et al., 2026). Closing the Loop retrieves historical latent chunks and uses pose-and-depth correspondences to guide attention during revisits (Ma et al., 2026a). Such retrieval can restore context after it leaves the active window. WorldRetain instead retains history within the native cache without an external bank, making keep-or-drop decisions before eviction, when discarded context is still available.

2.3 KV Cache Management

Language-model caches keep sinks or high-attention tokens under a fixed budget (Xiao et al., 2024; Zhang et al., 2023). Video methods reduce native context through sparsification, temporal compaction, and deep sinks (Li et al., 2026; Yi et al., 2025; Samuel et al., 2026; Meng et al., 2026). FlowCache combines chunkwise feature reuse with fixed-budget KV selection based on current attention importance and key redundancy (Ma et al., 2026b).

Relax Forcing partitions memory into Sink, History, and Tail, selecting History frames with a relaxation score that rewards key similarity to Sink anchors and penalizes redundancy with recent Tail frames to balance stability and motion evolution (Zhao et al., 2026). Future Forcing estimates future query relevance from historical statistics (Luo et al., 2026). WorldRetain learns retention scores from observed camera geometry and resident cache features, retaining and compressing history to support viewpoint revisits within a bounded native cache.

3 Method

WorldRetain optimizes the historical KV context used by an autoregressive video generator. It combines two operations: *predictive retention* selects history that may support future views, and *spatial compression* reduces the token cost of the selected history (Figure 2). LingBot-World-Fast and Matrix-Game 2.0 use the same retention algorithm, learner design, and compression procedure, adapted to their native KV interfaces. The method requires historical KV chunks and their observed camera poses; a small retention predictor is trained offline.

3.1 Cache optimization formulation

A generator emits latent chunks $c_1, c_2, \ldots$. Chunk i contributes KV tensors (K_i, V_i) and has an associated observed camera pose $T_i = (R_i, p_i) \in \mathrm{SE}(3)$. After the cache update at step t, let $\mathcal{A}_t$ denote the resident chunk indices and $(\widetilde{K}_{t,i}, \widetilde{V}_{t,i})$ their stored representations, containing $\widetilde{n}_{t,i}$ tokens. The next chunk uses

$$\mathbf{K}_t = \mathrm{concat}_{i \in \pi_t} \widetilde{K}_{t,i}, \qquad \mathbf{V}_t = \mathrm{concat}_{i \in \pi_t} \widetilde{V}_{t,i}, \qquad \sum_{i \in \mathcal{A}_t} \widetilde{n}_{t,i} \le B, \tag{1}$$

where π_t gives the backbone's attention order and B bounds the retained KV tokens. Key positional encoding is applied when a chunk enters the cache, before any archive compression. The two decisions are the resident set $\mathcal{A}_t$ and its stored representations. Selection aims to preserve context useful for future generation; compression reduces its spatial footprint. The token bound applies after each update, rather than to total peak GPU memory, which also includes model weights, activations, and temporary buffers. We measure memory and throughput directly in the experiments.

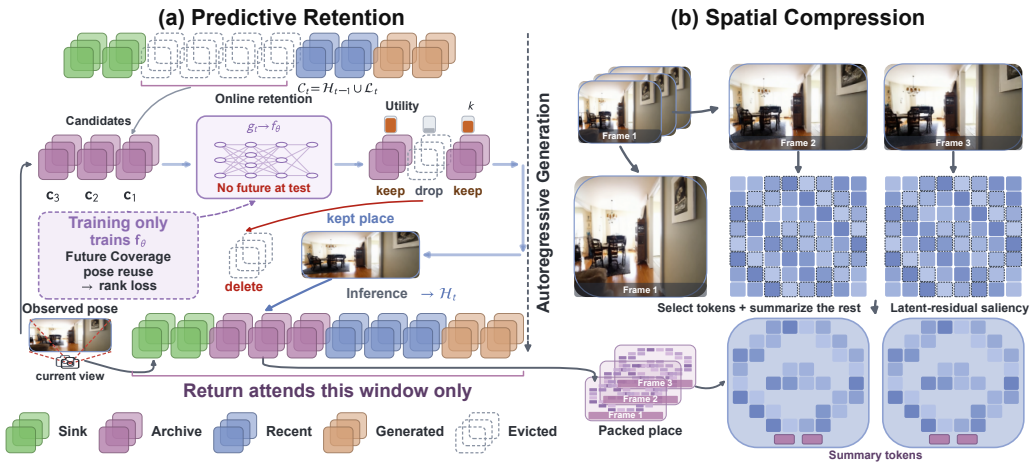


Figure 2: WorldRetain combines history selection and spatial compression within the generator's native KV cache. (a) Predictive retention scores candidates $\mathcal{C}_t$ with f_θ using observed poses and cache features. The retained archive $\mathcal{H}_t$ is combined with protected initial and recent context. (b) Spatial compression preserves selected spatial details and pooled summaries, with gist-only representations for lower-ranked history. Subsequent generation attends to the resulting compact cache.

For stable initial and local context, we retain a small protected initial set $\mathcal{S}_t$ and a recent window $\mathcal{R}_t$. The remaining history forms a selected archive $\mathcal{H}_t$:

$$\mathcal{A}_t = \mathcal{S}_t \uplus \mathcal{R}_t \uplus \mathcal{H}_t. \tag{2}$$

Here $\uplus$ denotes disjoint union: the initial, recent, and archive sets are pairwise disjoint. The two core operations act on the archive. Let $\mathcal{L}_t$ denote chunks leaving the recent window at step t. As shown in Figure 2, they join the surviving archive to form the candidate set:

$$\mathcal{L}_t = \mathcal{R}_{t-1} \setminus \mathcal{R}_t, \qquad \mathcal{C}_t = \mathcal{H}_{t-1} \cup \mathcal{L}_t, \qquad \mathcal{H}_0 = \emptyset. \tag{3}$$

Only resident history can be selected. Dropped chunks are released and cannot re-enter a later candidate set.

