Addressable Memory for Video World Models

Xindi Wu[1 2] **Sven Elflein**[1 3 4] **James Lucas**[1] **Olga Russakovsky**[2]
Laura Leal-Taixé[1] **Despoina Paschalidou**[1] **Jonathan Lorraine**[1] **Aljosa Osep**[1]

[1]NVIDIA [2]Princeton University [3]University of Toronto [4]Vector Institute

Project page: research.nvidia.com/labs/sil/projects/WorldTrace/

Abstract

We study visual persistence in interactive video world models. These models rely on a Key-Value (KV) cache as a growing visual memory to carry forward previously generated frames. However, we find that models can no longer reliably address stored content once rollouts extend beyond the training horizon, because temporal Rotary Positional Embedding (RoPE) offsets can exceed the training range and the model struggles to retrieve the relevant visual information through attention. Moreover, naively compressing the cache in the RoPE-rotated space degrades memory by averaging keys with incompatible positional phases. To address this, we propose WORLDTRACE, a training-free memory framework for long-horizon visual persistence. WORLDTRACE keeps compressed memory addressable by assigning each summary slot a distinct, in-distribution virtual position. Within this cache, WORLDTRACE uses the model's own attention to decide which observations to retain verbatim and which redundant ones to merge into compressed summaries, all within one bounded addressable memory bank. We evaluate this ability with LOOPBENCH, a benchmark that tests whether a compressed cache can reconstruct a previously visited scene after a long detour. On LOOPBENCH, WORLDTRACE improves average DINOv2 recall similarity by 32.6% over a sliding window at the same cache budget, extending visually persistent generation without retraining.

1 Introduction

Memory is what makes a generated world persistent. A world model should not only predict what comes next, but also preserve what has already been seen. World models aim to simulate worlds we can explore [1, 2, 3], and recent autoregressive video world models [4, 5, 6] pursue this vision by generating video scenes chunk-by-chunk, with each chunk attending over a Key-Value (KV) cache of prior context. These models promise interactive applications such as next-generation game engines [7, 6, 5] and closed-loop robot simulators [8, 9], in which users can move freely through a visual world and revisit prior locations. Such revisits demand *visual persistence*, meaning that when an agent returns, the generated results should be consistent with the scene's original appearance, not a plausible-looking alternative.

In practice, visual persistence degrades rapidly once generation exceeds the training context length. A natural way to address this issue is to compress the linearly growing KV cache into a fixed-size memory [10, 11, 12, 13] equal to the size of the context seen during training. However, we find that the bottleneck is not simply whether past content is stored in the cache, but whether it remains addressable and how it is compressed. As the generation horizon extends beyond the training context length, the positional encoding these models rely on (e.g., RoPE [14]) is queried at position indices outside its training-bounded range, so the model needs to attend over relative distances that are out-of-distribution (OOD), degrading attention-based memory retrieval. In other words, even if past memories are stored in the KV cache, the model cannot reliably retrieve them. A summary whose offset exceeds the trained range can therefore become harder to reach. This reveals that long-horizon failure is fundamentally a problem of *addressability*, a prerequisite for long-horizon visual persistence.

WORLDTRACE. To tackle this problem, we propose WORLDTRACE, a training-free approach for addressable compressed memory over long horizons. We assign each summary slot a fixed

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

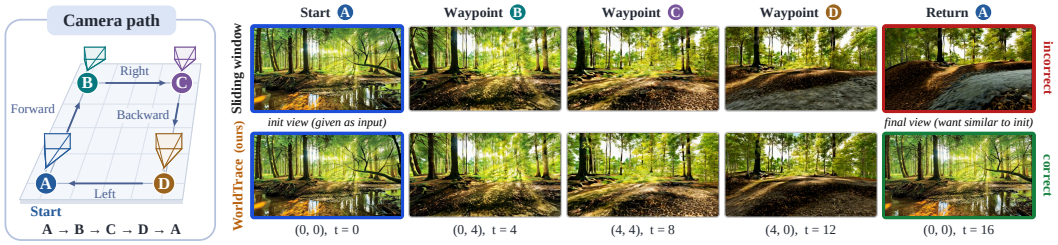


Figure 1: **Addressable memory maintains long-horizon visual persistence.** *(Left)* Loop topology A→B→C→D→A. *(Right)* Representative frames on Matrix-Game-2 along the path, covering the initial scene A (blue border), waypoints B, C, and D, and the return to A. *Sliding window* (top) mismatches the reference appearance at the return (red border). Our WORLDTRACE with frozen landmark keys (green border) matches scene A at the return.

temporal offset from the current frame within the range seen during training, so that content from distant frames remains accessible to attention. The cache is organized into a recent window and summary slots for the distant horizon. Within this cache, we formulate memory compression as a structured projection of the growing visual history into a fixed set of retrievable traces. The projection can average redundant visual content into compact summaries while preserving observations with distinctive details verbatim. Both types of memory fit within the same cache and share a position assignment that keeps them addressable as generation continues.

Our experiments show that addressable memory extends the effective memory horizon. To evaluate episodic recall under controlled scene revisits, we propose LOOPBENCH, a *memory* benchmark that tests whether the compressed KV cache can retrieve a previously visited scene after a long detour. While the sliding-window baseline begins to forget scene structure after only a few seconds and loses structural fidelity when revisiting the same location, our WORLDTRACE maintains consistent scene reconstruction over long rollouts and recovers the original scene appearance (Fig. 1).

Our contributions are:

1. We identify two coupled challenges for long-horizon generation in autoregressive video world models, *where* memory is placed and *what* it stores. Temporal offsets beyond the range seen during training can impair retrieval from the cache, and RoPE phase cancellation makes naively compressed summaries uninformative (Sec. 2).
2. We propose WORLDTRACE, a training-free framework that keeps compressed memory both addressable and informative. It assigns each summary a virtual position whose offset stays within the training range (Sec. 3.1) and organizes the growing visual history into a fixed set of retrievable traces. The same cache supports temporal-group averages through WORLDTRACE-FIELD (Sec. 3.3) and retained scene-entry frames through WORLDTRACE-LANDMARK (Sec. 3.4).
3. We introduce LOOPBENCH, a *memory* benchmark of controlled scene revisits that measures whether a compressed cache can reconstruct a previously visited scene (Sec. 4). On Matrix-Game 2, WORLDTRACE improves average DINOv2 recall similarity across the four LOOPBENCH task families by 32.6% without retraining over the default sliding window baseline.

2 Rethinking Memory in Video World Models

Autoregressive video world models retain prior context as cached keys and values during generation. Long-horizon memory therefore depends both on whether relevant past context remains inside the cache and on whether the model can effectively attend to it during future generation. Long-horizon memory requires both retrieving retained content and preserving it during compression. Retrieval can become less reliable when a cached key's temporal offset exceeds the training range (Sec. 2.1). Compression introduces a separate problem. Averaging keys with different RoPE rotations can partially cancel their components and change the query-key scores used to retrieve the corresponding values (Sec. 2.2). We summarize the notation in App. A.

2.1 Position Determines Whether Memory Is Addressable

Attention retrieves information from cached values using weights computed from query–key scores. With the cached key's temporal position held fixed, its relative temporal offset from the query grows

as later frames are generated. The offset can exceed the trained range, which can reduce the reliability of attention-based retrieval. Concretely, during training the query only attends over a bounded range of relative offsets $\delta_{q,k}$ between the current frame and each cached frame:

$$0 \leq \delta_{q,k} = q - k \leq \Delta t_{\text{train}},$$

where q is the absolute position of the current query frame, k that of a cached key frame, with $k \leq q$ under causal autoregressive attention, and Δt_{train} the largest offset seen during training. For temporal RoPE pair f at angular frequency θ_f, its contribution to the query-key attention score is:

$$a^f_{q,k} = \text{Re}\big(A^f_{q,k} e^{i\theta_f \delta_{q,k}}\big),$$

where $\text{Re}(\cdot)$ is the real part and $A^f_{q,k}$ is the content-dependent inner product in canonical coordinates between the query at q and the key at k. During long-horizon inference, these offsets grow ever larger, and $\delta_{q,k}$ can exceed Δt_{train}. The affected RoPE components then take on rotation angles the model never saw during training. Each temporal RoPE pair rotates at a rate determined by its frequency. High-frequency pairs may complete multiple rotations within the training window, whereas low-frequency pairs may cover only a limited range of phases. At longer temporal offsets, these low-frequency pairs can encounter phases outside that range. The effect on retrieval depends on both the relative phase and the content-dependent query-key interactions. We report the rotation angle at each temporal frequency in App. C.

Several concurrent methods investigate these positional issues. Infinity-RoPE [15] proposes Block-relative RoPE, which reindexes each cached frame relative to the current AR chunk so that $\delta_{q,k}$ always stays within the trained range, keeping generation in-distribution. It targets infinite-length video generation rather than memory compression. Applied to a compressed cache, however, this reindexing clips every offset beyond the trained range to Δt_{train}, so summary slots that stand for older history share one temporal position and lose their distinct positional cues. MemRoPE [13] compresses history into dual EMA memory streams with unrotated key storage and contiguous online indexing, a design that avoids this collapse for its two summary positions but does not extend to an N-slot cache, where capped Block-relative positions would again collide. Our WORLDTRACE targets an N-slot compressed cache directly by assigning every slot a distinct, in-distribution slot-rank position, so an arbitrary number of summaries stay individually addressable at any horizon. We discuss related work in App. D.

2.2 Content Compression Determines Whether Memory Is Informative

In an interactive world model, compressed memory needs to support reconstructing a previously visited place after it has left the current view. During causal generation, however, attention weights are computed for the current query, so a past frame that receives little attention now may still be needed once the camera returns to the place it shows. Compression therefore needs to reduce redundancy among repeated observations while preserving distinctions that may become useful again under a different viewing condition. A summary at an in-distribution temporal position still needs to preserve the distinctions between the source views. A common way to compress cached keys is averaging [16, 17]. With RoPE-rotated keys, the averaged key at frequency f is:

$$\bar{K}^f_{\text{naive}} = \frac{1}{M} \sum\nolimits_{m=1}^{M} R(\theta_f t_m)\ K^f_m,$$

where M is the number of source frames and t_m is the timestamp of source frame m. K^f_m denotes that frame's canonical key at frequency f, and $R(\theta_f t_m)$ is its temporal RoPE rotation. Each frame is rotated by a different angle $\theta_f t_m$ before averaging, so $\bar{K}^f_{\text{naive}}$ sums vectors that point in different directions. For a given frequency, nearby timestamps produce similar rotations when their phase difference is small. Averaging then introduces little additional cancellation from the positional rotations. If they come from distant timestamps, their angles can point in opposite directions and partially cancel, weakening that frequency's contribution to the summary regardless of what it encoded. Naive averaging in RoPE-rotated space can therefore distort the compressed memory.

The two failures are coupled. Most prior methods fix only one of the two, with one line of work adjusting *where* memories are placed, reparameterizing or capping offsets (e.g., Block-relative [15]), while another decides *what* each summary stores, whether by smoothing past keys with dual-rate EMA [13], merging them by temporal averaging [16, 17], or evicting entries [10, 11, 12]. Addressable positions cannot help if compression cancels the signal a summary should carry, and an informative summary cannot help if it sits at an out-of-distribution position. Maintaining visual persistence over long rollouts therefore requires compressed memory that stays both addressable and informative beyond the training window.

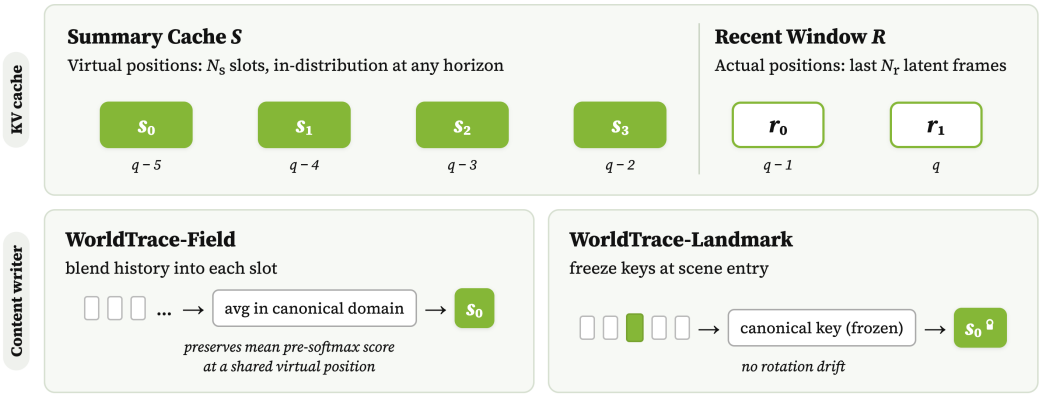


Figure 2: **WORLDTRACE overview.** WORLDTRACE partitions the context window into a compressed *summary* cache $\mathcal{S}$ for the distant past and a verbatim *recent* window $\mathcal{R}$ (top). Each slot occupies one latent frame in the cache, with recent slots holding a verbatim frame and summary slots holding compressed history at a virtual position. Both WORLDTRACE-FIELD (bottom left), which averages unrotated keys, and WORLDTRACE-LANDMARK (bottom right), which freezes the canonical keys of scene-entry landmarks, store keys in canonical space and re-rotate them at each slot's virtual position.

3 WORLDTRACE

We propose WORLDTRACE, a training-free approach for addressable memory in video world models. WORLDTRACE maintains a bounded trace of distant history in fixed summary slots via canonical-space key assignment, while keeping each slot *addressable* (Sec. 3.1). We formulate this cached memory compression for visual persistence as a structured attention approximation (Sec. 3.2). Within this slot structure, WORLDTRACE-FIELD (Sec. 3.3) averages groups of frames to cover longer history and reduce redundancy, while WORLDTRACE-LANDMARK (Sec. 3.4) retains selected frames for episodic recall, and WORLDTRACE combines both in a single memory bank (Sec. 3.5).

3.1 Virtual Position Assignment with WORLDTRACE Slot Indexing

Cache structure. As shown in Fig. 2, we partition the local attention window L_{attn} into a **recent window** $\mathcal{R}$ of N_r verbatim latent frames and a **summary cache** $\mathcal{S}$ of N_s slots storing compressed history, so that $N_s + N_r = L_{\text{attn}}$. We assign each summary slot a distinct temporal offset from the current query. Keeping these offsets fixed as generation advances keeps them within the trained range at every horizon. Slot rank determines the offset through the linear assignment below. Consider a full cache, after at least L_{attn} frames have been generated, and choose $L_{\text{attn}}-1 \le \Delta t_{\text{train}}$. The recent window occupies the latest N_r positions. A summary position needs to precede that window while staying no more than Δt_{train} steps behind the current query, so it satisfies:

$$t^v_{\min} = \max\big(0,\ q - \Delta t_{\text{train}}\big), \qquad t^v_{\max} = q - N_r. \tag{1}$$

We place the summary slots consecutively within this interval, ordered from oldest to newest.

