SRAM: SPATIALLY REGISTERED ADAPTIVE MEMORY FOR PERSISTENT VIDEO WORLD MODELS

Anonymous authors
Paper under double-blind review

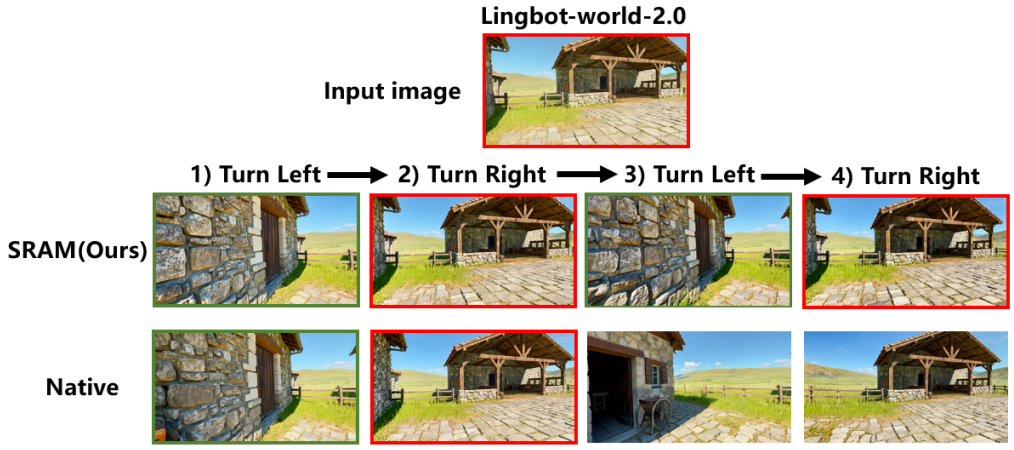

Figure 1: **Persistent video worlds with SRAM.** SRAM equips implicit historical memory with explicit spatial organization. Shared place records support pose-aware retrieval and correspondence-guided compression, retaining scene evidence across repeated revisits while filtering redundant observations.

ABSTRACT

Interactive video world models must preserve previously generated places as an agent repeatedly leaves and returns to them. When history is organized around individual observations, successive visits can accumulate redundant or inconsistent versions of the same place, increasing storage while weakening earlier scene evidence. We introduce Spatially Registered Adaptive Memory (SRAM), which organizes history into persistent place records and adapts memory selection to the spatial content already stored. Camera poses associate observations from different visits with a shared record, while a correspondence-guided compression module removes local redundancy and retains additional spatial evidence without overwriting accepted memory. To learn these correspondences, we propose a Spatial Correspondence Distillation (SCD) loss that aligns local descriptors of geometrically corresponding regions across views. This design combines global spatial organization with selective reuse of the generator's existing memory, without constructing an explicit 3D representation. We further introduce PersistBench, a benchmark that evaluates how well video world models recall previously generated scenes, how stable this memory remains across repeated revisits, and how efficiently it is stored. Experiments on MBench and PersistBench show improved scene retention and lower additional peak storage relative to the compared persistent-bank baselines. Further results is available on our anonymous project page.

1 INTRODUCTION

Interactive video world models (Bruce et al., 2024; He et al., 2025; Gao et al., 2026; Wu et al., 2026a) generate visual observations in response to camera or agent actions, allowing users to explore

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



Figure 2: Repeated revisits reveal scene-memory decay. In this rollout generated with Infinite-World, the first return largely preserves the initial scene, while later returns show increasing changes.

generated environments. Realistic transitions alone, however, do not ensure that an environment persists. When an agent returns to an earlier place, its geometry, objects, and appearance should remain consistent with the initial visit. This requirement extends beyond a single return: previously established places should remain recognizable after repeated departures, intervening exploration, and subsequent revisits.

Maintaining this information requires memory beyond the generator's recent context. Explicit memory (Zhao et al., 2026; Ren et al., 2025; Yu et al., 2024; 2025a) registers observations in spatial representations such as point clouds, meshes, or 3D feature structures. These representations provide a shared reference across views, but constructing and updating them introduces additional computation and storage. Implicit memory (Yu et al., 2025b; Yi et al., 2026) instead retains historical frames or internal representations that can be reused more directly by the generator. This offers an efficient way to preserve historical context, although the stored units often remain individual observations or temporal chunks, even when several units describe the same place.