3.2 Predictive History Retention

Current relevance can underestimate future usefulness: a chunk depicting a place behind the camera may become valuable when the camera returns. We therefore learn a retention score from supervision over future views. For each candidate $i \in \mathcal{C}_t$, a compact MLP computes the retention score $u_t(i)$, labeled *Utility* in Figure 2:

$$u_t(i) = f_\theta\big(g_t(i)\big), \tag{4}$$

where $g_t(i)$ is a fixed-order 11-dimensional descriptor of candidate i at the current step (Table 4): scaled geometry, motion and reachability, layer-0 key and value summaries, and the count and recency of geometric alignment. Every coordinate is available at step t. The predictor standardizes each coordinate using training-set statistics and applies two 32-unit hidden layers with GELU activations and a scalar output layer, totaling 1,473 trainable parameters. Both backbones use the same predictor architecture, training configuration, and sample counts, with teacher rollouts collected from their respective generators on training clips disjoint from evaluation. Implementation details are given in Appendix E.

Learning retention scores. We construct a *Future Coverage* target from offline teacher rollouts. Let $\widetilde{A}_{t+h}(i)$ measure the attention assigned to chunk i relative to a uniform-attention baseline, and let $\mathrm{PoseReuse}_{t+h}(i)$ measure its geometric agreement with the future camera. For horizon H, the target is

$$y_{t,i} = \sum_{h=1}^{H} \gamma^{h-1} \left[\alpha\, \widetilde{A}_{t+h}(i) + (1-\alpha)\, \mathrm{PoseReuse}_{t+h}(i)\right]. \tag{5}$$

The attention term reflects the teacher's use of stored visual context; the pose term supplies a geometric proxy for later reuse. We use $\widetilde{A} = A/(A + m/L)$, where A is the chunk's attention mass, m its key-token count, and L the total key-token count in the same attention distribution. This bounded score accounts for chunk size and context length. Detailed definitions are given in Appendix E. Future Coverage provides a supervision signal for ranking anticipated future use, rather than an estimate of the marginal generation loss caused by removing a chunk. Each ranking group comprises the candidates at one generation step. Let $\mathcal{P}_t = \{(i,j) \in \mathcal{C}_t^2 : y_{t,i} \neq y_{t,j}\}$ denote the valid comparison pairs. We minimize the mean pairwise loss:

$$\mathcal{L}_{\mathrm{rank}}(t) = \frac{1}{|\mathcal{P}_t|} \sum_{(i,j)\in\mathcal{P}_t} \mathrm{softplus}\big(-\mathrm{sign}(y_{t,i} - y_{t,j})(u_t(i) - u_t(j))\big). \tag{6}$$

Groups with no valid comparison pairs are excluded. Future poses and attention are used only to construct training labels. Online scoring uses observed poses and resident cache features, without access to the future camera trajectory.

Online selection. Given an archive capacity k_t (the Top-k step in Figure 2), we retain the highest-scoring candidates:

$$\widehat{\mathcal{H}}_t = \mathrm{TopK}_{k_t}\big(\{u_t(i)\}_{i\in\mathcal{C}_t}\big). \tag{7}$$

The capacity rule is fixed for a backbone and shared by its scoring variants; its keep-ratio implementation is given in Appendix E. This step determines which places are represented before compression and the final token-budget check. It allows a currently distant candidate to survive when its retention score exceeds that of other residents. The supervision ablation compares future pose reuse alone, future attention alone, and their combination, keeping the predictor inputs, architecture, and cache policy fixed (Section 4.3).

3.3 Spatial compression and online inference

Every chunk enters the cache at full resolution. Compression begins only when it leaves the recent window and is selected for the archive; protected initial and recent chunks remain at full resolution. Within the archive, we preserve selected spatial details and summarize the remaining content.

Residual-guided packing. Let X_i be the denoised latent tensor of chunk i. We compute token saliency at insertion from adjacent chunks:

$$s_i = \mathrm{vec}\left[\mathrm{AvgPool}_{2\times 2}\left(\frac{1}{C}\sum_{c=1}^{C} |X_i^{(c)} - X_{i-1}^{(c)}|\right)\right], \tag{8}$$

where C is the latent channel count and pooling aligns the residual with the spatial token grid. This is a proxy for local visual change, not an attention score or a predicted future-use score. For a chunk with n_i tokens, packing keeps at most $\lceil \rho n_i \rceil$ original tokens, selecting the smallest saliency-ranked prefix that covers a target fraction η of total saliency, subject to this cap. A uniform lattice replaces top-ranked selection when saliency is unavailable or nearly uniform. The omitted positions are grouped into spatial cells and their KV vectors are mean-pooled into at most m summary tokens:

$$(K_i', V_i') = \Pi(K_i, V_i; s_i), \qquad n_i' = k_i + q_i, \quad k_i \le \lceil \rho n_i \rceil, \quad q_i \le m, \tag{9}$$

where k_i and q_i count retained original tokens and summaries. We fix the per-chunk L1 summary cap at $m = 64$ in all experiments on both backbones. Thus, the retention ratio bounds original tokens, not the total packed length. The same saliency is shared across transformer layers. Both backbones use the same summary-KV construction procedure. Packing averages keys after their absolute-coordinate RoPE has been applied. Retained original keys keep their phases; summaries are used directly in attention without assigning a new position or applying RoPE again. Details are given in Appendix E.1.