Definition 1 (WORLDTRACE Slot Indexing)**.** Given current query position q, local attention window L_{attn} $(= N_s + N_r)$, and summary cache size N_s, the virtual position of summary slot s is:

$$t^v_s \ = \ q - (L_{\text{attn}}-1-s), \quad s = 0, \ldots, N_s-1. \tag{2}$$

The oldest summary slot has offset $L_{\text{attn}}-1$, and the newest has offset N_r. Both endpoints lie within the trained range under the cache-size condition above. The relative offsets depend only on slot rank, not the absolute horizon.

Why canonical keys. A slot's virtual position t^v_s is set by slot rank and generally differs from its source frames' timestamps. We therefore remove each source key's temporal RoPE rotation before storing or averaging it, and apply the rotation at t^v_s when the key is read. Without this step, (i) keys with distinct rotations $R(\theta_f t_m)$ would partially cancel when averaged, and (ii) the compressed key would carry the phase of its source timestamps rather than that of t^v_s. MemRoPE [13] also stores keys canonically but uses contiguous online indexing, whereas we pair canonical storage with slot-rank positions (Def. 1). Both WORLDTRACE writers share this canonical key domain and differ only in which keys they write. The content writer determines which canonical keys contribute to each slot.

3.2 Memory Compression as Structured Sparse Attention

With slot positions fixed, the remaining choice is *what* to store in each slot. Storing every past frame makes the cache grow with generation length. Evicting all distant frames bounds its size but removes the cached evidence for later revisits. We use a cache of L_{attn} slots to approximate full attention over all T past frames. We consider two structured projections of history: sparse selection of landmark frames and averaging within temporal groups. The projection matrix $P \in \mathbb{R}^{L_{\text{attn}} \times T}$ defines this pattern by giving slot i the compressed key $P_i K^f_m$, rotated to the slot's virtual position. The recent window uses verbatim identity rows for the last N_r frames, and the N_s summary rows carry the compressed information.

Approximating full attention. Attending over all T past frames would produce a softmax-weighted combination of their values, with weights α_q determined by query-key similarity. In order to keep the cache bounded, memory compression replaces this with projected attention over L_{attn} compressed keys, where row i of P defines which past frames contribute to compressed key i, given by $P_i K^f_m$. Re-encoding each compressed key at t^v_s keeps the approximation in-distribution at any horizon. We approximate the error with the *distribution mismatch* between the full attention weights α_q and the weights induced by the compressed cache (Prop. 1, App. B). The choice of P therefore determines what history is preserved under a fixed slot budget.

Choosing the projection matrix. This view reduces memory compression design to the choice of P, with a good P minimizing the distribution mismatch above for the queries encountered during rollout. Relaxing the entries of P from fixed patterns to arbitrary nonnegative values turns this minimization into a nonnegative matrix factorization (NMF) problem [18], approximately factoring the matrix of full attention weights into the product of two nonnegative low-rank factors (App. B). The optimal row structure of P then depends on these queries. Rows that average contiguous temporal groups suit future attention that spreads smoothly over history, while rows that select single frames verbatim suit attention concentrated on a few salient past frames. These two attention patterns motivate the two writers we introduce next, WORLDTRACE-FIELD (Sec. 3.3) and WORLDTRACE-LANDMARK (Sec. 3.4), which together constitute the unified WORLDTRACE memory (Sec. 3.5).

3.3 Canonical Key Averaging with WORLDTRACE-FIELD

WORLDTRACE-FIELD uses each summary row of P to average a contiguous temporal group of frames. This represents older history at a coarser temporal resolution while preserving a verbatim recent window. We average in canonical coordinates to avoid cancellation from the source frames' different RoPE rotations. For each temporal pair f, the compressed key at virtual position t^v is:

$$K^f_{\text{field}}(t^v) \;=\; R(\theta_f t^v)\; \frac{1}{M} \sum_{m=1}^{M} R(-\theta_f t_m)\; K^f_{t_m}, \tag{3}$$

we unrotate each key to its canonical content $R(-\theta_f t_m)\; K^f_{t_m}$, average in that space, and then re-encode at virtual position t^v. RoPE rotates queries and keys but leaves values unchanged. Slot values are the average of their source-frame values. Here T is the number of latent frames seen so far. We split the $T-N_r$ frames outside the recent window into N_s contiguous groups, ordered from oldest to newest. Each slot therefore represents approximately $M \approx (T-N_r)/N_s$ source frames. This count is determined by the grouping and grows with rollout length.

Each new chunk adds F frames, displacing older frames from the recent window and changing which frames each group covers, so the slot averages have to be maintained as the rollout grows. The writer keeps the canonical key of each evicted frame in host memory. When group boundaries change, it recomputes each slot mean from the original source keys assigned to that group.

Remark 1 (Mean score preservation)**.** *A summary slot represents M source frames. Canonical averaging preserves their mean query-key score when all source keys are assigned the same virtual position t^v. It removes the source keys' positional phases before averaging, avoiding the phase cancellation that can distort scores when rotated keys are averaged (Sec. 2.2). By linearity of the dot product, the score of the average key equals the average of the source-key scores at that shared position. Attention weights are different. Softmax exponentiates each score and normalizes across keys, so averaging several keys generally changes their total weight.*

3.4 Frozen Landmark Keys with WORLDTRACE-LANDMARK

To optimize for episodic recall, WORLDTRACE-LANDMARK fills the N_s slots with verbatim high-value past frames rather than averaged summaries. Landmark Attention inserts landmark tokens in

trained transformers [19]. Here, we select existing scene-entry frames and store their canonical keys at slot-rank virtual positions without retraining.

Landmark selection. For each incoming frame, we compute the cosine distance between its canonical keys and those of the preceding frame. A distance above threshold τ marks a scene-entry event. WORLDTRACE-LANDMARK fills the N_s summary slots with the most recent scene-entry frames verbatim, and while fewer than N_s scene-entry events have been detected, the remaining slots repeat the oldest available landmark. Once more than N_s scene-entry events have been detected over a long rollout, the oldest is evicted to make room for the newest. Each retained landmark frame is assigned its summary slot's virtual position using Eq. (2). Its original timestamp is used only to remove the source RoPE rotation.

Frozen landmark keys. Slots are ordered from oldest (s=0) to newest. Once the landmark bank is full, inserting a new scene-entry frame evicts the oldest landmark and shifts the remaining landmarks one slot toward the older end. Between insertions, slot ranks stay fixed, while virtual positions change as the query timestamp advances. Repeatedly unrotating a stored key and re-rotating it at an updated virtual position can accumulate floating-point error, particularly in bfloat16 precision [20]. WORLDTRACE-LANDMARK removes this drift by freezing each selected key in canonical form, storing that key once at landmark time and applying a single fresh rotation to the current virtual position whenever the key is attended to:

$$K^f_{\text{land}}(t^v_s) \;=\; R(\theta_f t^v_s)\; R(-\theta_f t_{\ell^*})\; K^f_{t_{\ell^*}}, \tag{4}$$

where t_{ℓ^*} is the original timestamp of the selected landmark frame. This is the single-frame case of Eq. (3), with M=1, so the stored key is retained without averaging and rotated to its current virtual position when read. The canonical key $R(-\theta_f t_{\ell^*})\; K^f_{t_{\ell^*}}$ is computed once at landmark time and reused across subsequent reads. The canonical-caching mechanism is shared with concurrent work that stores RoPE keys canonically and rotates at attention time (e.g., MemRoPE [13]). WORLDTRACE-LANDMARK differs in (i) applying it selectively at detected scene-entry events rather than to every cached key, and (ii) combining it with the slot-rank position assignment, so each frozen landmark occupies a distinct in-distribution virtual position at any horizon.

3.5 Combining Retained Frames and Summaries

Memory compression under a fixed budget. A fixed memory budget limits how many observations we can retain. Merging similar observations makes room for a longer history, but can remove details needed to recall a specific scene. We therefore evaluate each candidate merge over a range of possible future queries, using the attention-response error it introduces, defined as the difference between each query's attention output before and after the merge, together with a penalty for losing unique information. WORLDTRACE-LANDMARK preserves selected observations as individual key-value pairs for episodic recall. WORLDTRACE-FIELD instead combines compatible observations into averaged summaries, weighting each summary by the number of observations it represents. This retains a longer history within the same budget. These summaries reduce storage and attention costs compared with keeping every observation.

The model can retrieve a previous view without an explicit camera trajectory because its attention query depends on the current hidden state. This state can reflect visual context, motion, and action conditioning. For example, as the camera turns back toward a garage, the query can match keys from an earlier view of that garage and retrieve the associated values. Successful recall depends on preserving the features that distinguish the garage from other scenes.

Attention-guided compression. WORLDTRACE combines individual observations and compressed summaries in a single bounded memory bank. The model reads both through the same attention mechanism. To decide which slots to merge or evict, we evaluate candidate updates using attention-response error on sampled queries. We select the update that minimizes this error, so that compression alters what the model retrieves as little as possible. Compatible observations are averaged, with each summary weighted by the number of observations it represents. For incompatible observations, we retain a paired exemplar. Under this rule, an observation retained without merging forms a landmark, while a group of averaged observations forms a field summary. WORLDTRACE-LANDMARK and WORLDTRACE-FIELD correspond to these two cases. Both use slot indexing to assign each memory slot a distinct in-distribution virtual position. Whether a slot holds an averaged key or a frozen landmark key, we store it in canonical form and re-encode it at the slot's virtual position. This allows the memory update to select content independently of its original position.

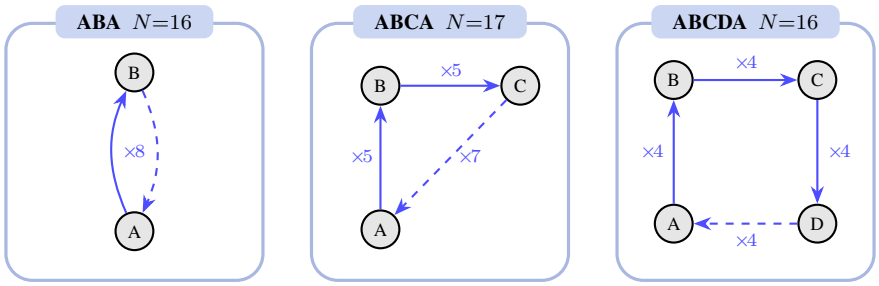


Figure 3: **Three LOOPBENCH geometries.** We propose LOOPBENCH, a memory benchmark with diverse geometries, including ABA (straight reversal), ABCA (approximate L-triangle with diagonal $5\sqrt{2} \approx 7$), and ABCDA (square). N is rollout length in AR chunks. Each path returns to scene A. We provide a full gallery of rollout-length, orientation, and multi-revisit settings in App. G (Fig. 7).

4 Experiments

We evaluate whether WORLDTRACE helps a video world model reconstruct a previously visited scene after a detour. Sec. 4.1 introduces the models, LOOPBENCH protocol, metrics, and baselines. Sec. 4.2 reports recall across four task families. We compare WORLDTRACE-FIELD, WORLDTRACE-LANDMARK, and the combined WORLDTRACE (which uses both) at the same attended-cache size in Tab. 2, with total memory use reported separately.

4.1 Setup

Models. Our main evaluation uses Matrix-Game-2 (MG2-1.3B) [5], a distilled 1.3B-parameter autoregressive game world model based on Wan 1.3B T2V [21] with 3D-RoPE. Each AR chunk contains F=3 latent frames. The training-time KV-cache extent is $L_{\text{train}} = 6$ chunks, while the local attention window spans 2 chunks, or 6 latent frames. The corresponding maximum temporal offset is $\Delta t_{\text{train}} = 5$ latent-frame steps. We provide additional experiments on the 14B LingBot-World model [4] in App. F.

Metrics. We compare the frames generated at the revisited location (e.g., A in ABA) on the first visit and on the return, using two recall metrics. Scene Consistency (↑) measures the cosine similarity of DINOv2 features [22], and Return LPIPS (↓) measures a learned perceptual distance [23]. These measure scene-feature similarity and perceptual appearance, respectively. Neither alone establishes exact reconstruction. We also report Attention Mismatch (Δ_{attn} ↓), a compression diagnostic that compares attention distributions under identical queries and source history. We provide the full definitions and preprocessing in App. E.

Baselines. Our baseline is MG2's default sliding-window cache. It evicts the oldest entries without compressing them and uses no additional position correction. Retaining only recent entries already bounds their relative offsets. We compare it with the WORLDTRACE-FIELD and WORLDTRACE-LANDMARK writers and with WORLDTRACE, which combines both. The evaluated WORLDTRACE implementation keeps a bank of verbatim frames alongside compressed summaries and selects an entry using the model's attention queries (Tab. 2). All methods in Tab. 1 share the same attended-cache size. Their total memory use differs. We compare with the training-free positional baseline MemRoPE [13].

Benchmark. We introduce LOOPBENCH, a *memory* benchmark for scene revisits in AR world models. The model follows a path through waypoints and returns to a previously visited location. We evaluate the return frame against the earlier generated view at a geometrically matched location, without an external reference image. This tests whether a model preserves a particular scene across a detour. A plausible new scene can still differ from the one previously generated. Varying the detour separates the effects of elapsed rollout length and intervening views. LOOPBENCH varies waypoint count (K, Fig. 3), rollout length (N), camera orientation, and multi-revisit patterns. Fig. 7 in App. G shows all evaluated configurations.