While implicit memory makes historical context available, preserving a scene across repeated revisits also depends on how that context is organized and updated. A successful return may itself be stored as another observation of the same place. Subsequent visits can then draw on multiple versions of that scene, including regenerated observations with small errors. As these records accumulate, they consume storage without adding much spatial coverage and may weaken the influence of earlier evidence. We refer to the loss of consistency with previously established scene content as *memory decay*, and the incorporation of inconsistent generated observations into history as *memory pollution*. Figure 2 illustrates the loss of scene consistency in a rollout generated with Infinite-World (Wu et al., 2026a): the first return largely preserves the initial scene, while later returns exhibit increasing changes. Recovering a place once therefore does not establish whether its memory will remain reliable over subsequent visits.

These observations motivate maintaining a shared record for each place rather than treating every visit as independent history. We introduce *Spatially Registered Adaptive Memory* (SRAM), which combines a Global Spatial Memory Bank with Correspondence-Guided Memory Compression. Camera poses associate observations with persistent place records, allowing temporally distant visits to access the same historical evidence. Within each record, the compression module compares newly observed content with what is already stored. Memory selection adapts to this spatial overlap: repeated content is filtered, while additional spatial evidence can extend the record without overwriting accepted entries. Memory can thus grow with newly observed content rather than automatically expanding with every revisit.

To support this selection, we introduce a Spatial Correspondence Distillation (SCD) loss for learning local correspondence across views. Geometric correspondence supervision aligns descriptors of the same spatial region and distinguishes them from non-corresponding content. The learned relationships identify repeated scene evidence despite changes in image position or viewpoint, guiding the selection of the generator's existing historical memory without constructing an explicit 3D representation.

Evaluations centered on short-term recall or a single return provide limited insight into whether scene memory remains reliable across successive visits. We therefore introduce PersistBench, a benchmark for evaluating memory capability, stability under repeated revisits, and storage efficiency. Its cyclic and nested trajectories repeatedly leave and return to registered viewpoints, with each model's first generated visit serving as the reference for later returns. This allows the benchmark to distinguish how well a place is recalled from how that recall changes as revisits accumulate. Experiments on MBench and PersistBench compare memory mechanisms under a common video-generation backbone and show improved scene retention with lower additional peak storage relative to the compared persistent-bank baselines.

Our contributions are threefold:

i) We study repeated-revisit memory decay and pollution, highlighting how repeated observations can weaken established scene evidence even when a model successfully recovers a place on its first return.

ii) We introduce SRAM, which combines global spatial registration with correspondence-guided memory compression. The proposed SCD loss learns the local spatial relationships used to identify redundant history and adapt memory selection to previously stored evidence.

iii) We introduce PersistBench to evaluate how well video world models recall previously generated scenes, how stable their memory remains across repeated revisits, and how efficiently it is stored.

2 Related Work

Memory for video world models. Existing approaches maintain historical information through explicit spatial representations or implicit visual context (Sun et al., 2025; Yu et al., 2026; Li et al., 2025; Qian et al., 2026; Yang et al., 2025; Zhang & Agrawala, 2025). Spatia (Zhao et al., 2026) maintains a point-cloud memory and updates it through visual SLAM, providing a shared geometric reference for subsequent generation. Implicit approaches instead retain historical frames or internal representations (Xiao et al., 2026). Context-as-Memory (Yu et al., 2025b) selects historical frames using camera field-of-view overlap, while CaR (Peng et al., 2026) combines compressed context with relative-pose-conditioned retrieval. Compression further controls the amount of retained history: WorldKV (Yi et al., 2026) retrieves evicted historical chunks and compresses them using within-chunk similarity, whereas Infinite-World (Wu et al., 2026a) uses local and global hierarchical compression to represent long histories within a fixed budget. Beyond storage and retrieval, WorldTrace (Wu et al., 2026b) addresses the positional accessibility of compressed history through in-distribution virtual temporal positions. These approaches study different aspects of memory management, including spatial representation, context selection, compression, and positional organization.

Memory evaluation. Memory benchmarks evaluate (Huang et al., 2024; Zheng et al., 2025) whether generated content remains consistent after it leaves the current view. MBench (Zhang et al., 2026) organizes this evaluation into Entity, Environment, and Causal consistency. MIND (Ye et al., 2026) jointly measures memory consistency and action control through closed-loop revisits, distinguishing scene preservation from the execution of the requested motion. LoopBench (Wu et al., 2026b) evaluates recall of previously visited scenes after different detours and includes multiple-return trajectories. These protocols examine complementary aspects of world-model memory, from preservation of individual entities and scene attributes to recovery of a previously generated place after intervening exploration.