Compact archive and inference. The default archive also represents lower-ranked retained chunks using only pooled summaries (*gists*), under a separate total gist-token budget. Packing and gist are two representations within spatial compression: one preserves selected details, while the

Algorithm 1 WorldRetain: shared cache update after chunk t

Require: Resident KV and poses through step t; previous archive $\mathcal{H}_{t-1}$; predictor f_θ; budget B
Ensure: Updated native KV with at most B retained tokens

Store chunk t's full-resolution KV, selector features, and residual saliency
Update protected initial and recent sets $\mathcal{S}_t, \mathcal{R}_t$
$\mathcal{L}_t \leftarrow \mathcal{R}_{t-1} \setminus \mathcal{R}_t$
$\mathcal{C}_t \leftarrow \mathcal{H}_{t-1} \cup \mathcal{L}_t$
$u_t(i) \leftarrow f_\theta(g_t(i))$ for each $i \in \mathcal{C}_t$ ▷ predictive retention
$\widehat{\mathcal{H}}_t \leftarrow \mathrm{TopK}_{k_t}(\{u_t(i)\}_{i \in \mathcal{C}_t})$
Compress newly promoted archive chunks by rank into packed or gist form; apply gist downgrades without restoring removed tokens
Remove the weakest gist chunks if their total budget is exceeded
Remove these discarded chunks from $\widehat{\mathcal{H}}_t$
$\mathcal{H}_t \leftarrow \widehat{\mathcal{H}}_t$
while tokens in $\mathcal{S}_t \uplus \mathcal{R}_t \uplus \mathcal{H}_t$ exceed B **do**
    Remove the lowest-scoring chunk from $\mathcal{H}_t$
Release discarded KV; form $\mathcal{A}_t \leftarrow \mathcal{S}_t \uplus \mathcal{R}_t \uplus \mathcal{H}_t$
Return the native cache in the backbone's attention order

other retains only coarse context. Already packed chunks are not packed repeatedly; any later conversion to gist uses their currently stored KV. Appendix E.1 specifies token allocation, pooling, and representation transitions. All summaries count toward the cache budget in Eq. (1). If the resulting cache still exceeds B, the lowest-scoring archive chunks are removed until the bound holds. The final archive $\mathcal{H}_t \subseteq \widehat{\mathcal{H}}_t$ is then combined with protected initial and recent context for subsequent generation. We require B to accommodate the protected context. Algorithm 1 summarizes this update. Backbone-specific cache interfaces and protected-window settings are described in Appendix E.

4 Experiments

4.1 Experimental Setup

Benchmark. To evaluate revisit consistency and inference cost, we construct camera-loop trajectories from the camera annotations provided by RealCam-Vid (Zheng et al., 2025), using 100 clips on each backbone. Each evaluation trajectory has 481 RGB frames. The trajectories revisit previously observed viewpoints, and all methods share the same evaluation pairs on a given backbone. Pairing details are provided in Appendix A.

Base models. LingBot-World-Fast (Team et al., 2026) is a 14B camera-conditioned generator distilled for long rollouts, with a long native cache. Matrix-Game 2.0 (He et al., 2026) is a 1.3B real-time model whose official inference uses a 6-frame sliding window. Both emit three-frame latent chunks. We run the same cache policies on both frozen generators. All runs use eight NVIDIA H20 GPUs. LingBot uses two GPUs per clip, except WorldKV, which uses four; Matrix-Game 2.0 uses one GPU per clip. Implementation details are provided in Appendix E.

Baselines. For each backbone we compare six inference-time cache policies. *Window* is FIFO sliding KV, the practical native default. *Relax Forcing* is temporal KV sparsification (Zhao et al., 2026), matched on both backbones. *Future Forcing* is a training-free cache policy that ranks KV tokens by predicted future attention using a query proxy derived from historical statistics (Luo et al., 2026). *WorldKV* is the official pose-indexed retrieval method with store-time compression (Yi et al., 2026). *FlowCache* combines chunkwise feature caching with importance–redundancy-based KV compression (Ma et al., 2026b). *WorldRetain* follows Algorithm 1. We exclude methods that require backbone training or fine-tuning, such as WorldPack (Oshima et al., 2026), to isolate inference-time cache optimization on fixed generators.

Metrics. Qwen3-VL-32B-Instruct (Bai et al., 2025), GPT-5.6-Sol, and Gemini 3.6 Flash independently score *same-place* in $\{0, 1, 2, 3\}$ from condition, first-visit, and revisit frames; we average their scores (Appendix D). PSNR and LPIPS (Zhang et al., 2018) average over paired generated visits; FID (Heusel et al., 2017) compares the feature distributions of the two generated frame sets. These

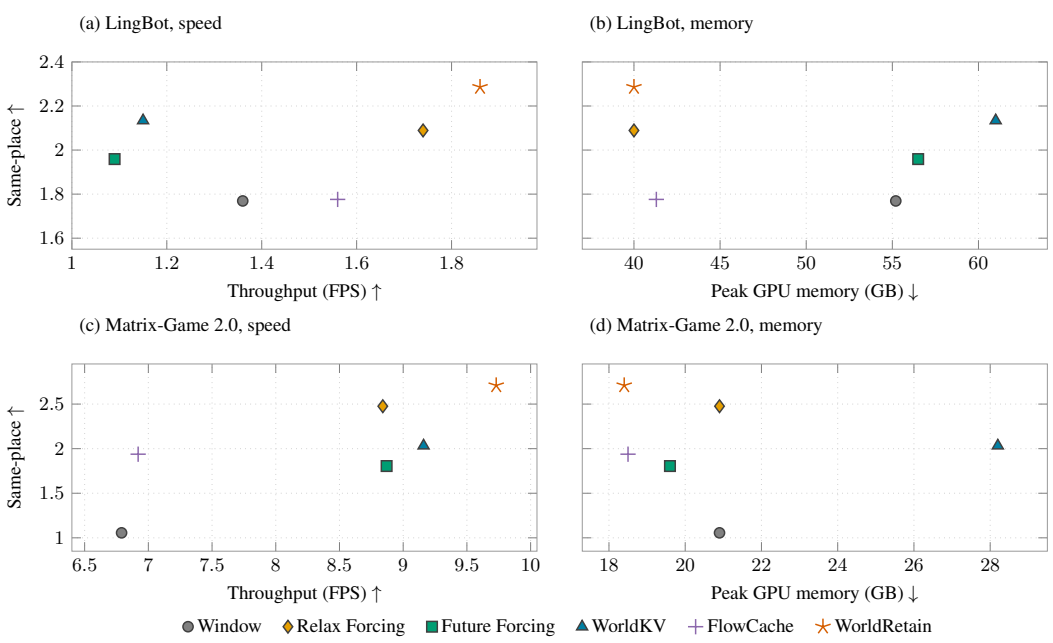


Figure 3: Consistency–efficiency trade-offs on RealCam-Vid loops: 100 on LingBot-World-Fast (top) and 100 on Matrix-Game 2.0 (bottom). Same-place consistency versus throughput (left; upper-right is better) and peak GPU memory (right; upper-left is better).

measure revisit consistency, not fidelity to real frames. Efficiency includes mean active KV tokens, peak GPU memory, and FPS: decoded RGB frames per synchronized wall-clock second, averaged equally across clips rather than taken from the video's playback rate.