4.2 Main Results

We report DINOv2 similarity and LPIPS on LOOPBENCH. The qualitative examples in Fig. 4 show the sliding-window baseline failing to reconstruct scene A at revisits, whereas WORLDTRACE restores

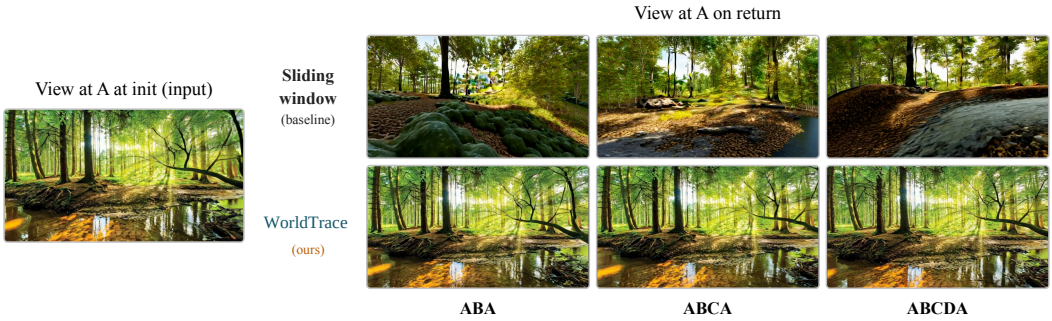


Figure 4: **LOOPBENCH qualitative results** on the three revisit geometries in Fig. 3 (ABA, ABCA, and ABCDA). The sliding-window baseline fails to reconstruct scene A when revisiting. WORLDTRACE matches scene A in all three, keeping memory addressable across waypoints.

Table 1: **Verbatim, addressable memory improves episodic recall in every LOOPBENCH tier.** DINOv2 (↑) / LPIPS (↓) between the frames generated at the revisited location (e.g., A in ABA) on the first visit and on the return, mean over n=51 rollouts, six-frame attended window for every method (N_s=4, N_r=2 for the WorldTrace variants). The WORLDTRACE row reports the combined variant that uses both WORLDTRACE-FIELD and WORLDTRACE-LANDMARK. Loop geometries are in Fig. 7 (App. G).

Tier 1, topology	ABA (K=1, N=16)	ABCA (K=2, N=17)	ABCDA (K=3, N=16)
Sliding window	0.577 / 0.643	0.627 / 0.643	0.599 / 0.640
MemRoPE	0.623 / 0.621	0.630 / 0.630	0.607 / 0.626
WORLDTRACE	**0.798 / 0.512**	**0.767 / 0.548**	**0.805 / 0.489**
Tier 2, rollout length	ABA short (N=8)	ABA (N=16)	ABA long (N=32)
Sliding window	0.805 / 0.525	0.577 / 0.643	0.422 / 0.700
MemRoPE	0.816 / 0.505	0.623 / 0.621	0.464 / 0.694
WORLDTRACE	**0.891 / 0.460**	**0.798 / 0.512**	**0.750 / 0.537**
Tier 3, camera orientation	Pan 90° (N=4)	Pan 180° (N=8)	Pan 360° (N=8)
Sliding window	0.838 / 0.561	0.560 / **0.634**	0.406 / 0.740
MemRoPE	0.844 / 0.547	0.561 / 0.639	0.394 / 0.742
WORLDTRACE	**0.911 / 0.453**	**0.784** / 0.649	**0.423 / 0.722**
Tier 4, multi-revisit	ABABA (N=32)	ABCBA (N=20)	ABCDBA (N=20)
Sliding window	0.462 / 0.692	0.571 / 0.659	0.595 / 0.653
MemRoPE	0.496 / 0.675	0.584 / 0.642	0.602 / 0.634
WORLDTRACE	**0.791 / 0.518**	**0.807 / 0.515**	**0.808 / 0.503**

its appearance across the three geometries. The WORLDTRACE row in Tab. 1 reports the combined variant, whose values also appear in Tab. 2. The four evaluation task families are *topology* (ABA, ABCA, ABCDA), *rollout length* (ABA at $N \in \{8, 16, 32\}$), *camera-orientation* (pan 90°/180°/360° with the agent staying at A), and *multi-revisit* (ABABA, ABCBA, ABCDBA). The WorldTrace variants use N_s=4, N_r=2.

WORLDTRACE has the highest mean DINOv2 score among the methods in Tab. 1 in every listed MG2 condition, improving over the sliding window by approximately 32.6% when DINOv2 is averaged across the 12 conditions. Its LPIPS is also lower than the sliding window in all but Pan 180° (0.649 versus 0.634). MemRoPE [13], evaluated under the same attended-cache size, stays within 0.046 DINOv2 of the sliding window across the listed conditions, with lower LPIPS in all but Pan 180° and Pan 360°. The gap to the sliding window is larger at long horizons. In the horizon sweep (Tab. 7), at N=128, WORLDTRACE reaches 0.698 DINOv2 / 0.572 LPIPS, compared with 0.251 / 0.751 for the sliding window. We report cross-architecture results in App. F.

4.3 Ablations

We ablate the two writers by comparing a field-only memory (WORLDTRACE-FIELD), which keeps only averaged summaries, a landmark-only memory (WORLDTRACE-LANDMARK), which keeps

Table 2: **WORLDTRACE performance on LOOPBENCH.** DINOv2 (↑) / LPIPS (↓) between the frames generated at the revisited location (e.g., A in ABA) on the first visit and on the return on LOOPBENCH, mean over n=51 rollouts, six-frame attended window. WORLDTRACE-FIELD keeps compressed summary slots only, WORLDTRACE-LANDMARK fills all N_s slots with verbatim frames, and WORLDTRACE combines both, keeping a bank of verbatim frames alongside compressed summaries and exposing the one the queries select.

Tier 1, topology	ABA (K=1, N=16)	ABCA (K=2, N=17)	ABCDA (K=3, N=16)
WORLDTRACE-FIELD	0.591 / 0.636	0.610 / 0.640	0.588 / 0.637
WORLDTRACE-LANDMARK	0.723 / 0.543	0.709 / 0.561	0.721 / 0.557
WORLDTRACE	**0.798 / 0.512**	**0.767 / 0.548**	**0.805 / 0.489**
Tier 2, rollout length	ABA short (N=8)	ABA (N=16)	ABA long (N=32)
WORLDTRACE-FIELD	0.785 / 0.530	0.591 / 0.636	0.409 / 0.703
WORLDTRACE-LANDMARK	0.865 / 0.466	0.723 / 0.543	0.632 / 0.588
WORLDTRACE	**0.891 / 0.460**	**0.798 / 0.512**	**0.750 / 0.537**
Tier 3, camera orientation	Pan 90° (N=4)	Pan 180° (N=8)	Pan 360° (N=8)
WORLDTRACE-FIELD	0.818 / 0.576	0.555 / 0.650	0.378 / 0.746
WORLDTRACE-LANDMARK	**0.917** / 0.495	0.731 / **0.608**	0.400 / 0.729
WORLDTRACE	0.911 / **0.453**	**0.784** / 0.649	**0.423 / 0.722**
Tier 4, multi-revisit	ABABA (N=32)	ABCBA (N=20)	ABCDBA (N=20)
WORLDTRACE-FIELD	0.438 / 0.696	0.573 / 0.657	0.578 / 0.649
WORLDTRACE-LANDMARK	0.719 / 0.555	0.709 / 0.551	0.735 / 0.547
WORLDTRACE	**0.791 / 0.518**	**0.807 / 0.515**	**0.808 / 0.503**

only verbatim landmarks, and our unified WORLDTRACE, which combines both under the attention-guided update of Sec. 3.5. All three share the same virtual positions, canonical keys, and six-frame attended window, so the comparison isolates what the slots store. Averaging alone does not restore recall. WORLDTRACE-FIELD stays within 0.03 DINOv2 of the sliding window in every condition of Tab. 2 (mean 0.576 versus 0.587), since averaging keeps distant history addressable but blurs the details that distinguish one place from another.

Verbatim landmarks drive recall. WORLDTRACE-LANDMARK raises mean DINOv2 to 0.715 and improves over the sliding window in every condition except Pan 360° (0.400 versus 0.406). Combining both writers is best. WORLDTRACE reaches a mean DINOv2 of 0.778 and is highest in 11 of 12 conditions, with the largest gains over WORLDTRACE-LANDMARK on long and multi-revisit rollouts such as ABA long (0.750 versus 0.632) and ABCBA (0.807 versus 0.709), suggesting that merging redundant observations frees slots for distinctive ones. The corresponding standard errors are reported in Tab. 6.

5 Conclusion

Long-horizon failure in autoregressive video world models is not only a matter of retaining more past content, but of keeping that content addressable. We show that temporal RoPE extrapolation can leave cached content unaddressable once rollouts move beyond the training horizon, and that averaging keys in the rotated space cancels what they encode. We introduce WORLDTRACE, a training-free memory framework that stores keys canonically and places every compressed slot at a distinct, in-distribution virtual position. We view memory compression as structured sparse attention, in which each slot is a structured projection of the frame history. Within this bounded memory bank, the model's own attention decides which observations to keep verbatim and which redundant ones to merge into summaries. On LOOPBENCH, WORLDTRACE improves average DINOv2 recall similarity by 32.6% over a sliding window at the same cache budget, without retraining. More broadly, these results suggest that memory for video world models should be designed around two questions, how to keep past content addressable and which visual traces to store for future queries. WORLDTRACE takes a step toward persistent interactive worlds that users can explore and revisit without losing the visual identity of previously seen places.

Acknowledgments and Disclosure of Funding

This work was supported by NVIDIA. We thank our colleagues at NVIDIA's Spatial Intelligence Lab and at Princeton University for many helpful discussions and valuable feedback.

References

[1] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[2] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025.

[3] Google DeepMind. Genie 3: A new frontier for world models. DeepMind Blog, https://deepmind.google/blog/genie-3-a-new-frontier-for-world-models, 2025.

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

[5] 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. *arXiv preprint arXiv:2508.13009*, 2025.

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

[7] Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *International Conference on Learning Representations (ICLR)*, 2025.

[8] Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

[9] Aarti Basant, Amlan Kar, Despoina Paschalidou, Fangyin Wei, Francesco Ferroni, Guillermo Garcia Cobo, Haithem Turki, Huan Ling, Jaewoo Seo, James Lucas, et al. Nvidia omnidreams: Real-time generative world model for closed-loop autonomous vehicle simulation. *arXiv preprint arXiv:2606.03159*, 2026.

[10] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H_2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[11] Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2404.14469.

[12] Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and Wen Xiao. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling. In *Conference on Language Modeling (COLM)*, 2025.

[13] Youngrae Kim, Qixin Hu, C.-C. Jay Kuo, and Peter A. Beerel. MemRoPE: Training-free infinite video generation via evolving memory tokens. *arXiv preprint arXiv:2603.12513*, 2026.

[14] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[15] Hidir Yesiltepe, Tuna Han Salih Meral, Adil Kaan Akan, Kaan Oktay, and Pinar Yanardag. ∞-RoPE: Action-controllable infinite video generation emerges from autoregressive self-rollout. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2511.20649.

[16] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations (ICLR)*, 2023.

[17] Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations (ICLR)*, 2020.

[18] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[19] Amirkeivan Mohtashami and Martin Jaggi. Landmark attention: Random-access infinite context length for transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2305.16300.

[20] Haonan Wang, Qian Liu, Chao Du, Tongyao Zhu, Cunxiao Du, Kenji Kawaguchi, and Tianyu Pang. When precision meets position: BFloat16 breaks down RoPE in long-context training. *Transactions on Machine Learning Research (TMLR)*, 2025. arXiv:2411.13476.

[21] Wan Team. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[22] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[23] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[24] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[25] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A Nyström-based algorithm for approximating self-attention. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[26] Apoorv Vyas, Angelos Katharopoulos, and François Fleuret. Fast transformers with clustered attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[27] Pavlo Vasylenko, Hugo Pitorro, André F. T. Martins, and Marcos Treviso. Long-context generalization with sparse attention. *arXiv preprint arXiv:2506.16640*, 2025.

[28] Jiaqi Leng, Xiang Hu, Junxiong Wang, Jianguo Li, Wei Wu, and Yucheng Lu. Understanding and improving length generalization in hierarchical sparse attention models. *arXiv preprint arXiv:2510.17196*, 2025.

[29] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*, 2024.

[30] Deguang Kong, Chris Ding, and Heng Huang. Robust nonnegative matrix factorization using l21-norm. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 673–682, 2011.

[31] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2309.00071.

[32] Federico Barbero, Alex Vitvitskyi, Christos Perivolaropoulos, Razvan Pascanu, and Petar Veličković. Round and round we go! what makes rotary positional encodings useful? In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.06205.

[33] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2978–2988, 2019. arXiv:1901.02860.

[34] Yuhuai Wu, Markus N. Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. In *International Conference on Learning Representations (ICLR)*, 2022.

[35] Aydar Bulatov, Yuri Kuratov, and Mikhail S. Burtsev. Recurrent memory transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[36] DeLesley Hutchins, Imanol Schlag, Yuhuai Wu, Ethan Dyer, and Behnam Neyshabur. Block-recurrent transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. arXiv:2203.07852.

[37] Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. Leave no context behind: Efficient infinite context transformers with Infini-attention. *arXiv preprint arXiv:2404.07143*, 2024.

[38] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning (ICML)*, 2020.

[39] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling (COLM)*, 2024.

[40] Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (learn at test time): RNNs with expressive hidden states. In *International Conference on Machine Learning (ICML)*, 2025.

[41] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[42] Yuxuan Tian, Zihan Wang, Yebo Peng, Aomufei Yuan, Zhiming Wang, Bairen Yi, Xin Liu, Yong Cui, and Tong Yang. KeepKV: Achieving periodic lossless KV cache compression for efficient LLM inference. In *AAAI Conference on Artificial Intelligence*, 2026. arXiv:2504.09936.

[43] Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2-bit quantization for KV cache. In *International Conference on Machine Learning (ICML)*, 2024. arXiv:2402.02750.

[44] Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. DuoAttention: Efficient long-context LLM inference with retrieval and streaming heads. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.10819.

[45] Yuhao Zhou, Sirui Song, Boyang Liu, Zhiheng Xi, Senjie Jin, Xiaoran Fan, Zhihao Zhang, Wei Li, and Xuanjing Huang. EliteKV: Scalable KV cache compression via RoPE frequency selection and joint low-rank projection. *arXiv preprint arXiv:2503.01586*, 2025.

[46] Junhui He, Junna Xing, Nan Wang, Rui Xu, Shangyu Wu, Peng Zhou, Qiang Liu, Chun Jason Xue, and Qingan Li. A^2ATS: Retrieval-based KV cache reduction via windowed rotary position embedding and query-aware vector quantization. In *Findings of the Association for Computational Linguistics (ACL Findings)*, 2025. arXiv:2502.12665.

[47] Giulio Corallo and Paolo Papotti. FINCH: Prompt-guided key-value cache compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1517–1532, 2024.

[48] NVIDIA. KVPress: A compression library for transformer KV caches. https://github.com/NVIDIA/kvpress, 2024.