3 Method

3.1 Preliminaries

Interactive video world models. Let I_0 denote an initial observation and I_t the frame or video chunk generated at step t. The action a_t specifies the transition from step $t-1$ to t through agent controls or camera movement. Conditioned on an action sequence, an autoregressive video world model (Chen et al., 2024; Huang et al., 2026; Peebles & Xie, 2023) factorizes a rollout as

$$p_\psi(I_{1:T} \mid I_0, a_{1:T}) = \prod_{t=1}^{T} p_\psi(I_t \mid I_{0:t-1}, a_{1:t}), \tag{1}$$

where ψ denotes the generator parameters. The camera pose associated with I_t is $p_t = (R_t, \mathbf{c}_t)$, where R_t and $\mathbf{c}_t$ denote orientation and position in a shared world coordinate system.

Generation with historical memory. A finite-window generator conditions on recent observations $\mathcal{C}_t = I_{\max(0,t-w):t-1}$, where w is the context length (Henschel et al., 2025). Retrieved historical memory M_t^{ret} supplies additional context:

$$I_t \sim p_\psi(I_t \mid \mathcal{C}_t, a_t, M_t^{\text{ret}}). \tag{2}$$

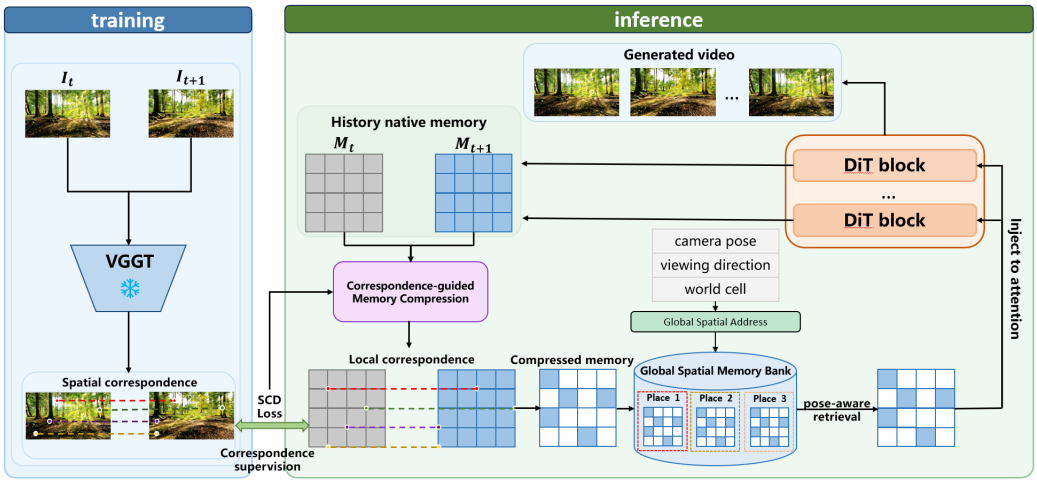

Figure 3: Overview of SRAM. The Global Spatial Memory Bank (Sec. 3.3) registers history to place records that support both retrieval and redundancy checking. Correspondence-Guided Memory Compression (Sec. 3.4) uses learned local correspondence to select additional historical evidence. Pose-aware retrieval reuses the stored memory for subsequent generation (Sec. 3.5).

We denote the native historical memory recorded from the generator for I_t by M_t, and its associated clean latent window by u_t. The spatial indices of u_t are aligned with the memory entries in M_t. Thus, M_t and M_{t+1} refer to memory associated with successive observations, rather than successive states of the entire memory bank.

3.2 Overview

SRAM equips the generator's implicit historical memory with explicit spatial organization for retrieval and compression (Fig. 3). The Global Spatial Memory Bank associates observations with persistent place records using camera pose. Each record serves both as a retrieval unit and as the reference for checking newly observed content. Correspondence-Guided Memory Compression compares local content against this reference and selects entries that provide additional spatial evidence.

Before each generation step, SRAM retrieves relevant history to condition the video model. After generation, the new memory M_t and its clean latent u_t enter the write path. Spatial registration identifies the destination record, while correspondence predictions from u_t determine which entries of M_t are retained. The selected memory is appended to that record for subsequent generation.