4.2 Main Results

Figure 3 and Table 1 evaluate the shared method on both backbones, reporting scene-identity and appearance consistency across revisits alongside measured throughput and memory. Relative to Window, WorldRetain raises same-place from 1.77 to 2.29 on LingBot and from 1.06 to 2.71 on Matrix-Game 2.0. Throughput increases from 1.36 to 1.86 FPS (1.37×) and from 6.79 to 9.73 FPS (1.43×), respectively. Consistency and throughput therefore improve together on both backbones, without a larger active context.

Scene-identity and appearance consistency. WorldRetain achieves the highest same-place score on both backbones. Relative to Window, PSNR increases from 10.65 to 11.20 dB on LingBot and from 17.20 to 18.28 dB on Matrix-Game 2.0; FID decreases from 30.2 to 26.4 and from 92.6 to 17.9, respectively. LPIPS decreases from 0.653 to 0.637 on LingBot and from 0.430 to 0.361 on Matrix-Game 2.0. These results indicate stronger pixel-level, perceptual, and distribution-level agreement between generated visits (Section 4.1). WorldRetain also improves same-place over all compared baselines on both backbones. On LingBot, WorldRetain also records the lowest FID among compared methods (26.4 versus 27.0 for Relax Forcing). Clip-level paired comparisons with baseline methods confirm significant same-place improvements on both generators (Wilcoxon signed-rank tests with Holm correction, $p < 0.05$); details are provided in Appendix B.

Inference efficiency. Active KV decreases from 41.5k to 15.1k tokens on LingBot and from 29.8k to 5.6k on Matrix-Game 2.0 relative to Window. Peak GPU memory decreases from 55.2 to 40.0 GB on LingBot, a 27.5% reduction, and from 20.9 to 18.4 GB on Matrix-Game 2.0, a 12.0% reduction. Relative to WorldKV, at the reported operating points, peak memory decreases from 61.0 to 40.0 GB on LingBot and from 28.2 to 18.4 GB on Matrix-Game 2.0, while throughput increases from 1.15 to 1.86 FPS and from 9.16 to 9.73 FPS. The memory plots make clear that fewer KV tokens do not imply a proportional reduction in total GPU memory. On these two backbones, the measurements support an improved consistency–efficiency trade-off at the reported operating points.

Table 1: Main results on 100 RealCam-Vid loops per backbone. Best results within each backbone are bold.

	Revisit consistency				Efficiency		
Method	SP↑	PSNR↑	LPIPS↓	FID↓	FPS↑	KV (k)↓	GB↓
(a) LingBot-World-Fast							
Window	1.77	10.65	0.653	30.2	1.36	41.5	55.2
Relax Forcing	2.09	10.90	0.642	27.0	1.74	17.4	**40.0**
Future Forcing	1.96	10.84	0.645	28.3	1.09	41.5	56.5
WorldKV	2.13	10.89	0.641	27.7	1.15	39.0	61.0
FlowCache	1.78	10.68	0.650	30.2	1.56	25.4	41.3
WorldRetain	**2.29**	**11.20**	**0.637**	**26.4**	**1.86**	**15.1**	**40.0**
(b) Matrix-Game 2.0							
Window	1.06	17.20	0.430	92.6	6.79	29.8	20.9
Relax Forcing	2.48	17.16	0.395	24.9	8.84	10.2	20.9
Future Forcing	1.81	16.54	0.438	47.5	8.87	6.9	19.6
WorldKV	2.03	17.44	0.380	30.8	9.16	12.4	28.2
FlowCache	1.94	17.42	0.398	39.8	6.92	7.7	18.5
WorldRetain	**2.71**	**18.28**	**0.361**	**17.9**	**9.73**	**5.6**	**18.4**

4.3 Ablation Study

We examine the supervision used for predictive retention and the representation used for spatial compression on LingBot-World-Fast. Both studies use the 100 LingBot RealCam-Vid loops from Section 4.1, with identical initial conditions, camera trajectories, and evaluation pairs across variants. We report same-place (SP) and pairwise PSNR and LPIPS, together with distribution-level FID, active KV tokens, peak GPU memory, and throughput. Efficiency measurements use the same hardware and timing protocol as the main experiments and include cache scoring and compression overhead.

Table 2: Ablation studies on LingBot-World-Fast.

	Revisit consistency				Efficiency		
Method	SP↑	PSNR↑	LPIPS↓	FID↓	FPS↑	KV (k)↓	GB↓
(a) Predictive Retention							
Future-pose-only	1.95	10.49	0.677	39.0	1.84	17.0	40.0
Future-attention-only	1.88	10.50	0.677	39.3	1.85	15.0	40.0
Future Coverage	**2.29**	**11.20**	**0.637**	**26.4**	**1.86**	15.1	40.0
(b) Spatial Compression							
No packing	1.77	10.34	0.680	39.2	1.80	**15.1**	41.2
Uniform	1.85	10.45	0.677	37.8	1.83	**15.1**	**40.0**
Proposed	**2.29**	**11.20**	**0.637**	**26.4**	**1.86**	**15.1**	**40.0**

Predictive Retention. Table 2(a) varies the supervision target while fixing the 11-dimensional inputs, MLP architecture, training protocol, and cache configuration. In Eq. (5), *Future-pose-only* uses $\alpha = 0$, *Future-attention-only* uses $\alpha = 1$, and *Future Coverage* uses the default combination $\alpha = 0.5$; all use the same horizon and discount factor. Future Coverage raises same-place to 2.29, compared with 1.95 for pose-only and 1.88 for attention-only supervision. These results support combining geometric reuse and attention signals for retention ranking. Figure 4(a) illustrates the same supervision comparison; its *Learned* row denotes Future Coverage.