[49] Weian Mao, Xi Lin, Wei Huang, Yuxin Xie, Tianfu Fu, Bohan Zhuang, Song Han, and Yukang Chen. Tri-Attention: Efficient long reasoning with trigonometric KV compression. *arXiv preprint arXiv:2604.04921*, 2026.

[50] Xuan Zhang, Cunxiao Du, Chao Du, Tianyu Pang, Wei Gao, and Min Lin. SimLayerKV: A simple framework for layer-level KV cache reduction. *arXiv preprint arXiv:2410.13846v1*, 2024.

[51] Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023.

[52] Ning Shang, Li Lyna Zhang, Siyuan Wang, Gaokai Zhang, Gilsinia Lopez, Fan Yang, Weizhu Chen, and Mao Yang. LongRoPE2: Near-lossless LLM context window scaling. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2502.20082.

[53] Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations (ICLR)*, 2022.

[54] Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. A length-extrapolatable transformer. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023. arXiv:2212.10554.

[55] Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2305.19466.

[56] Olga Golovneva, Tianlu Wang, Jason Weston, and Sainbayar Sukhbaatar. Contextual position encoding: Learning to count what's important. *arXiv preprint arXiv:2405.18719*, 2024.

[57] Ermo Hua, Che Jiang, Xingtai Lv, Kaiyan Zhang, Youbang Sun, Yuchen Fan, Xuekai Zhu, Biqing Qi, Ning Ding, and Bowen Zhou. Fourier position embedding: Enhancing attention's periodic extension for length generalization. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2412.17739.

[58] Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision (ECCV)*, 2024. arXiv:2403.13298.

[59] Xilin Wei, Xiaoran Liu, Yuhang Zang, Xiaoyi Dong, Pan Zhang, Yuhang Cao, Jian Tong, Haodong Duan, Qipeng Guo, Jiaqi Wang, Xipeng Qiu, and Dahua Lin. VideoRoPE: What makes for good video rotary position embedding? In *International Conference on Machine Learning (ICML)*, 2025. Oral; arXiv:2502.05173.

[60] Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. *Advances in Neural Information Processing Systems*, 38:15984–16009, 2026.

[61] Min Zhao, Guande He, Yixiao Chen, Hongzhou Zhu, Chongxuan Li, and Jun Zhu. RIFLEx: A free lunch for length extrapolation in video diffusion transformers. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2502.15894.

[62] Tian-Xing Xu, Zi-Xuan Wang, Guangyuan Wang, Li Hu, Zhongyi Zhang, Peng Zhang, Bang Zhang, and Song-Hai Zhang. UCM: Unifying camera control and memory with time-aware positional encoding warping for world models. *arXiv preprint arXiv:2602.22960*, 2026.

[63] Wei Yu, Runjia Qian, Yumeng Li, Liquan Wang, Songheng Yin, Sri Siddarth P. Chakaravarthy, Dennis Anthony, Yang Ye, Yidi Li, Weiwei Wan, and Animesh Garg. MosaicMem: Hybrid spatial memory for controllable video world models. *arXiv preprint arXiv:2603.17117*, 2026.

[64] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. Spotlight; arXiv:2506.08009.

[65] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[66] Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2502.06764.

[67] Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[68] Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Self-Forcing++: Towards minute-scale high-quality video generation. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2510.02283.

[69] Siqiao Huang, Jialong Wu, Qixing Zhou, Shangchen Miao, and Mingsheng Long. Vid2World: Crafting video diffusion models to interactive world models. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2505.14357.

[70] Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Lol: Longer than longer, scaling video generation to hour. *arXiv preprint arXiv:2601.16914*, 2026.

[71] Jia Li, Xiaomeng Fu, Xurui Peng, Weifeng Chen, Youwei Zheng, Tianyu Zhao, Jiexi Wang, Fangmin Chen, Xing Wang, and Hayden Kwok-Hay So. Train short, inference long: Training-free horizon extension for autoregressive video generation. *arXiv preprint arXiv:2602.14027*, 2026.

[72] 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.

[73] 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.

[74] Hanmo Chen, Chenghao Xu, Xu Yang, Xuan Chen, and Cheng Deng. Past- and future-informed kv cache policy with salience estimation in autoregressive video diffusion. *arXiv preprint arXiv:2601.21896*, 2026.

[75] Haodong Li, Shaoteng Liu, Zhe Lin, and Manmohan Chandraker. Rolling sink: Bridging limited-horizon training and open-ended testing in autoregressive video diffusion. *arXiv preprint arXiv:2602.07775*, 2026.

[76] 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.

[77] Yuexiao Ma, Xuzhe Zheng, Jing Xu, Xiwei Xu, Feng Ling, Xiawu Zheng, Huafeng Kuang, Huixia Li, Xing Wang, Xuefeng Xiao, Fei Chao, and Rongrong Ji. Flow caching for autoregressive video generation. *arXiv preprint arXiv:2602.10825*, 2026.

[78] Jintao Chen, Chengyu Bai, Junjun Hu, Xinda Xue, and Mu Xu. Grounded forcing: Bridging time-independent semantics and proximal dynamics in autoregressive video synthesis. *arXiv preprint arXiv:2604.06939*, 2026.

[79] Yang Yang, Tianyi Zhang, Wei Huang, Jinwei Chen, Boxi Wu, Xiaofei He, Deng Cai, Bo Li, and Peng-Tao Jiang. Anchor forcing: Anchor memory and tri-region RoPE for interactive streaming video diffusion. *arXiv preprint arXiv:2603.13405*, 2026.

[80] Shuo Chen, Cong Wei, Sun Sun, Ping Nie, Kai Zhou, Ge Zhang, Ming-Hsuan Yang, and Wenhu Chen. Context forcing: Consistent autoregressive video generation with long context. *arXiv preprint arXiv:2602.06028*, 2026.

[81] Ruibin Li, Tao Yang, Fangzhou Ai, Tianhe Wu, Shilei Wen, Bingyue Peng, and Lei Zhang. Long-horizon streaming video generation via hybrid attention with decoupled distillation. *arXiv preprint arXiv:2604.10103*, 2026.

[82] Wuyang Li, Wentao Pan, Po-Chien Luan, Yang Gao, and Alexandre Alahi. Stable video infinity: Infinite-length video generation with error recycling. *arXiv preprint arXiv:2510.09212*, 2025.

[83] 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.

[84] 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 (ICLR)*, 2026. arXiv:2509.25161.

[85] Sebastian Stapf, Pablo Acuaviva, Aram Davtyan, and Paolo Favaro. Composition of memory experts for diffusion world models. In *International Conference on Learning Representations (ICLR)*, 2026. OpenReview: https://openreview.net/forum?id=sUEdpZCHdp.

[86] Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2506.05284, project page https://spmem.github.io.

[87] Shengqu Cai, Ceyuan Yang, Lvmin Zhang, Yuwei Guo, Junfei Xiao, Ziyan Yang, Yinghao Xu, Zhenheng Yang, Alan Yuille, Leonidas Guibas, Maneesh Agrawala, Lu Jiang, and Gordon Wetzstein. Mixture of contexts for long video generation. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2508.21058, OpenReview: https://openreview.net/forum?id=y6XJZlEC2x.

[88] Yuchao Gu, Weijia Mao, and Mike Zheng Shou. Long-context autoregressive video modeling with next-frame prediction. *arXiv preprint arXiv:2503.19325*, 2025.

[89] Yuta Oshima, Yusuke Iwasawa, Masahiro Suzuki, Yutaka Matsuo, and Hiroki Furuta. WorldPack: Compressed memory improves spatial consistency in video world modeling. *arXiv preprint arXiv:2512.02473*, 2025.

[90] 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.

[91] Yifei Yu, Xiaoshan Wu, Xinting Hu, Tao Hu, Yang-Tian Sun, Xiaoyang Lyu, Bo Wang, Lin Ma, Yuewen Ma, Zhongrui Wang, and Xiaojuan Qi. VideoSSM: Autoregressive long video generation with hybrid state-space memory. *arXiv preprint arXiv:2512.04519*, 2025.

[92] Haoyi Zhu, Haozhe Liu, Yuyang Zhao, Tian Ye, Junsong Chen, Jincheng Yu, Tong He, Song Han, and Enze Xie. SANA-WM: Efficient minute-scale world modeling with hybrid linear diffusion transformer. *arXiv preprint arXiv:2605.15178*, 2026.

[93] Boxun Xu, Yuming Du, Zichang Liu, Siyu Yang, Ziyang Jiang, Siqi Yan, Rajasi Saha, Albert Pumarola, Wenchen Wang, and Peng Li. Sparse forcing: Native trainable sparse attention for real-time autoregressive diffusion video generation. *arXiv preprint arXiv:2604.21221*, 2026.

[94] Ryan Po, Yotam Nitzan, Richard Zhang, Berlin Chen, Tri Dao, Eli Shechtman, Gordon Wetzstein, and Xun Huang. Long-context state-space video world models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8733–8744. IEEE, 2025.

[95] 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.

[96] Xiaofeng Mao, Shaohao Rui, Kaining Ying, Bo Zheng, Chuanhao Li, Mingmin Chi, and Kaipeng Zhang. PackForcing: Short video training suffices for long video sampling and long context inference. *arXiv preprint arXiv:2603.25730*, 2026.

[97] Xinhang Gao, Junlin Guan, Shuhan Luo, Wenzhuo Li, Guanghuan Tan, and Jiacheng Wang. Memcam: Memory-augmented camera control for consistent video generation. *arXiv preprint arXiv:2603.26193*, 2026.

[98] Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. WorldMem: Long-term consistent world simulation with memory. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[99] Yanjun Guo, Zhengqiang Zhang, Pengfei Wang, Xinyue Liang, Zhiyuan Ma, and Lei Zhang. Memorize when needed: Decoupled memory control for spatially consistent long-horizon video generation. *arXiv preprint arXiv:2604.18215*, 2026.

[100] 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. *arXiv preprint arXiv:2604.08995*, 2026.

[101] 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 (NeurIPS)*, 2025. Spotlight; arXiv:2504.12626.

[102] Roberto Henschel, Levon Khachatryan, Hayk Poghosyan, Daniil Hayrapetyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. StreamingT2V: Consistent, dynamic, and extendable long video generation from text. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2403.14773.

[103] Xindi Wu, Uriel Singer, Zhaojiang Lin, Andrea Madotto, Xide Xia, Yifan Xu, Paul Crook, Xin Luna Dong, and Seungwhan Moon. Corgi: Cached memory guided video generation. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4585–4594. IEEE, 2025.

[104] Runjia Li, Philip Torr, Andrea Vedaldi, and Tomas Jakab. VMem: Consistent interactive video scene generation with surfel-indexed view memory. In *International Conference on Computer Vision (ICCV)*, 2025. arXiv:2506.18903.

[105] 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.

[106] Taiye Chen, Xun Hu, Zihan Ding, and Chi Jin. Learning world models for interactive video generation. *arXiv preprint arXiv:2505.21996*, 2025. Project page https://sites.google.com/view/vrag.

[107] Haoyi Duan, Hong-Xing Yu, Sirui Chen, Li Fei-Fei, and Jiajun Wu. WorldScore: A unified evaluation benchmark for world generation. In *International Conference on Computer Vision (ICCV)*, 2025.

[108] Yixuan Ye, Xuanyu Lu, Yuxin Jiang, Yuchao Gu, Rui Zhao, Qiwei Liang, Jiachun Pan, Fengda Zhang, Weijia Wu, and Alex Jinpeng Wang. MIND: Benchmarking memory consistency and action control in world models. *arXiv preprint arXiv:2602.08025*, 2026.

[109] Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Lulu Gu, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, Yu Qiao, and Ziwei Liu. VBench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

[110] Kewei Lian, Shaofei Cai, Yilun Du, and Yitao Liang. Toward memory-aided world models: Benchmarking via spatial consistency. *arXiv preprint arXiv:2505.22976*, 2025. Loop-based Minecraft navigation benchmark for spatial consistency in world models.

[111] Jonathan Lorraine, Nihesh Anderson, Chansoo Lee, Quentin De Laroussilhe, and Mehadi Hassen. Task selection for AutoML system evaluation. *arXiv preprint arXiv:2208.12754*, 2022.

[112] Jonathan Lorraine, Kevin Xie, Xiaohui Zeng, Chen-Hsuan Lin, Towaki Takikawa, Nicholas Sharp, Tsung-Yi Lin, Ming-Yu Liu, Sanja Fidler, and James Lucas. ATT3D: Amortized text-to-3D object synthesis. In *International Conference on Computer Vision (ICCV)*, 2023.

[113] Kevin Xie, Jonathan Lorraine, Tianshi Cao, Jun Gao, James Lucas, Antonio Torralba, Sanja Fidler, and Xiaohui Zeng. LATTE3D: Large-scale amortized text-to-enhanced 3D synthesis. In *European Conference on Computer Vision (ECCV)*, 2024.

[114] Jesse Bettencourt, Xindi Wu, Matan Atzmon, James Lucas, and Jonathan Lorraine. Variance reduction for expectations with diffusion teachers. *arXiv preprint arXiv:2605.21489*, 2026.

[115] Jessie Richter-Powell, Antonio Torralba, and Jonathan Lorraine. Score distillation sampling for audio: Source separation, synthesis, and beyond. In *ICML Workshop on AI Heard That!*, 2025. arXiv:2505.04621.

[116] Zhengyi Wang, Jonathan Lorraine, Yikai Wang, Hang Su, Jun Zhu, Sanja Fidler, and Xiaohui Zeng. LLaMA-Mesh: Unifying 3D mesh generation with language models. *arXiv preprint arXiv:2411.09595*, 2024.

[117] Xindi Wu, Despoina Paschalidou, Jun Gao, Antonio Torralba, Laura Leal-Taixé, Olga Russakovsky, Sanja Fidler, and Jonathan Lorraine. Motion attribution for video generation. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2601.08828.