3.3 Global Spatial Memory Bank

We associate historical memory with a global spatial address derived from camera pose. Before generation step t, the bank $\mathcal{B}_t$ contains records $(\bar{p}_z, \mathcal{D}_z, \mathcal{M}_z)$ for previously observed places. Here, $\bar{p}_z$ is the representative camera pose of place z, $\mathcal{D}_z$ contains its accepted local descriptors, and $\mathcal{M}_z$ stores the corresponding historical memory. The pose provides the address; the descriptors support local matching within the record.

A deterministic router assigns each completed observation to a place:

$$z_t = \mathcal{R}(p_t; \mathcal{B}_t, \delta_x, \delta_R), \tag{3}$$

where δ_x and δ_R are translation and rotation thresholds. An existing record is eligible when both pose differences fall within these thresholds. Among eligible records, the router selects the one with the smallest normalized pose distance; otherwise, it creates a new record. A place therefore groups observations with compatible camera positions and viewing directions, while finer spatial correspondence is determined by the compression module.

The first visit establishes the record's reference memory. Later visits are routed back to this record and checked against its accepted descriptors before writing. Additional entries may be stored while capacity remains, but existing entries are not overwritten. Both the number of resident places and

the capacity of each record are bounded. Eviction removes an entire place record; accepted evidence is preserved for as long as that record remains stored. The configured capacities and record lifetime are described in Sections D.2 and A.2.

3.4 Correspondence-Guided Memory Compression

Observations assigned to the same place need not agree in image coordinates: camera movement can shift a surface within the image without revealing new scene content. We learn local correspondence to identify this spatial overlap and use it to filter redundant entries before writing.

Local correspondence prediction. A lightweight predictor maps the clean latent window u_t to local descriptors and matchability scores, $(D_t, m_t) = C_\theta(u_t)$. Here, $D_t \in \mathbb{R}^{N\times d}$ contains a d-dimensional descriptor for each of the N locations aligned with M_t, and $m_t \in [0,1]^N$ predicts cross-view matchability. The descriptors establish correspondences between observations. Matchability participates in screening candidate entries; it does not directly measure whether their content is new to the memory bank.

Spatial correspondence distillation. During training, a frozen VGGT (Wang et al., 2025a) teacher processes paired RGB observations (I_t, I_{t+1}). Its confidence-filtered geometric predictions provide correspondence labels on the latent grid. These labels supervise C_θ through Spatial Correspondence Distillation (SCD):

$$\mathcal{L}_{\mathrm{SCD}} = \mathcal{L}_{\mathrm{corr}} + \lambda_{\mathrm{align}}\mathcal{L}_{\mathrm{align}} + \lambda_{\mathrm{match}}\mathcal{L}_{\mathrm{match}}. \tag{4}$$

In this compact notation, $\mathcal{L}_{\mathrm{corr}}$ groups correspondence classification, hard-negative ranking, and unmatched rejection with their prescribed weights. The alignment term brings descriptors of corresponding spatial locations together, while the matchability term supervises whether a valid cross-view match exists. The objective is averaged over both matching directions. The bidirectional loss decomposition is given in Section B.2.

SCD transfers geometric correspondence rather than the teacher's feature representation. Only the correspondence predictor is optimized; the video generator and geometry teacher remain frozen.

Memory selection. For the addressed place z_t, current descriptors are compared with the accepted descriptors $\mathcal{D}_{z_t}$. A reliable match identifies content already represented in memory and excludes the corresponding entry from writing. Unmatched entries are considered for storage only after passing the matchability-based admission check. Lack of a stored match alone therefore does not trigger a write.

Let S_t denote the candidate indices admitted within the remaining memory budget. Compression and writing take the form

$$\widehat{M}_t = \mathrm{Select}(M_t, S_t), \qquad \mathcal{M}_{z_t} \leftarrow \mathcal{M}_{z_t} \oplus \widehat{M}_t, \tag{5}$$

where $\oplus$ appends selected entries without altering accepted memory. Their descriptors are added to $\mathcal{D}_{z_t}$ for subsequent matching. If S_t is empty, the record remains unchanged. The predictor controls which entries are retained from M_t rather than generating new memory values. Section A.3 summarizes the write sequence, and Section D.2 lists the memory configuration.

3.5 Pose-Aware Retrieval and Attention Injection

Before generating I_t, SRAM queries $\mathcal{B}_t$ with the target pose p_t to obtain $M_t^{\mathrm{ret}} = \mathrm{Retrieve}(\mathcal{B}_t, p_t)$. Spatially compatible records are ranked by pose distance, and a fixed retrieval budget selects the historical context used in Eq. 2. Retrieval uses the same place organization as writing, allowing observations from temporally distant visits to access the same retained evidence.