Spatial compression. Table 2(b) fixes the learned scorer and varies only the retained representation: *No packing* keeps full spatial tokens, *Uniform* uses a uniform lattice, and *Proposed* applies

the saliency-guided operator in Section 3.3. Proposed raises same-place from 1.77 (no packing) and 1.85 (uniform) to 2.29 at the same reported mean active KV size of 15.1k tokens. It reaches 1.86 FPS while matching the lowest peak memory (40.0 GB). Figure 4(b) illustrates the corresponding revisit behavior.

5 Limitations & Future Work

Limitations. We evaluate revisits on two pose-aware generators; performance on other model families, open-world rollouts, and noisy or estimated poses remains untested. WorldRetain requires access to historical KV and offline predictor training. Future Coverage supervises future-use ranking through a proxy, not the measured effect of retaining each chunk on generation.

Future work. We plan to test longer and less structured trajectories, including competing revisits and dynamic scenes, and measure sensitivity to pose errors. Interventions that remove individual chunks could assess how well the ranking predicts their contribution to later revisits.

6 Conclusion

WorldRetain improves the consistency–efficiency trade-off of autoregressive video world model inference by combining predictive history retention with spatial compression. It preserves useful visual history in a compact native KV cache. On LingBot-World-Fast and Matrix-Game 2.0, it improves scene-identity and appearance consistency across revisits over sliding-window inference while increasing throughput and reducing peak GPU memory on both backbones. These results highlight the value of considering both which history to retain and how compactly to represent it.



Figure 4: Qualitative ablations on LingBot-World-Fast. (a) Predictive Retention supervision with fixed spatial compression; Learned denotes Future Coverage. (b) Spatial Compression with fixed predictive retention. Columns show the first visit, departure, and return to the same pose. Each panel uses the same initial condition and camera trajectory across variants.

AI USE STATEMENT

We used large language models to brainstorm research ideas, obtain feedback on the method and experimental design, and assist in interpreting results. They also helped translate, draft, and polish manuscript text and retrieve and summarize related work. AI-assisted code editors, including Cursor, were used to implement and edit research code. The authors reviewed the AI-assisted text, code, analyses, and citations, verified the reported claims and results, and take responsibility for the final submission.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

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, Size Wu, Wei Li, Xuchen Song, Yang Liu, Yangguang Li, and Yahui Zhou. Matrix-game 2.0: An open-source real-time and streaming interactive world model, 2026. URL https://arxiv.org/abs/2508.13009.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Yicong Hong, Yiqun Mei, Chongjian Ge, Yiran Xu, Yang Zhou, Sai Bi, Yannick Hold-Geoffroy, Mike Roberts, Matthew Fisher, Eli Shechtman, Kalyan Sunkavalli, Feng Liu, Zhengqi Li, and Hao Tan. RELIC: Interactive video world model with long-horizon memory. *arXiv preprint arXiv:2512.04040*, 2025. URL https://arxiv.org/abs/2512.04040.

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

Jiaqi Li, Junshu Tang, Zhiyong Xu, Longhuang Wu, Yuan Zhou, Shuai Shao, Tianbao Yu, Zhiguo Cao, and Qinglin Lu. Hunyuan-gamecraft: High-dynamic interactive game video generation with hybrid history condition. *arXiv preprint arXiv:2506.17201*, 2(3):6, 2025.

Kunyang Li, Mubarak Shah, and Yuzhang Shang. Packcache: A training-free acceleration method for unified autoregressive video generation via compact kv-cache. *arXiv preprint arXiv:2601.04359*, 2026.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. In *International Conference on Learning Representations*, volume 2026, pp. 91177–91196, 2026.

Jiayi Luo, Qiyan Liu, Tengyang Wang, JunHao Liu, Jiayu Chen, Cong Wang, Hanxin Zhu, Chen Gao, Xiaobin Hu, Qingyun Sun, et al. Future forcing: Future-aware training-free kv cache policy for autoregressive video generation. *arXiv preprint arXiv:2605.30083*, 2026.

Wenchao Ma, Changran Liu, Sharon X. Huang, and Haomiao Jiang. Closing the loop: Training-free revisit consistency for autoregressive generative rendering. *arXiv preprint arXiv:2607.21848*, 2026a. URL https://arxiv.org/abs/2607.21848.

Yuexiao Ma, Xuzhe Zheng, Jing Xu, Xiwei Xu, Feng Ling, Xiawu Zheng, Huafeng Kuang, Huixia Li, Xing Wang, Xuefeng Xiao, et al. Flow caching for autoregressive video generation. *arXiv preprint arXiv:2602.10825*, 2026b.

Xiaofeng Mao, Zhen Li, Chuanhao Li, Xiaojie Xu, Kaining Ying, and Kaipeng Zhang. Yume1. 5: A text-controlled interactive world generation model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7752–7761, 2026.

Yu Meng, Xiangyang Luo, Letian Li, Wenyuan Jiang, Chen Gao, Xinlei Chen, Yong Li, and Xiao-Ping Zhang. Tethercache: Stabilizing autoregressive long-form video generation with gated recall and trusted alignment. *arXiv preprint arXiv:2606.13035*, 2026.

Yuta Oshima, Yusuke Iwasawa, Masahiro Suzuki, Yutaka Matsuo, and Hiroki Furuta. Worldpack: Dynamic frame compression for long-context video world modeling, 2026. URL https://arxiv.org/abs/2512.02473.