Appendices

A Notation **17**

B Structured Attention Approximations through Selection and Averaging **18**
B.1 Full and Projected Attention . . . 18
B.2 Structured Attention Approximations Unified by P . . . 19
B.3 Approximation Quality and Error Decomposition . . . 19
B.4 Reduction to Nonnegative Matrix Factorization . . . 20

C Additional Properties **20**

D Related Work **21**
D.1 Memory-Augmented Transformers . . . 21
D.2 KV Cache Compression for Large Language Models . . . 21
D.3 Position Extrapolation for RoPE . . . 21
D.4 Autoregressive Video World Models . . . 22
D.5 Memory and KV Cache for Video World Models . . . 22

E Evaluation Metrics **23**

F Additional Experimental Results **23**
F.1 Scene Consistency Episodic Recall Sweep . . . 23
F.2 Cross-Architecture Experiments . . . 23
F.3 Attention Mismatch of Compressed Slots . . . 24

G The LOOPBENCH Memory Benchmark **24**
G.1 Task Design and Evaluation . . . 24
G.2 Difficulty Axes . . . 26

H Implementation Details **27**
H.1 Model Architecture and Self-Forcing Background . . . 27
H.2 WORLDTRACE Algorithm . . . 27
H.3 Hyperparameters, Compute, and Evaluation Protocol . . . 27

I Discussion **28**
I.1 Limitations . . . 28
I.2 Future Directions . . . 28
I.3 Broader Impact . . . 29

A Notation

Table 3: **Glossary and notation.**

Symbol	Description
Cache Structure	
N	Total number of AR chunks in a rollout (generation length, F latent frames per chunk)
N_s	Number of summary slots in the KV cache (one latent-frame-sized KV payload each)
N_r	Number of recent-window slots (one verbatim latent frame each)
L_{attn}	Local attention / inference KV window in latent frames ($N_s + N_r = L_{\text{attn}}$)
L_{train}	Training-time KV-cache extent in AR chunks during Self-Forcing rollouts (distinct from the local attention window L_{attn})
F	Latent frames per AR chunk
M	Source frames compressed into one summary slot under WORLDTRACE-FIELD
T	Number of past latent frames in the full-history reference, rather than the number retained by the bounded cache
s	Summary slot index, $s = 0, \ldots, N_s{-}1$ (0 = oldest)
$\mathcal{R}$	Recent-window cache containing verbatim KVs of the newest N_r latent frames
$\mathcal{S}$	Summary cache containing N_s compressed slots indexed by s
Indices	
f	RoPE frequency-pair index
m	Source-frame iteration index (inside $\sum_{m=1}^{M}$)
Virtual Position Assignment (Slot Indexing)	
q	Absolute timestamp of the current query frame. In the attention-approximation analysis, q instead denotes a query vector
k	Absolute timestamp of a cached key frame
$t^v_{\min}$	In-distribution lower bound $\max(0,\ q - \Delta t_{\text{train}})$
$t^v_{\max}$	In-distribution upper bound for summary slots $q - N_r$
t^v	Virtual position
t^v_s	Virtual position of slot s, given by $q - (L_{\text{attn}}{-}1{-}s)$, for the populated full-cache regime
LOOPBENCH *Benchmark*	
K	Number of intermediate waypoints between departure and return to scene A
R	Pattern-specific revisit count, measuring returns to A in ABABA and visits to B in ABCBA and ABCDBA (all shown with $R{=}2$)

Table 3: **Glossary and notation (continued).**

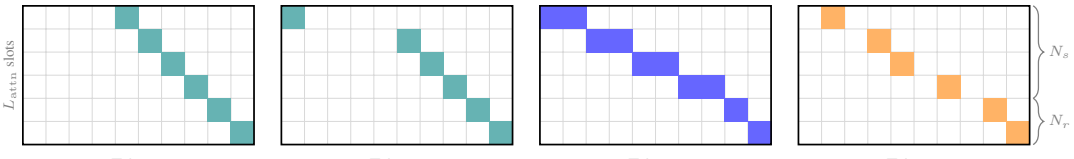
Symbol	Description
RoPE Parameters	
θ	RoPE base parameter (θ=10000 for MG2)
θ_f	RoPE angular frequency for temporal head-dimension pair f ($\theta_f = \theta^{-f/c_t}$)
c_t, c_h, c_w	Number of temporal, height, and width RoPE complex pairs per head
$R(\alpha)$	Rotation matrix by angle α (applied per frequency pair)
Key / Query Variants	
$K^f_{t_m}$	RoPE-rotated key at absolute position t_m, frequency pair f
K^f_m	Canonical (unrotated) key content $R(-\theta_f t_m)\ K^f_{t_m}$
$\bar{K}^f$	Canonical mean across M source frames
$\bar{K}^f_{\text{naive}}$	Naive RoPE-space average of M rotated keys
$K^f_{\text{field}}(t^v)$	WORLDTRACE-FIELD compressed key at virtual position t^v
$K^f_{\text{land}}(t^v)$	WORLDTRACE-LANDMARK key re-encoded at virtual position t^v from frozen canonical content
t_{ℓ^*}	Original timestamp of a selected landmark frame
$K^f_{t_{\ell^*}}$	RoPE-rotated landmark source key at t_{ℓ^*}, frequency pair f
τ	Scene-boundary threshold on the cosine distance between canonical keys of consecutive frames
Q^f_q	RoPE-rotated query at q, frequency pair f
Attention Quantities	
$\delta_{q,k}$	Query-key temporal offset $q - k$
Δt_{train}	Largest trained temporal offset (Δt_{train}=5 for MG2). An in-distribution cache requires $L_{\text{attn}} - 1 \le \Delta t_{\text{train}}$
$a^f_{q,k}$	Attention-score contribution from frequency pair f at offset $\delta_{q,k}$
$A^f_{q,k}$	Query-key content term in canonical coordinates for $a^f_{q,k}$
α_q	Full-reference attention weights over T frames for query vector q. The compression analysis uses a repositioned reference
$\mathcal{Q}$	Query set in the aggregate mismatch objective (App. B)
$\mathbf{K}_{\text{hist}}$	Canonical source keys of the T past frames at a fixed layer, head, and spatial token
P, P_i	Projection matrix $P \in \mathbb{R}^{L_{\text{attn}} \times T}$ and its i-th row, the weights with which cache slot i combines past frames
Δ_{attn}	Attention Mismatch between uncompressed and compressed attention (App. E)

B Structured Attention Approximations through Selection and Averaging

With full attention, each new chunk attends to all T past frames, so attention cost and the KV cache grow as $O(T)$, and video world models typically cap this with a sliding window over the most recent frames. WORLDTRACE instead compresses the full history to $L_{\text{attn}} \ll T$ slots through a projection matrix $P \in \mathbb{R}^{L_{\text{attn}} \times T}$, whose rows define which frames contribute to each slot. Throughout this section, K denotes the canonical (unrotated) key matrix on which P acts, and the slot-rank rotation of Def. 1 is applied afterwards at attention time.

B.1 Full and Projected Attention

For a single query q, the standard attention output is $\mathcal{A}(q, K, V) = \text{softmax}(q\,K^\top)\,V = \alpha_q\,V$, a weighted sum over all T value vectors with $\alpha_q \in \Delta_T$, where Δ_T is the probability simplex over the T past frames (non-negative weights summing to one). The *projected* attention approximation replaces the T-token sequence with $L_{\text{attn}} \ll T$ prototype tokens (one per compressed slot) defined by a row-stochastic matrix $P \in \mathbb{R}^{L_{\text{attn}} \times T}$, meaning each row of P is itself a distribution over past



(a) Sliding window **(b)** + attention sink **(c)** WORLDTRACE-FIELD **(d)** WORLDTRACE-LANDMARK

Figure 5: **Projection matrices P for the four cache structures**. **(a)** *Sliding window* retains the last L_{attn} frames verbatim. **(b)** *Sliding window with attention sink* retains the initial frame as an attention sink and fills the remaining slots with verbatim recent frames. **(c)** WORLDTRACE-FIELD keeps the newest frames verbatim in recent rows and averages a group of earlier frames in each summary slot. **(d)** WORLDTRACE-LANDMARK selects scene-entry frames with one-hot summary rows and keeps recent rows verbatim.

frames. The projected attention is:

$$\widehat{\mathcal{A}}(q, K, V;\, P) \;=\; \text{softmax}(q\,(PK)^\top)\,PV, \tag{5}$$

where $\hat{\alpha}_q = \text{softmax}(q\,(PK)^\top) \in \Delta_{L_{\text{attn}}}$ are the prototype attention weights, now a distribution over the L_{attn} compressed slots rather than the T frames, and P_iV is the compressed value for slot i.

Relation to prior structured-attention approximations. Eq. (5) is the low-rank factorization of attention studied by efficient transformers, whose projections are learned end-to-end [24, 25, 26, 16, 27, 28]. WORLDTRACE instead applies the projection *training-free* to a pretrained model's KV cache. Because of RoPE, P acts in the canonical key domain to avoid phase cancellation (Sec. 2.2), and the compressed slots are re-encoded at slot-rank virtual positions (Def. 1) to stay addressable. The row structure of P (uniform averaging for WORLDTRACE-FIELD, one-hot selection for WORLDTRACE-LANDMARK) determines whether compression better approximates group-level or frame-level attention, as formalized below.

B.2 Structured Attention Approximations Unified by P

The four cache structures (Fig. 5) in our experiments correspond to four structured choices of P.

Sliding window attention. Here $P = [\,0 \;\; I_{L_{\text{attn}}}\,]$ makes each row a one-hot vector selecting one of the most recent L_{attn} frames, discarding all earlier history (Fig. 5a). Adding an *attention sink anchor* is a simple modification (Fig. 5b) in which row 0 is pinned to the first (sink) frame while the remaining $L_{\text{attn}}-1$ rows select the most recent $L_{\text{attn}}-1$ frames verbatim, always keeping a global context token in cache regardless of generation length. This anchor mitigates the collapse of the attention distribution onto only local tokens at long generation horizons [29].

WORLDTRACE writers as grouped and sparse attention. In both writers, rows for the N_r recent slots are identity rows as above. For WORLDTRACE-FIELD, each remaining row is a uniform average over the M frames assigned to its summary slot (Fig. 5c), with $P_{i,t} = 1/M$ if frame t is assigned to slot i and 0 otherwise. For WORLDTRACE-LANDMARK, summary rows are one-hot selectors at the detected scene-entry frames (Fig. 5d), so the row storing the landmark at timestamp t_{ℓ^*} has $P_{i,t} = 1$ for $t = t_{\ell^*}$ and 0 otherwise, copying the frozen key $P_iK = K[t_{\ell^*}]$ of Eq. (4). The effective attention weights $\hat{\alpha}_q P$ are therefore supported only on retained frames for WORLDTRACE-LANDMARK, giving structured sparse attention, and spread with equal weights within each temporal group for WORLDTRACE-FIELD, giving grouped attention.

B.3 Approximation Quality and Error Decomposition

The cache contents, and hence P, change as generation proceeds, since the recent window slides and summary rows are filled or overwritten. All statements in this and the following subsections therefore refer to one generation step at a time, where T and P denote the history length and the projection realized by the cache at the current step, and $\mathcal{Q}$ denotes the queries of the chunk being generated (one attention row per query token and head). Every bound holds at each step separately, and no single static P is claimed to serve all steps.

Two sources of error. The compressed cache differs from an unbounded cache in two independent ways. Keys are *repositioned* (re-encoded at slot-rank virtual positions rather than their original timestamps), and keys are *compressed* (merged through P). Writing $\mathcal{A}_{\text{orig}}$ for full attention with keys

at their original positions, $\mathcal{A}$ for full attention with keys re-encoded at virtual positions, and $\widehat{\mathcal{A}}$ for the projected attention of Eq. (5) on the re-encoded cache, the total error decomposes as:

$$\|\widehat{\mathcal{A}} - \mathcal{A}_{\text{orig}}\| \;\le\; \underbrace{\|\mathcal{A} - \mathcal{A}_{\text{orig}}\|}_{\text{repositioning}} \;+\; \underbrace{\|\widehat{\mathcal{A}} - \mathcal{A}\|}_{\text{compression}}.$$

The repositioning term is not an artifact of compression but the trade-off for addressability. Beyond the training horizon the original offsets are out-of-distribution and attention degrades outright (Sec. 2.1), so slot-rank re-encoding trades an uncontrolled OOD failure for a bounded, in-distribution position shift. We do not bound this term analytically. The compression bound below does not quantify the effect of repositioning. Prop. 1 below bounds the compression term. All statements in this section are relative to the repositioned reference $\mathcal{A}$.

For a single query q, let $\hat{\alpha}_q = \mathrm{softmax}(q\,(PK)^\top) \in \Delta_{L_{\text{attn}}}$ be the prototype attention weights under the compressed cache, and define the output error $r_q(P) = \widehat{\mathcal{A}}(q, K, V;\, P) - \mathcal{A}(q, K, V)$.

Proposition 1 (Approximation bound)**.**

$$\|r_q(P)\|_2 \;\le\; \underbrace{\|\hat{\alpha}_q P - \alpha_q\|_1}_{\textit{distribution mismatch}} \cdot \|V\|_{\infty,\text{row}},$$

where $\|V\|_{\infty,\text{row}} = \max_t \|V[t]\|_2$.

Proof. By associativity $\widehat{\mathcal{A}} = \hat{\alpha}_q(PV) = (\hat{\alpha}_q P)V$, so $r_q(P) = (\hat{\alpha}_q P - \alpha_q)V = \sum_t c_t\, V[t]$ with $c = \hat{\alpha}_q P - \alpha_q$. Applying the vector-valued Hölder inequality (ℓ_1/ℓ_∞ duality) to the rows of V, we obtain $\|r_q(P)\|_2 \le \|c\|_1\, \|V\|_{\infty,\text{row}}$. □

The bound motivates minimizing distribution mismatch $\|\hat{\alpha}_q P - \alpha_q\|_1$ independently of value geometry. For WORLDTRACE-FIELD, linearity of the dot product ensures the compressed key $K^f_{\text{field}}(t^v)$ reproduces the mean attention score of its source keys at the shared virtual position, limiting per-slot mismatch when query preferences vary smoothly across a group. For WORLDTRACE-LANDMARK, a verbatim scene-entry key keeps the mismatch small for queries concentrated on that event.

B.4 Reduction to Nonnegative Matrix Factorization

Nonnegative matrix factorization (NMF) approximates a nonnegative matrix as the product of two low-rank nonnegative factors [18], which form a compact summary of the original data. A relaxation of attention compression through P has this form. Treat $\hat{\alpha}_q \in \Delta_{L_{\text{attn}}}$ as a free variable (decoupling it from the softmax in Eq. (5)), and define $\mathbf{A} \in \mathbb{R}^{|\mathcal{Q}| \times T}$ as the matrix whose rows are the full attention distributions α_q, one row per query in $\mathcal{Q}$, and $\mathbf{B} \in \mathbb{R}^{|\mathcal{Q}| \times L_{\text{attn}}}$ as the matrix whose rows are the corresponding prototype weights $\hat{\alpha}_q$. The aggregate distribution-mismatch objective then becomes:

$$\min_{\mathbf{B}\ge 0,\; P \ge 0} \|\mathbf{B}P - \mathbf{A}\|_1 \quad \text{subject to row-sum constraints on } \mathbf{B} \text{ and } P,$$

which is NMF with an ℓ_1 fidelity term, in the spirit of robust NMF formulations [30]. This relaxation serves the analysis only, because at inference $\hat{\alpha}_q$ is not free but determined by P through the softmax, $\hat{\alpha}_q = \mathrm{softmax}(q\,(PK)^\top)$ for each query q and key matrix K. Under this factorization, WORLDTRACE-FIELD rows are group means, like k-means centroids, and WORLDTRACE-LANDMARK rows are retained exemplars, like k-medoids. The unified WORLDTRACE memory mixes both row types in one P, using attention mismatch to choose which history to average and which to keep verbatim.

C Additional Properties

Temporal RoPE at long horizons. We provide temporal-RoPE background for the addressability failure in Sec. 2.1. MG2-1.3B inherits the 3D RoPE split of Wan 2.1 [21], with $2c_t$=44 temporal and $2c_h$=$2c_w$=42 height and width dimensions per head and base $\theta = 10000$, so the temporal frequencies $\theta_f = \theta^{-f/c_t}$ range from 1.0 to about 1.5×10^{-4}. Frequency f enters through the rotation $R(\theta_f \delta_{q,k})$, and following YaRN's NTK-by-parts view [31], we compare each component's wavelength to the trained context. At $\Delta t_{\text{train}} = 5$, components $f \ge 10$ rotate by less than 0.08 rad, while at $|\delta_{q,k}| = 30$ the three fastest exceed 3π rad (Fig. 6), and an unbounded rollout pushes ever

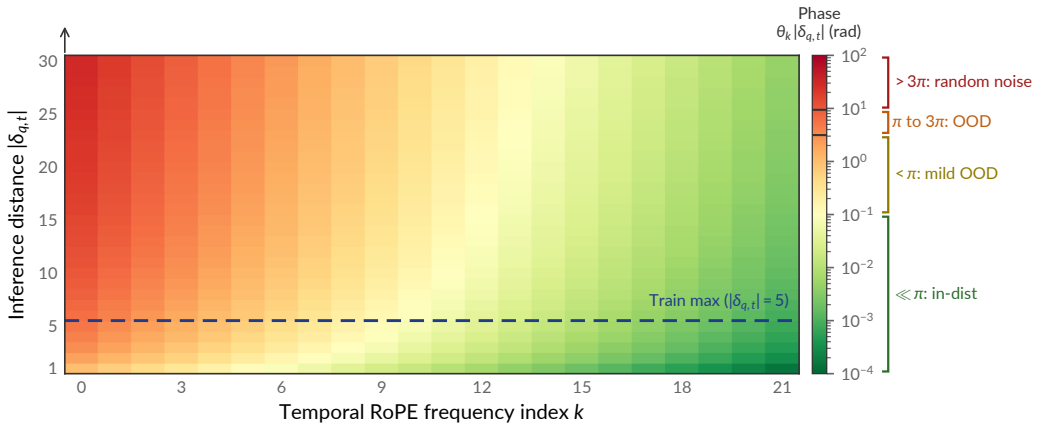


Figure 6: **Temporal RoPE rotation angle per frequency.** Per-frequency unwrapped phase $\theta_k|\delta_{q,t}|$ on MG2-1.3B. In this figure, k indexes temporal frequencies and $\delta_{q,t} = q - t$ is the query-key offset. Color shows phase magnitude on a log scale, and the dashed line marks the training maximum $|\delta_{q,t}|$=5. The bracket labels are schematic phase bands, not empirically established thresholds for attention or recall failure.

more components beyond π. This describes unwrapped phase, not measured attention quality, and low-frequency channels can also fail under extrapolation [32]. WORLDTRACE keeps all temporal offsets within the trained window.

D Related Work

D.1 Memory-Augmented Transformers

Memory-augmented transformers pair a recent verbatim window with a compressed record of older context, realized as segment-level recurrence with relative positions [33], a compressed memory of older tokens [17], external kNN memory [34], learned memory tokens passed between segments [35, 36], and a fused sliding window plus compressive memory [37]. WORLDTRACE adopts the same two-tier structure, with a recent verbatim window plus compressed summary slots. Landmark Attention [19] shares vocabulary with WORLDTRACE-LANDMARK, but its landmarks are *trained* representatives that gate attention to off-cache blocks at 1D position, whereas WORLDTRACE-LANDMARK is training-free, stores verbatim canonical-frame keys inside the $O(1)$ cache, and freezes them against unrotate-and-rerotate drift in 3D RoPE. The broader spectrum toward compressed state spans linear attention [38], Mamba [39], and Test-Time Training [40]. We target the fixed-budget, training-free setting in AR video world models.

D.2 KV Cache Compression for Large Language Models

The three main families of LLM KV-cache methods are *window/eviction* [41, 29, 10, 11, 12], *merging* [16, 42], and *quantization* [43], the last orthogonal to the position-OOD failure we target. DuoAttention [44] splits heads into retrieval and streaming families, parallel to our recent-window and summary-slot split. These methods change *content* only, picking which tokens to keep or merge rather than which virtual position a summary spanning several merged frames should occupy. The closest exceptions touch per-token positional structure [45, 46] or unrotate and rerotate individually retained keys [47, 48], but none reassign summary-slot positions. Our setting also differs from LLM compression in that keys accumulate across independently denoised AR video chunks, so importance needs to be ranked under denoising-timestep-conditioned queries. Under such queries, angular-concentration scores [49] lose the stable pre-RoPE key geometry they assume, and layer-budget reallocations [12, 50] tune which layers keep tokens rather than how summary slots are addressed. WORLDTRACE-FIELD instead averages canonical keys and assigns slot-rank positions.

D.3 Position Extrapolation for RoPE

RoPE [14] degrades past trained relative offsets. Existing fixes span index rescaling [51, 31, 52], alternative distance biases [53, 54], position-free training [55], and frequency- or content-aware encodings [56, 57], while vision and video extensions add 2D/3D splits [58, 59], camera-relative

encodings [60], and training-free diffusion extrapolation [61]. All of this remaps positions for individually retained tokens or fixed-length inputs. Our cache instead needs to be *compressed*, since a fixed $O(1)$ budget cannot keep every past frame verbatim. Each merged summary then needs an explicit virtual position, a quantity that does not arise when tokens keep separate identities, and WORLDTRACE assigns slot-rank offsets for that role without retraining. Wang et al. [20] tie position to numerical precision, relevant to the frozen-key discussion in Sec. 3.4. UCM [62] and MosaicMem [63] warp 3D positions for camera-controlled worlds via trained geometric operators.

D.4 Autoregressive Video World Models

Interactive AR video world models [3, 5, 6, 7] generate via chunk-wise diffusion prediction, dominantly under the Self-Forcing paradigm [64, 65, 66, 67], which aligns training rollouts with the model's own KV cache and inherits a local attention window not sized for arbitrarily long inference. Training-time horizon extensions [68, 69] and memory-specific methods (Sec. D.5) address this limit through finetuning. Foundation backbones [21] and the world-model lineage from Ha and Schmidhuber [1] through Dreamer [2] to closed-loop driving simulators [9] motivate interactive rollouts beyond passive generation.

D.5 Memory and KV Cache for Video World Models

Inference-time KV cache. Closest to WORLDTRACE, inference-time methods modify cache read/write without retraining. *Position-side* fixes remap or cap temporal offsets via block-relative RoPE with aggressive flushing [15], inference-time position adjustment [70, 71], or per-token virtual positions with cross-frame decay [72]. *Content-side* writers decide what to merge, prune, or retain, through re-aligned sink tokens [73], dual-rate EMA summaries [13], importance pruning [74, 75], and temporal-correspondence merging [76]. FlowCache [77] targets per-step compute rather than long-horizon recall, but its recent-summary structure parallels ours. MemRoPE already jointly uses canonical memory updates and contiguous online indices [13]. WORLDTRACE studies canonical averaging and retained-frame memory for episodic recall under a fixed attended-cache size, evaluated with controlled scene revisits on LOOPBENCH.

Training-time extensions. Training-time methods finetune the generator itself for longer effective context. One line pairs partitioned caches or memory hierarchies with RoPE re-alignment or long-context teacher supervision [78, 79, 80, 81, 82, 83, 84]. Memory-decomposition approaches assign different parts of history to dedicated modules [85, 86, 87]. Architectural long-horizon models train for extended rollouts rather than patching a fixed checkpoint [88, 89, 90, 91, 92], and Sparse Forcing [93] learns block-sparse attention with persistent anchors end-to-end. Po et al. [94] extend the effective context of a video world model with a trained state-space memory and evaluate it on Memory Maze with a spatial retrieval task, where the model is given a trajectory as context and then backtracks the exact action sequence to the starting position. WORLDTRACE needs no retraining, restoring addressability via cache read/write on a fixed checkpoint.

Geometry-conditioned memory. Trained methods decide what to keep in memory based on camera pose, actions, or scene geometry, drawing on pose- and action-annotated compressed latents [95], warped positions or latents for camera-controlled worlds [63, 62, 96], co-visibility-based selection [97], pose-tagged frame re-injection [98], and camera-aware memory gating [99]. WORLDTRACE is geometry-free, replacing pose-, action-, or co-visibility-driven memory selection with slot-rank positions and canonical cache writes.

External memory banks and retrieval. Rather than compressing inside the KV cache, these systems store or retrieve frames outside it and re-inject them via auxiliary attention or re-encoding, through FOV-based frame selection [100], importance-weighted frame contexts [101], first-chunk appearance anchoring [102], cached latent banks [103, 104, 105], and global-state retrieval [106]. The evaluated WORLDTRACE variant also uses a bank outside the attended cache. It retains frame KVs and selects an entry using the model's attention queries (Tab. 2). The bank and query storage add memory beyond the fixed attended-cache size.

Comparison. In Tab. 4, we qualitatively compare memory approaches by considering whether positional OOD is fixed, whether memory is bounded, whether intermediate history is preserved, and whether retraining is required. For evaluation, WorldScore [107], MIND [108], and VBench-2.0 [109] assess long-horizon quality, and Lian et al. [110] targets revisit consistency in Minecraft. The Memory Maze retrieval task of Po et al. [94] is the closest published revisit test to ours and scores

Table 4: **Memory approach comparison.** Representative baselines and inference-time cache methods on four axes. Memory complexity in this table refers to the attended KV cache.

Method	Fixes pos. OOD?	Bounded attended cache?	Preserves history?	Training-free?
Full KV	No	No ($O(N)$)	Yes (exact)	Yes
Sliding window	N/A (no long-range reads)	Yes ($O(1)$)	Recent only	Yes
KV Flush + Block-relative [15]	Yes	Yes ($O(1)$)	No	Yes
Naive avg. + Block-relative	Partial (slots collapse)	Yes ($O(1)$)	No (distorted)	Yes
MemRoPE [13]	Yes	Yes ($O(1)$)	EMA-diluted	Yes
Deep Forcing [73]	Partial	Yes ($O(1)$)	Sink plus pruned recent	Yes
WORLDTRACE (ours)	**Yes**	**Yes ($O(1)$)**	**Yes**	**Yes**

Table 5: **Evaluation metrics.** Higher is better for metrics marked ↑, and lower is better for those marked ↓.

Metric	**Evaluation focus**	**Better**	**Definition**
Scene Consistency (DINOv2)	Primary recall	↑	DINOv2 feature cosine similarity across revisits.
Return LPIPS	Primary recall	↓	Perceptual distance between the same revisited views.
Attention Mismatch (Δ_{attn})	Compression diagnostic	↓	Attention-distribution difference induced by compression.

a backtracked trajectory against the recorded ground-truth frames, whereas LOOPBENCH sweeps loop topology, rollout length, camera orientation, and multi-revisit patterns and scores each return against the model's own earlier generation at matched poses, so it needs no ground-truth footage of the revisited scene.

E Evaluation Metrics

We evaluate memory through recall across revisits and attention fidelity under compression (Tab. 5). Recall compares generated views from two visits to the same location, using the earlier view as the reference. Attention fidelity measures the effect of compression under matched queries and history.

Recall. Scene Consistency is the cosine similarity between DINOv2 [22] features of the frames generated at the revisited location (e.g., A in ABA) on the first visit and on the return (higher is better). Return LPIPS [23] is the perceptual distance between the same two frames (lower is better). DINOv2 captures overall scene content and structure, while LPIPS is more sensitive to appearance and spatial layout. Neither alone establishes exact reconstruction, and viewpoint differences can affect both. We always use the frame that ends the return, without searching for a best-matching return frame. For DINOv2, we use the ViT-S/14 class-token embedding on frames resized to 252×448 with standard ImageNet normalization. For LPIPS, we use the VGG backbone on full-resolution frames.

Compression fidelity. Attention Mismatch compares uncompressed and compressed attention distributions under identical queries and source history, mapping compressed-slot weights back to the observations they represent. This controlled comparison measures how compression changes attention allocation, rather than comparing attention from independently diverging rollouts. Lower mismatch indicates greater attention fidelity but does not guarantee improved recall. We therefore use it as a compression diagnostic alongside the two recall metrics. Measurements are in App. F.3.

F Additional Experimental Results

F.1 Scene Consistency Episodic Recall Sweep

We report DINOv2 and LPIPS across $N \in \{16, 32, 48, 64, 128\}$ on ABA loops (N_s=4, N_r=2) in Tab. 7. At N=128, the sliding window reaches 0.251 DINOv2, while WORLDTRACE reaches 0.698. WORLDTRACE also keeps lower LPIPS than the sliding window at every evaluated horizon.

F.2 Cross-Architecture Experiments

Tab. 8 repeats LOOPBENCH on LingBot-World [4]. It is a 14B diffusion transformer with Plücker camera conditioning, whereas MG2 is a dense 1.3B model without it, so this comparison varies both scale and architecture. The camera loops are built on the release's example trajectory with the same

Table 6: **LOOPBENCH results with standard errors.** DINOv2 (↑) / LPIPS (↓), mean$_{\pm \text{s.e.m.}}$ over n=51 rollouts. All methods share the six-frame attended window (N_s=4, N_r=2).