Dvir Samuel, Issar Tzachor, Matan Levy, Michael Green, Gal Chechik, and Rami Ben-Ari. Fast autoregressive video diffusion and world models with temporal cache compression and sparse attention. *arXiv preprint arXiv:2602.01801*, 2026.

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

Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, et al. Advancing open-source world models, 2026. URL https://arxiv.org/abs/2601.20540.

Weijie Wang, Haoyu Zhao, Yifan Yang, Feng Chen, Zeyu Zhang, Yefei He, Zicheng Duan, Donny Y. Chen, Yuqing Yang, and Bohan Zhuang. Latent spatial memory for video world models. *arXiv preprint arXiv:2606.09828*, 2026. URL https://arxiv.org/abs/2606.09828.

Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory. *arXiv preprint arXiv:2506.05284*, 2025. URL https://arxiv.org/abs/2506.05284.

Xindi Wu, Sven Elflein, James Lucas, Olga Russakovsky, Laura Leal-Taixé, Despoina Paschalidou, Jonathan Lorraine, and Aljoša Ošep. Addressable memory for video world models. *arXiv preprint arXiv:2608.07408*, 2026.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pp. 21875–21895, 2024.

Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. Worldmem: Long-term consistent world simulation with memory. *Advances in Neural Information Processing Systems*, 38:49632–49652, 2026.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, et al. Longlive: Real-time interactive long video generation. *arXiv preprint arXiv:2509.22622*, 2025.

Jung Yi, Wooseok Jang, Paul Hyunbin Cho, Jisu Nam, Heeji Yoon, and Seungryong Kim. Deep forcing: Training-free long video generation with deep sink and participative compression. *arXiv preprint arXiv:2512.05081*, 2025.

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

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

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in neural information processing systems*, 36:34661–34710, 2023.

Zengqun Zhao, Yanzuo Lu, Ziquan Liu, Jifei Song, Jiankang Deng, and Ioannis Patras. Relax forcing: Relaxed kv-memory for consistent long video generation. *arXiv preprint arXiv:2603.21366*, 2026.

Guangcong Zheng, Teng Li, Xianpan Zhou, and Xi Li. Realcam-vid: High-resolution video dataset with dynamic scenes and metric-scale camera movements. *arXiv preprint arXiv:2504.08212*, 2025.

APPENDIX

A ADDITIONAL PROTOCOL DETAILS

Revisit pairs use the provided OpenCV camera-to-world matrices $T_s = (R_s, p_s)$ and $T_t = (R_t, p_t)$, with pose distance

$$d(s,t) = \frac{\|p_s - p_t\|_2}{\sigma} + \frac{1}{\pi} \arccos\left(\mathrm{clip}\left(\frac{\mathrm{tr}(R_s^\top R_t) - 1}{2}, -1, 1\right)\right). \tag{10}$$

We use a fixed translation scale $\sigma = 1$ in the stored pose units, without per-scene normalization. Pairs satisfy $d(s,t) \leq 0.15$ and $t - s \geq 30$ decoded RGB frames.

B ADDITIONAL EXPERIMENTS

Clip-level statistical analysis. The main table and statistical tests use the same three-judge mean scores. For both backbones, we average the eight same-place scores per clip and apply one-sided paired Wilcoxon signed-rank tests, with Holm correction across baseline comparisons within each backbone. We estimate 95% BCa confidence intervals from 10,000 paired clip resamples (seed 42). Table 3 reports the mean paired improvement ΔSP, its confidence interval, the adjusted p-value, and clip-level win/tie/loss counts for WorldRetain relative to each baseline. Each backbone is evaluated on 100 clips.

Table 3: Clip-level same-place comparisons on 100 loops per backbone. Positive ΔSP favors WorldRetain.

Baseline	ΔSP	95% CI	Holm-adjusted p	Win/Tie/Loss
(a) LingBot-World-Fast				
Window	+0.518	[0.415, 0.620]	2.7×10^{-12}	79/15/6
Relax Forcing	+0.198	[0.111, 0.285]	5.3×10^{-5}	59/18/23
Future Forcing	+0.328	[0.230, 0.421]	2.0×10^{-8}	67/18/15
WorldKV	+0.152	[0.054, 0.241]	8.2×10^{-5}	58/14/28
FlowCache	+0.510	[0.403, 0.632]	4.6×10^{-13}	78/11/11
(b) Matrix-Game 2.0				
Window	+1.653	[1.555, 1.746]	9.0×10^{-18}	100/0/0
Relax Forcing	+0.233	[0.155, 0.325]	1.3×10^{-7}	48/41/11
Future Forcing	+0.904	[0.757, 1.055]	4.3×10^{-14}	77/22/1
WorldKV	+0.676	[0.535, 0.829]	2.4×10^{-12}	64/33/3
FlowCache	+0.771	[0.623, 0.931]	1.0×10^{-12}	68/29/3

Spatial-compression controls. The three variants in Table 2(b) use identical archive membership and gist rules; only the L1 spatial representation changes.

C ADDITIONAL QUALITATIVE RESULTS

Figure 5 shows the camera-loop comparison on Matrix-Game 2.0. Each row follows the same trajectory from an initial view through departure to a return at the same pose, as in Figure 1. In this example, WorldRetain better preserves the initial scene layout at the return.



Figure 5: Camera-loop generation on Matrix-Game 2.0. Each row follows the same camera trajectory from an initial view through departure to a return at the same pose. In this example, WorldRetain better preserves the initial scene layout at the return.

D SAME-PLACE JUDGE PROMPT

Reported same-place scores jointly use Qwen3-VL-32B-Instruct (Bai et al., 2025), GPT-5.6-Sol, and Gemini 3.6 Flash. Each query is a single user turn that presents three images in order (condition frame, first-visit frame, revisit frame) followed by the text prompt below. Each model is asked for a JSON object with an integer `same_place` in $\{0, 1, 2, 3\}$. An `identity_drift` label and a short rationale are collected for diagnostics and are not used as headline metrics. Invalid JSON is retried once. For each sample, the reported same-place score is the arithmetic mean of the three parsed integer scores.