Tier 1, topology	ABA (K=1, N=16)	ABCA (K=2, N=17)	ABCDA (K=3, N=16)
Sliding window	$0.577_{\pm 0.026}$ / $0.643_{\pm 0.008}$	$0.627_{\pm 0.018}$ / $0.643_{\pm 0.008}$	$0.599_{\pm 0.024}$ / $0.640_{\pm 0.008}$
WORLDTRACE-FIELD	$0.591_{\pm 0.023}$ / $0.636_{\pm 0.008}$	$0.610_{\pm 0.021}$ / $0.640_{\pm 0.008}$	$0.588_{\pm 0.021}$ / $0.637_{\pm 0.008}$
WORLDTRACE-LANDMARK	$0.723_{\pm 0.024}$ / $0.543_{\pm 0.011}$	$0.709_{\pm 0.024}$ / $0.561_{\pm 0.011}$	$0.721_{\pm 0.022}$ / $0.557_{\pm 0.011}$
WORLDTRACE	$\mathbf{0.798}_{\pm 0.020}$ / $\mathbf{0.512}_{\pm 0.012}$	$\mathbf{0.767}_{\pm 0.023}$ / $\mathbf{0.548}_{\pm 0.011}$	$\mathbf{0.805}_{\pm 0.018}$ / $\mathbf{0.489}_{\pm 0.015}$
Tier 2, rollout length	ABA short (N=8)	ABA (N=16)	ABA long (N=32)
Sliding window	$0.805_{\pm 0.011}$ / $0.525_{\pm 0.008}$	$0.577_{\pm 0.026}$ / $0.643_{\pm 0.008}$	$0.422_{\pm 0.025}$ / $0.700_{\pm 0.005}$
WORLDTRACE-FIELD	$0.785_{\pm 0.013}$ / $0.530_{\pm 0.008}$	$0.591_{\pm 0.023}$ / $0.636_{\pm 0.008}$	$0.409_{\pm 0.025}$ / $0.703_{\pm 0.005}$
WORLDTRACE-LANDMARK	$0.865_{\pm 0.010}$ / $0.466_{\pm 0.010}$	$0.723_{\pm 0.024}$ / $0.543_{\pm 0.011}$	$0.632_{\pm 0.025}$ / $0.588_{\pm 0.011}$
WORLDTRACE	$\mathbf{0.891}_{\pm 0.007}$ / $\mathbf{0.460}_{\pm 0.011}$	$\mathbf{0.798}_{\pm 0.020}$ / $\mathbf{0.512}_{\pm 0.012}$	$\mathbf{0.750}_{\pm 0.024}$ / $\mathbf{0.537}_{\pm 0.013}$
Tier 3, camera orientation	Pan 90° (N=4)	Pan 180° (N=8)	Pan 360° (N=8)
Sliding window	$0.838_{\pm 0.014}$ / $0.561_{\pm 0.006}$	$0.560_{\pm 0.024}$ / $0.634_{\pm 0.008}$	$0.406_{\pm 0.025}$ / $0.740_{\pm 0.011}$
WORLDTRACE-FIELD	$0.818_{\pm 0.015}$ / $0.576_{\pm 0.007}$	$0.555_{\pm 0.023}$ / $0.650_{\pm 0.008}$	$0.378_{\pm 0.024}$ / $0.746_{\pm 0.012}$
WORLDTRACE-LANDMARK	$\mathbf{0.917}_{\pm 0.004}$ / $0.495_{\pm 0.005}$	$0.731_{\pm 0.021}$ / $\mathbf{0.608}_{\pm 0.008}$	$0.400_{\pm 0.025}$ / $0.729_{\pm 0.011}$
WORLDTRACE	$0.911_{\pm 0.008}$ / $\mathbf{0.453}_{\pm 0.005}$	$\mathbf{0.784}_{\pm 0.013}$ / $0.649_{\pm 0.008}$	$\mathbf{0.423}_{\pm 0.023}$ / $\mathbf{0.722}_{\pm 0.011}$
Tier 4, multi-revisit	ABABA (N=32)	ABCBA (N=20)	ABCDBA (N=20)
Sliding window	$0.462_{\pm 0.024}$ / $0.692_{\pm 0.006}$	$0.571_{\pm 0.018}$ / $0.659_{\pm 0.006}$	$0.595_{\pm 0.020}$ / $0.653_{\pm 0.006}$
WORLDTRACE-FIELD	$0.438_{\pm 0.025}$ / $0.696_{\pm 0.006}$	$0.573_{\pm 0.021}$ / $0.657_{\pm 0.007}$	$0.578_{\pm 0.020}$ / $0.649_{\pm 0.007}$
WORLDTRACE-LANDMARK	$0.719_{\pm 0.024}$ / $0.555_{\pm 0.012}$	$0.709_{\pm 0.023}$ / $0.551_{\pm 0.011}$	$0.735_{\pm 0.022}$ / $0.547_{\pm 0.010}$
WORLDTRACE	$\mathbf{0.791}_{\pm 0.019}$ / $\mathbf{0.518}_{\pm 0.012}$	$\mathbf{0.807}_{\pm 0.018}$ / $\mathbf{0.515}_{\pm 0.011}$	$\mathbf{0.808}_{\pm 0.020}$ / $\mathbf{0.503}_{\pm 0.011}$

Table 7: **Episodic recall as the horizon grows.** ABA loops, DINOv2 (↑) / LPIPS (↓), N_s=4, N_r=2.

Method	N=16	N=32	N=48	N=64	N=128
Sliding window	0.577 / 0.643	0.422 / 0.700	0.301 / 0.730	0.286 / 0.734	0.251 / 0.751
WORLDTRACE	**0.798 / 0.512**	**0.750 / 0.537**	**0.714 / 0.560**	**0.745 / 0.534**	**0.698 / 0.572**

waypoint structure as Tab. 1. The cache holds N_s=2 summary slots and N_r=16 recent frames (18 cached frames plus the current three-frame chunk form the trained window of 21 latent frames), every memory is applied per layer from that layer's own keys and values, and MemRoPE uses its released layout fitted to the two slots (one sink frame and the long EMA frame). Three trials per cell differ in the noise seed. WORLDTRACE leads on ABA, ABCDA, Pan 180° and the three multi-revisit loops, by 0.2 to 0.45 DINOv2 over the sliding window on ABA, ABCBA and ABCDBA, which repeats the MG2 result on a backbone with explicit camera conditioning. MemRoPE leads on ABCA, ABA long and Pan 360°, and is marginally ahead in DINOv2 on ABA short.

F.3 Attention Mismatch of Compressed Slots

We measure the Attention Mismatch of App. E inside the rollout. At every attention call, the same queries are scored both against a summary slot and against the M frames it stands for, kept verbatim at the slot's position, so that only content differs. The slot's weight is mapped back onto its M frames, and Δ_{attn} is half the L_1 distance between the two per-frame distributions, averaged over 96 queries per denoising pass, all heads, and all layers. We compare WORLDTRACE, whose averaged slot covers the newest temporal group, with a full-history average that puts all n_{old} old frames into a single slot. In Tab. 9, the mismatch grows with M and roughly doubles on the return leg, and the reference places far more attention on the summarized frames than the slot receives (.41 against .04 at N=32). This local comparison does not measure the loss of omitted frames or overall memory fidelity.

G The LOOPBENCH Memory Benchmark

G.1 Task Design and Evaluation

We propose **LOOPBENCH**, a benchmark for episodic memory in autoregressive world models. Each task is a navigation path that departs from a location, visits other locations, and returns. The frames generated on the first visit serve as the reference, and we measure how closely the frames generated on the return reproduce them. LOOPBENCH therefore does not need external reference video, as the rollout itself produces both the target and the prediction.

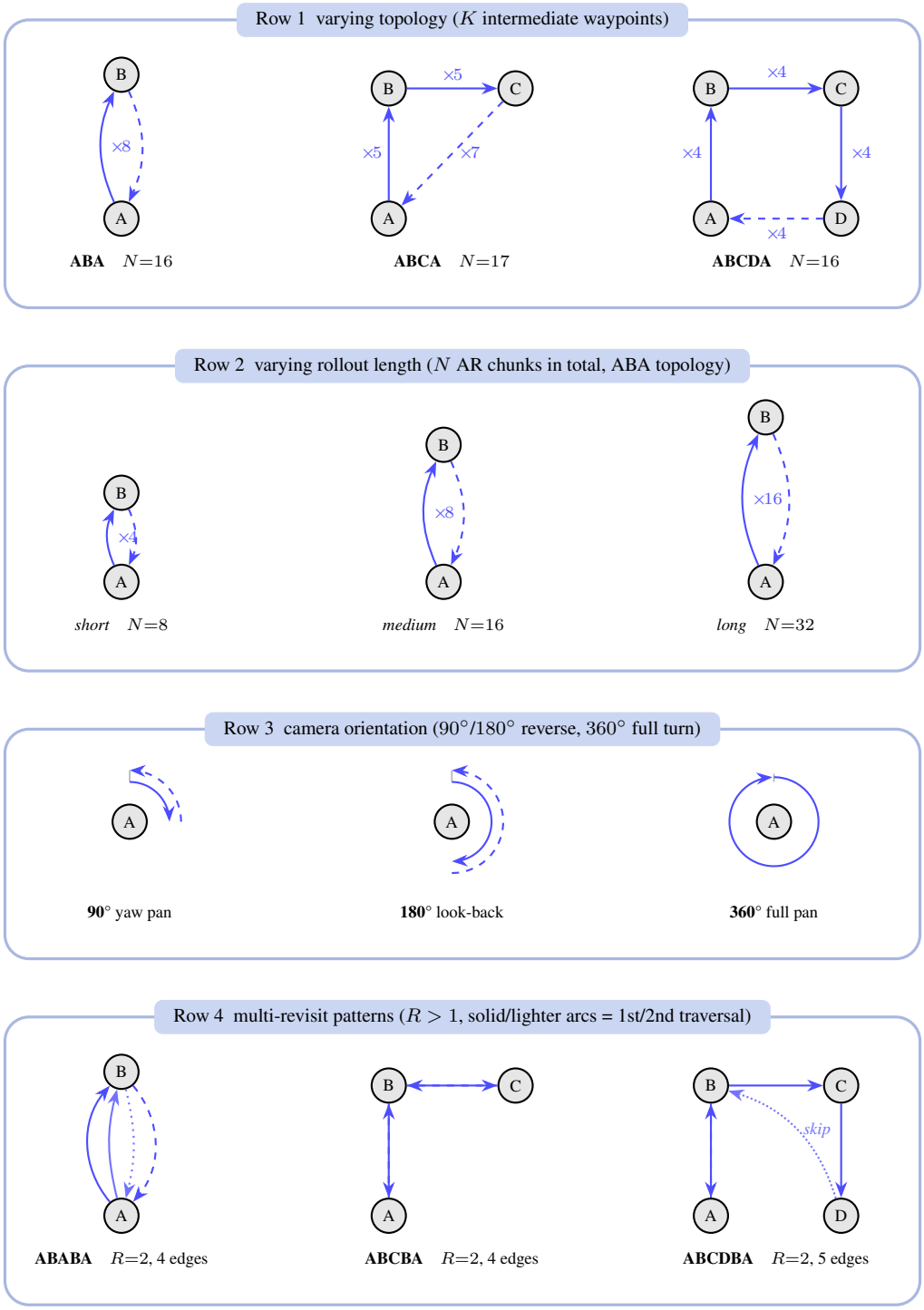


Figure 7: **LoopBench benchmark gallery.** Solid and dashed blue arrows mark the outbound and return legs, lighter arcs mark a second traversal, and dotted arrows mark shortcuts that skip intermediate waypoints. **Row 1** shows topology (ABA, ABCA, ABCDA). **Row 2** varies rollout length on the ABA topology, increasing the interval between departure and final return. **Row 3** varies camera orientation at A. The 90° and 180° pans reverse to the initial orientation, whereas the 360° condition completes one revolution. **Row 4** shows multi-revisit patterns, where R counts returns to A in ABABA, and visits to B in the ABCBA palindrome and the ABCDBA shortcut. These pattern-specific counts are all two.

Table 8: **LOOPBENCH on LingBot-World [4].** DINOv2 (↑) / LPIPS (↓) between the frames generated at the revisited location (e.g., A in ABA) on the first visit and on the return, mean over three trials, same four tiers as Tab. 1 with camera loops built on the release's example track. Every memory is applied per layer. MemRoPE is the released layout fitted to the two slots (one sink frame and the long EMA frame).

Tier 1, topology	ABA (K=1, N=16)	ABCA (K=2, N=18)	ABCDA (K=3, N=16)
Sliding window	0.586 / 0.670	0.343 / 0.694	0.409 / 0.693
MemRoPE	0.468 / 0.666	**0.713** / 0.670	0.545 / 0.672
WORLDTRACE	**0.812 / 0.656**	0.449 / **0.661**	**0.597 / 0.663**
Tier 2, rollout length	ABA short (N=8)	ABA (N=16)	ABA long (N=32)
Sliding window	0.864 / 0.657	0.586 / 0.670	0.319 / 0.710
MemRoPE	**0.873** / 0.656	0.468 / 0.666	**0.467 / 0.675**
WORLDTRACE	0.870 / **0.652**	**0.812 / 0.656**	0.274 / 0.686
Tier 3, camera orientation	Pan 90° (N=4)	Pan 180° (N=8)	Pan 360° (N=8)
Sliding window	0.628 / 0.638	0.408 / 0.666	0.457 / 0.690
MemRoPE	0.628 / 0.638	0.392 / 0.663	**0.522 / 0.670**
WORLDTRACE	0.628 / 0.638	**0.540 / 0.653**	0.460 / 0.680
Tier 4, multi-revisit	ABABA (N=32)	ABCBA (N=20)	ABCDBA (N=20)
Sliding window	0.399 / 0.701	0.302 / 0.693	0.271 / 0.706
MemRoPE	0.404 / 0.696	0.587 / **0.674**	0.682 / 0.687
WORLDTRACE	**0.430 / 0.675**	**0.748** / 0.680	**0.722 / 0.679**

Table 9: **Attention mismatch of compressed summary slots.** Δ_{attn} (↓), forward / return leg, on ABA loops for writers that replace M verbatim frames with one slot (N_s=4, N_r=2), measured against attending to those M frames verbatim at the slot's position. Each writer is measured on its newest summary slot, and full-history average puts all n_{old} old frames in one slot. Attention share (reference / slot) is at N=32, return leg, and DINOv2 (↑) is for the same single-seed rollouts.