The prompt used for every reported same-place score is:

Same-place evaluation prompt

```
# Revisit Memory Judge (absolute)

You are judging world-memory consistency for a camera-controlled video
world model.

You are given THREE images in order:
1) CONDITION -- the conditioning first frame of the scene
2) FIRST-VISIT -- a frame when the camera first saw this viewpoint
3) REVISIT -- a later frame when the camera returned near the same
viewpoint
```

Same-place evaluation prompt (continued)

```
Task: decide whether FIRST-VISIT and REVISIT depict the **same
physical place** (layout / landmarks), ignoring mild lighting or
compression differences.

Score `same_place` as an integer:
- 0: clearly different place / severe geometry collapse
- 1: weak resemblance; major layout or landmark errors
- 2: mostly same place with noticeable appearance drift
- 3: clearly the same place; only minor differences

Choose exactly one `identity_drift` label for the dominant failure
(use "ok" if same_place is 3):
layout | appearance | object_missing | flicker | ok

Write a 1-2 sentence `rationale` in English.

Reply with ONLY a single JSON object (no markdown fences):
{"same_place": <0-3>, "identity_drift": "<label>", "rationale":
"<text>"}
```

E CACHE AND LEARNER IMPLEMENTATION DETAILS

This appendix specifies the cache layout and learner implementation. The predictor architecture, training configuration, and sample counts reported below apply equally to LingBot-World-Fast and Matrix-Game 2.0. Each generator supplies its own teacher rollouts.

Generation and measurement. All runs use eight NVIDIA H20 GPUs. LingBot uses two GPUs per clip with FSDP and sequence parallelism, except WorldKV, which uses four. Its DiT and T5 use bfloat16 and its VAE uses float32, with a 480×832 resolution preset adapted to the conditioning image's aspect ratio. Matrix-Game 2.0 uses one GPU per clip, a bfloat16 DiT, a float16 VAE decoder, and 352×640 images. LingBot and Matrix-Game use four and three denoising steps per chunk, respectively, followed by a clean-context forward pass to write KV. FPS is decoded RGB frames divided by synchronized wall-clock time, including all chunks; reported values average equally across clips.

Cache configuration. LingBot uses one-chunk initial and recent windows; Matrix-Game 2.0 uses a two-frame initial window, a one-chunk recent window, and its native six-frame budget. The preliminary archive capacity is

$$k_t = \min\big(|\mathcal{C}_t|, \max(k_{\min}, \lceil r|\mathcal{C}_t| \rceil)\big), \tag{11}$$

with recency breaking score ties. LingBot uses $r = 0.5$ and $k_{\min} = 2$. The final token ceiling is $B = 47{,}000$; Table 1 reports the realized mean active KV count, not this ceiling. Archive selection and compression precede the final budget check in Algorithm 1.

Causal descriptor. Table 4 lists the descriptor in MLP input order. For candidate chunk i at step t, $d_i = \|p_t - p_i\|_2 / s$ is the camera translation in units of the trajectory step s (default $s = 1$). t_i^{last} is the latest step at which chunk i geometrically aligned with the camera, and $\Delta t_i = \max(1, t - t_i^{\text{last}})$. The geodesic angle is $d_{\text{SO}(3)}(R_i, R_t) = \arccos((\text{tr}(R_i^\top R_t) - 1)/2)$, and $c_i = \langle f_i, f_t \rangle$ is the forward-direction cosine, or 0 when a pose is missing. m_i is the peak-mean control motion, from the Plücker or action difference, recorded when chunk i was generated. n_i^{rev} counts chunks j with $i < j \le t$ whose $\text{SE}(3)$ displacement from chunk i is at most 0.15. The remaining constants are positive scales r_{mot} and r_V, together with $v_{\text{ref}} = v_{\max} = 1$, $\varepsilon = 10^{-6}$, and $\tau = 16$ chunks.

Table 4: Coordinates of $g_t(i)$, in MLP input order.

#	Coordinate	Role
1	$\tanh(m_i/r_{\text{mot}})$	Control motion at generation; this coordinate is set to 1 for the first chunk.
2	d_i	Scaled camera translation.
3	$d_{\text{SO}(3)}(R_i, R_t)/\pi$	Geodesic rotation, normalized to $[0, 1]$.
4	c_i	Camera-forward cosine.
5	$\text{ov}(T_i, T_t)$	Depth-free frustum overlap.
6	$\tanh\big(d_i/(v_{\text{ref}}\Delta t_i + \varepsilon)\big)$	Translation divided by chunks since the last alignment.
7	$\exp\big(-d_i/(v_{\max}\Delta t_i + \varepsilon)\big)\,(1 + c_i)/2$	Reachability at that elapsed time, weighted by orientation.
8	$\cos(\bar{K}_i, \bar{K}_t)$	Cosine of the layer-0 key centroids.
9	$\tanh(\text{mean}\,\lvert V_i\rvert/r_V)$	Layer-0 mean absolute value.
10	$\log(1 + n_i^{\text{rev}})/\log 16$	Soft revisit count.
11	$\tanh\big((t - t_i^{\text{last}})/\tau\big)$	Chunks since the last geometric alignment.

Coordinate 5 casts nine image locations through the intrinsics at depths $\{2s, 5s, 10s\}$ and counts a hit when the ray projects inside the other image. It averages the $i \to t$ and $t \to i$ hit rates and does not test occlusion. Coordinates 8 and 9 use layer 0 only. Every coordinate is computed from poses and resident KV observed at or before step t. Future poses and future attention enter only the Future Coverage labels.

Predictor. The 11-dimensional inputs use training-set z-score statistics, fixed at inference. The MLP has two 32-unit GELU hidden layers and a scalar output (1,473 parameters), with biases and no dropout or LayerNorm.