		Δ_{attn} (↓), forward / return			Attention share	DINOv2 (↑)
Writer	M (N=8/16/32)	N=8	N=16	N=32	reference / slot	N=8/16/32
WORLDTRACE	7 / 13 / 25	.08 / .18	.14 / .27	.21 / .37	.41 / .04	.80 / .56 / .42
Full-history average	19 / 43 / 91	.19 / .36	.28 / .45	.37 / .54	.57 / .03	.81 / .58 / .44

G.2 Difficulty Axes

Four properties of the path control task difficulty. (1) **Waypoint count** (K) measures the number of intermediate locations between departure and return to A (K=1 for ABA and K=3 for ABCDA). (2) **Rollout length** (N) measures the total number of AR chunks in the path. Longer rollouts put more chunks between the first visit and the return, so the revisited scene sits deeper in the compressed cache. (3) **Camera orientation** varies while the agent stays at A. The 90° and 180° conditions pan away and then reverse to the initial orientation. The 360° condition makes one complete revolution. (4) **Multi-revisit patterns** include ABABA, which returns to A twice, while ABCBA and ABCDBA revisit B before their final return to A. The pattern-specific count R in the gallery is not a common depth measure across these paths. The reported metric compares the frames generated at A on the first visit and on the final return. These tiers are controlled families of tasks rather than independent single-factor interventions. Topology changes path geometry and can change duration, and the pan conditions also use different durations. The families are intended to distinguish memory behavior across navigation settings [111]. We show all twelve configurations in Fig. 7 and evaluate every one of them in Tab. 1. We show qualitative reconstructions on the ABA (axis 1) and pan 90° (axis 3) settings in Fig. 8 and Fig. 9.



Figure 8: **ABA qualitative results.** Each sample group of three frames shows trajectory keyframes at chunks 0 (A) / 7 (B) / 15 (A return). The two rows compare the sliding-window baseline (top) vs. WORLDTRACE (bottom). The sliding window drifts away from the scene-A appearance on the return leg across the four samples. WORLDTRACE brings the return frames back to the original scene appearance.

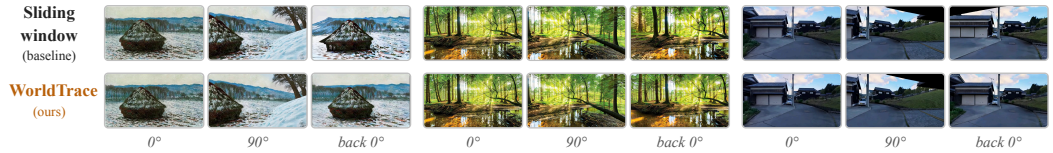


Figure 9: **Pan $90°$ qualitative results.** In camera-orientation Tier 3, the agent stays at A while the camera pans right by $\sim 90°$ and then pans back. Each sample group of three frames shows the start ($0°$), the pan peak ($\sim 90°$), and the return to $0°$. The two rows compare the sliding-window baseline (top) and WORLDTRACE (bottom). WORLDTRACE restores the initial-view appearance on the return. The sliding window does not.

H Implementation Details

H.1 Model Architecture and Self-Forcing Background

We evaluate on MG2-1.3B [5], a 1.3B-parameter autoregressive video world model built on Wan 1.3B T2V [21] with 30 layers, 12 attention heads, and head dimension 128. It generates video in chunks of 3 latent frames at 44×80 latent resolution (352×640 pixels), or 880 tokens per latent frame, and denoises each chunk with a distilled flow-matching sampler conditioned on the KV cache of prior chunks. Self-Forcing training [64] rolls the model out on its own KV cache rather than on teacher-forced context. Attention is restricted to a local window of L_{attn}=6 latent frames, so temporal offsets stay $\leq \Delta t_{\text{train}}$=5. WORLDTRACE keeps the cache at this fixed budget, about 0.97 GB in fp16 at batch size 1, independent of generation length.

H.2 WORLDTRACE Algorithm

After each new chunk is generated, its keys and values enter the verbatim recent window. When the recent window exceeds N_r chunks, the oldest chunk is evicted into the summary cache as a new canonical slot. If the summary cache then exceeds N_s slots, attention-response error on sampled queries selects which slots to merge or evict, and key-value similarity decides whether they are averaged into a field summary or kept as a landmark (Sec. 3.5). We list these update steps in Alg. 1, together with the final step that re-rotates each canonical summary key to its virtual position when attention scores are computed.

H.3 Hyperparameters, Compute, and Evaluation Protocol

Inference hyperparameters. MG2-1.3B [5] uses 3 distilled denoising steps ($t \in \{1000, 666, 333\}$ with timestep shift 5.0), no classifier-free guidance at inference, a single conditioning image, and F=3 latent frames per AR chunk. Each chunk is additionally conditioned on per-frame keyboard and mouse actions through the model's action module, which is how the camera trajectories are controlled. In our setting, the first chunk is generated with an empty KV cache, and every later chunk attends to the accumulated compressed cache. The MG2 LOOPBENCH and main ablation experiments use N_s=4 and N_r=2.

Table 10: **Runtime per chunk.** Wall-clock time at N=8 with VAE decode, on one L40 48 GB (batch 1, 3 distilled denoising steps, 352×640), mean over 51 rollouts.

Method	s/chunk
Sliding window	1.10
WORLDTRACE-FIELD	1.14
WORLDTRACE-LANDMARK	1.20
WORLDTRACE	1.40

Algorithm 1 WORLDTRACE cache update. $\mathcal{S}$ is one bounded bank of canonical slots with counts c. $\mathrm{Attn}(q;\cdot)$ is the attention output of query q over a cache, and $\mathcal{E}(u)$ is the attention-response error of update u, the change in the sampled queries' attention outputs plus a penalty for losing unique information. Each step applies the update that minimizes this error, so compression alters what the model retrieves as little as possible.

Require: recent cache $\mathcal{R}$, summary cache $\mathcal{S}$, new chunk's KVs, sampled queries $\mathcal{Q}$
Ensure: updated $(\mathcal{R}, \mathcal{S})$

$\mathcal{R} \leftarrow \mathcal{R} \cup \{\text{new KVs}\}$ ▷ recent window is verbatim
if $|\mathcal{R}| > N_r$ **then**
  Pop the oldest (K, V) at time t from $\mathcal{R}$
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{(R(-\theta_f t)\, K,\ V,\ c{=}1)\}$ ▷ canonical key
  if $|\mathcal{S}| > N_s$ **then**
    $u^\star \leftarrow \arg\min_u \mathcal{E}(u)$ over merges and evictions ▷ min. attention-response error
    if $u^\star$ merges compatible slots i, j **then**
      $(K_i, V_i, c_i) \leftarrow (\bar{K}, \bar{V}, \bar{c})$, remove slot j ▷ WORLDTRACE-FIELD
    else if $u^\star$ merges incompatible slots i, j **then**
      Retain a paired exemplar ▷ WORLDTRACE-LANDMARK
    else
      Remove the evicted slot
At attention time, $K_s \leftarrow R(\theta_f t_s^v)\ \mathcal{S}[s].K$ for $s{=}0, \ldots, N_s{-}1$ ▷ Def. 1

Compute. We report wall-clock time per AR chunk (including VAE decode) on one L40 48 GB at batch size 1 in Tab. 10. At the horizon $N{=}8$, WORLDTRACE-FIELD and WORLDTRACE-LANDMARK stay within 4% and 9% of the sliding-window baseline, since the forward pass dominates the cache update. WORLDTRACE adds 27% and the recompute writer revisits a growing history.

I Discussion

I.1 Limitations

WORLDTRACE operates entirely at inference time, leaving the pretrained weights frozen and changing only what information the fixed-size KV cache retains (compressed summaries in place of evicted frames) and the virtual RoPE positions at which those entries are read, with the goal of improving long-horizon memory. Everything else, including per-chunk visual quality, motion dynamics, and action following, is inherited unchanged from the pretrained backbone. The method also assumes an autoregressive generator with temporal RoPE applied to keys and a known local attention window. Compression into a fixed budget of N_s slots is lossy by construction, since each WORLDTRACE-FIELD slot averages roughly T/N_s frames, a ratio that grows with horizon, so frame-level detail inside a group is progressively blurred. Under WORLDTRACE-LANDMARK, a scene can be recalled later only if the scene-entry detector stored it as a landmark. Moreover, the cache holds only N_s landmarks, so a rollout that enters more scenes than that loses the oldest ones and can reliably revisit at most N_s distinct places. Future work could relax this cap, for example by merging evicted landmarks into a coarse residual summary slot so that discarded scenes leave a recoverable trace, or by selecting which landmark to evict based on predicted revisit likelihood.

I.2 Future Directions

Geometry-aware canonical keys. Coupling the unrotate/rerotate primitive with camera-pose warping (e.g., the Plücker and Warped-RoPE writers of MosaicMem [63] and UCM [62]) would align frames to a shared scene coordinate system and then average, so each summary token blends observations of similar scene content.

Learned scene-entry policies. A small policy trained on action discontinuities, agent-pose changes, or scene-segmentation logits could let WORLDTRACE-LANDMARK commit landmarks during continuous motion and decide *which* landmark to select. The fixed split between N_s and N_r could become an autotunable schedule and adapt per backbone or horizon without a manual sweep.

Fine-tuning extensions. WORLDTRACE stays within the attention window L_{attn}, where the budget $N_s + N_r = L_{\mathrm{attn}}$ keeps every summary slot in-distribution. Two light fine-tuning paths relax this budget. (i) *Context-extension fine-tuning* on synthetic long rollouts would let WORLDTRACE allocate

more summary slots for longer recall at the same in-distribution cost. (ii) *Position-aware fine-tuning* that trains the model on the slot-rank offsets of Eq. (2) would tighten the canonical-mean approximation of WORLDTRACE-FIELD.

Training memory policies from revisit supervision. The memory policy that decides what to store, compress, and evict could itself be trained, and revisit tasks supply the training signal for free, since whether the model reproduces the original scene on its return is directly measurable. Closed-loop simulators can collect such revisit episodes at scale, supporting supervised fine-tuning on rollouts where recall succeeds as well as reinforcement learning with the revisit score as reward. In both cases the backbone stays frozen and WORLDTRACE keeps every stored summary in-distribution, so only the lightweight policy over the cache is trained.

Scaling LOOPBENCH. We propose LOOPBENCH (App. G) with four difficulty tiers, and it can grow beyond them, since longer paths, more waypoints, and richer camera trajectories are all generated by the same recipe, and evaluating other world models on it would show how well their memory mechanisms support revisits, complementing WorldScore [107], MIND [108], and VBench-2.0 [109].

I.3 Broader Impact

WORLDTRACE is an inference-time cache mechanism that alters no training data, modifies no weights, and expands none of the generative capabilities of the underlying video model, so the relevant considerations are those already attached to pretrained video world models. By keeping long-horizon generation within a constant memory budget, it also lowers the compute cost of building on pretrained generators, a cost that shapes the wider ecosystem reusing them, from score-distilled 3D synthesis [112, 113] with compute-aware gradient estimators [114, 115] to LLM-conditioned mesh generation [116] and motion attribution for video models [117]. We release no new datasets, and downstream use should follow the base models' content policies.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper studies addressable memory and evaluates episodic recall on LOOP-BENCH (Sec. 4.2, App. G). The headline comparison uses DINOv2 and the tier/condition averaging convention defined in Sec. 4.2. The LPIPS exception and the distinction between attended-cache size and total memory are stated explicitly. The active evaluation does not establish a temporal-coherence improvement.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: App. I.1 discusses several limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: App. C proves mean pre-softmax score preservation at a shared virtual position. App. B gives a distribution-mismatch bound and conditional results for a relaxation with freely chosen prototype weights. These are not unconditional guarantees for inference-time attention. The phase-cancellation mechanism is described in Sec. 2.2.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Sec. 3 fully specifies the method, including slot indexing (Eq. (2)), canonical key averaging (Eq. (3)), and landmark selection and re-encoding (Eq. (4)). App. G describes how LOOPBENCH is constructed, App. E defines the recall metrics and preprocessing, and App. H lists the model settings, seeds, replication counts, and hardware.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will release the code and the LOOPBENCH evaluation suite once they complete NVIDIA's internal legal review. App. H and App. G describe the implementation and benchmark construction in enough detail to reproduce the main results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: No training is performed. Sec. 4, App. E, and App. H report the benchmarks, scoring protocol, and model-specific evaluation settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Tab. 6 reports means and standard errors for the sliding window and WorldTrace variants. The errors are sample standard deviations across 51 scene/seed rollouts divided by $\sqrt{51}$; they do not adjust for dependence between seeds from the same scene. The active

submission reports neither paired t-tests nor p-values, and these errors do not establish significance of small differences.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: MG2 evaluations use a single NVIDIA L40 48 GB GPU. Tab. 10 reports per-chunk time at N=8, including VAE decoding. LingBot uses a separate protocol, and its NTK-aware scaling baseline uses four L40 GPUs (Tab. 8). App. H reports approximately 600 L40 GPU-hours for the experiment set; runtime is not reported at multiple horizons.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research involves no human subjects, no scraped personal data, and no dual-use capabilities beyond those of the underlying pretrained video generation models. All experiments use synthetic game-engine environments (MatrixGame-2 and LingBot-World). We have reviewed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: App. I.3 discusses broader impact. WORLDTRACE is an inference-time cache mechanism that does not alter training data or model weights and does not expand generative capabilities beyond those of the base model.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This submission releases no new pretrained model weights or datasets. Matrix-Game 2.0 [5], on which MG2-1.3B is based, is released by its original authors with their own usage terms; we do not redistribute its weights. LingBot-World [4] is used under its stated research-use terms and is not redistributed. WORLDTRACE is an inference-time KV-cache modification that does not change the underlying generative capabilities of either base model, so no additional misuse risk is introduced beyond that of the existing pretrained models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: This submission releases no new pretrained model weights or datasets. Matrix-Game 2.0 [5] is released by its original authors under the MIT License; its Wan 2.1 backbone [21] is released under the Apache 2.0 License. The active recall metrics use DINOv2 [22] and LPIPS [23]. LingBot-World [4] is used under its stated research-use terms and is not redistributed. Evaluation use of all assets is permitted under the named licenses; no derivative model weights are released.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces LoopBench for controlled scene revisits (Sec. 4.2, App. G). It documents the trajectory families, rollout lengths, camera-orientation conditions, and multi-revisit patterns, together with DINOv2 and LPIPS scoring and a configuration gallery (Fig. 7). The benchmark uses earlier generated frames as references. No new pretrained models or datasets are released.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper involves no crowdsourcing and no research with human subjects. Recall evaluation is automated using DINOv2 feature similarity and LPIPS perceptual distance.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subjects are involved in this research. All experiments use synthetic game-engine environments with automated metrics.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not a component of the proposed methods or evaluation. The models studied (MG2, LingBot-World) are video diffusion transformers, not language models.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.