Future Coverage labels. Attention is recorded in the clean-context forward pass after each chunk's denoising, using DiT layers whose indices are divisible by four. Let $\bar{a}_{t+h}$ be the attention distribution over cached keys, averaged over these layers, heads, and queries. Let L_{t+h} count all keys in this distribution, including Sink and Recent keys, and let $\Omega_{t+h}(i)$ contain the key-token indices of chunk i, with $m_{t+h,i} = |\Omega_{t+h}(i)|$. For a chunk present in the teacher cache, its raw attention mass and uniform-attention baseline are

$$A_{t+h}(i) = \sum_{p \in \Omega_{t+h}(i)} \bar{a}_{t+h}(p), \qquad b_{t+h}(i) = \frac{m_{t+h,i}}{L_{t+h}}.$$

The attention term in Eq. (5) is the bounded relative score

$$\widetilde{A}_{t+h}(i) = \frac{A_{t+h}(i)}{A_{t+h}(i) + b_{t+h}(i)}. \tag{12}$$

If chunk i is absent from the teacher cache, its attention mass is zero and we set $\widetilde{A}_{t+h}(i) = 0$. Under uniform attention, this score equals $1/2$ regardless of context length; higher values indicate attention above the uniform baseline. The geometric term is

$$\text{PoseReuse}_{t+h}(i) = \tfrac{1}{2}\left(\text{ov}(T_i, T_{t+h}) + \frac{1 + \langle f_i, f_{t+h}\rangle}{2}\right). \tag{13}$$

Both terms lie in $[0, 1]$; this gives a common range but does not imply that they have matched distributions or equal predictive value. The label-construction settings are $H = 8$, $\gamma = 0.9$, and $\alpha = 0.5$. With three latent frames per chunk, the supervision window spans eight future chunks, or 24 latent frames. These settings define offline supervision; the online predictor receives no future poses or attention.

Training. Each backbone supplies full-history teacher rollouts from 48 training trajectories of 481 RGB frames, disjoint from evaluation clips. After excluding constant-label groups, each backbone has 1,632 ranking groups, split at the group level into 1,306 training and 326 validation groups. Both predictors use Adam for 300 epochs, learning rate 10^{-3}, zero weight decay, no scheduler, and seed 0. Each update minimizes Eq. (6) over one ranking group.

E.1 SPATIAL COMPRESSION DETAILS

Spatial compression has two archive representations: *L1* retains selected original tokens plus packing summaries; *L2* retains only a gist. Chunks outside the retained archive are discarded.

Saliency. Eq. (8) uses channel-averaged adjacent-chunk residuals and spatial 2×2 pooling to align with the token grid. Saliency is shared across layers; missing saliency uses uniform selection. Selector features are stored from full-resolution KV at insertion.

Number and location of retained tokens. Let $s_{i,(1)} \geq \cdots \geq s_{i,(n_i)} \geq 0$ be sorted saliency values. For positive total saliency, define

$$k_i^{\text{cap}} = \max\{1, \min\{n_i, \lceil \rho n_i \rceil\}\}, \tag{14}$$

$$k_i^{\text{energy}} = \min\left\{ k \in \{1, \ldots, n_i\} : \sum_{j=1}^{k} s_{i,(j)} \geq \eta \sum_{j=1}^{n_i} s_{i,(j)} \right\}, \tag{15}$$

$$k_i = \min\{k_i^{\text{cap}}, k_i^{\text{energy}}\}. \tag{16}$$

The energy criterion applies only to positive saliency mass; missing or zero saliency uses uniform selection. The defaults are $\rho = 0.5$ and $\eta = 0.9$. The energy target need not be achieved when the cap binds. Token locations follow standard top-k_i saliency selection unless the saliency Gini coefficient is below 0.20, in which case a uniform lattice supplies the locations. Missing saliency also uses the lattice.

Packing summaries. Let $\mathcal{I}_i$ be the selected original token indices and $\mathcal{D}_i = \{1, \ldots, n_i\} \setminus \mathcal{I}_i$. Spatial cell pooling groups omitted positions into nonempty cells $\mathcal{D}_{i,1}, \ldots, \mathcal{D}_{i,q_i}$, with $q_i \leq m$. Here $m = 64$ is the fixed per-chunk L1 summary cap in Eq. (9) for all experiments on both backbones; fewer summaries are stored when fewer nonempty cells are available. For each layer and attention head, the raw pooled vectors are

$$\bar{K}_{i,b} = \frac{1}{|\mathcal{D}_{i,b}|} \sum_{j \in \mathcal{D}_{i,b}} K_{i,j}, \qquad \bar{V}_{i,b} = \frac{1}{|\mathcal{D}_{i,b}|} \sum_{j \in \mathcal{D}_{i,b}} V_{i,j}. \tag{17}$$

Summary keys additionally undergo mean-absolute-magnitude alignment; values use the pooled means. This summary-KV procedure is identical on LingBot-World-Fast and Matrix-Game 2.0. The summaries are concatenated after the selected original KV tokens, so L1 stores $k_i + q_i$ tokens. Neither protected initial nor recent chunks enter this operator.

Positional encoding. Keys receive absolute temporal and spatial RoPE at insertion. Compression pools these already rotated keys; retained original keys keep their phases. Summaries are used directly in attention, without assigning new coordinates or applying RoPE again.

Gist allocation. The default configuration uses both L1 packing and L2 gists. The lower-ranked half of the retained archive is assigned to L2, with $m_{\text{gist}} = 64$ tokens per chunk and a separate total budget $B_{\text{gist}} = 512$; overflow removes the lowest-scoring gist chunks. The L1 summary cap $m = 64$ and L2 gist length are separate parameters, fixed on both backbones. L1 summaries supplement original tokens, whereas L2 gists replace the chunk representation. Conversion from L1 to L2 pools the currently stored KV, including existing summaries, without restoring removed tokens or their coordinates. The chunk-retention ratio r controls archive membership, while ρ caps original tokens within L1.