The retrieved entries retain their source temporal positions (Su et al., 2024), which are adjusted for reuse in the current generation context. This read-time adjustment is separate from the global spatial addressing used to select records. The adjusted history is combined with recent context through the video model's existing attention interface. After generation, Eq. 5 updates the addressed record, producing the bank available at the next step, $\mathcal{B}_{t+1}$.

4 PERSISTBENCH: BENCHMARKING PERSISTENT MEMORY

Persistent memory requires more than recovering a scene once: a world model should preserve previously established places across successive departures and returns. We introduce *PersistBench* to evaluate this capability through controlled revisit trajectories. PersistBench contains 24 scenes spanning game and synthetic environments as well as reconstructed real-world scenes (Straub et al., 2019; Ramakrishnan et al., 2021; Dosovitskiy et al., 2017; Lumberyard, 2017; Bitterli, 2016; Blender Foundation, n.d.). Its trajectories range from single-return routes to nested and repeated-return sequences, including `ABA`, `ABCBA`, `ABACA`, `ABABACA`, `ABCACBA`, and `ABACADA`. Each letter denotes a registered place, and repeated occurrences indicate returns to that place.

For each revisited place, the model's own first generated visit serves as a fixed reference for all subsequent returns. The central evaluation is *revisit-order retention*, which measures consistency with this reference across successive revisits. For each trajectory-duration setting, each curve includes all trajectories supporting its reported return horizon. This set remains fixed across revisit orders and compared methods, separating sustained scene recall from a single successful recovery.

We additionally report degradation across returns and worst-return retention to capture changes in consistency and failures obscured by average scores. Memory cost is measured as the increase in peak storage relative to the native backbone. Generation fidelity and action adherence provide complementary checks against high consistency caused by failed motion. The scene inventory, evaluation coverage, and metric definitions are given in Section C.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Benchmarks. We evaluate all cases in MBench (Zhang et al., 2026) and PersistBench. Each of the ten reported MBench dimensions uses all applicable benchmark cases and belongs to Entity, Environment, or Causal memory. PersistBench reports revisit-order retention, curve-level degradation, and additional peak storage relative to Native, using all trajectories that support each reported return horizon. Metric definitions and evaluation coverage are given in Sections C.2 and D.3.

Baselines. We use the same frozen LingBot-World 2.0 backbone for the controlled comparisons. Deep Forcing (Yi et al., 2025), MemRoPE (Kim et al., 2026), Infinity-RoPE (Yesiltepe et al., 2026), WorldKV (Yi et al., 2026), and Context-as-Memory (Yu et al., 2025b) are adapted to this backbone, alongside its native context-window baseline. All LingBot-based methods share the input conditions, camera trajectories, sampling configuration, and generation seed. Infinite-World (Wu et al., 2026a) and CaR (Peng et al., 2026) are additionally reported with their native Wan2.1-based (Wan et al., 2025) systems as system-level references, rather than controlled memory-policy comparisons. Baseline adaptations, method-specific hyperparameters, and memory budgets are detailed in the supplementary material.

Training details. All training and evaluation experiments are conducted on NVIDIA A800 GPUs with 80 GB of memory per GPU. We train the correspondence predictor on the video dataset released with Context-as-Memory (Yu et al., 2025b). Paired observations are encoded into clean video latents, and the corresponding RGB frames are used to precompute geometric correspondence labels with a frozen VGGT (Wang et al., 2025a) teacher. Only the predictor is optimized using the SCD objective in Sec. 3.4; the video generation backbone remains frozen. We use AdamW (Loshchilov & Hutter, 2017) with a learning rate of 2×10^{-4}, weight decay of 10^{-4}, and gradient-norm clipping at 1.0. Four independent runs are trained for 20K steps each, and the deployment checkpoint is selected by validation correspondence performance. The training inputs, loss decomposition, and optimization settings are summarized in Section B.

5.2 QUANTITATIVE RESULTS ON MBENCH

Table 1 reports the ten evaluated MBench dimensions. Under the common LingBot-World 2.0 backbone, SRAM obtains the highest Object Texture, Reprojection, Style, and Lighting scores, and ranks second in Epipolar Consistency and Prompt Interaction. The improvement over Native is particularly pronounced in Object Texture, which increases from 0.2322 to 0.5704, and Reprojection, which increases from 0.0256 to 0.0929. Full KV is included as an additional full-history reference.

Table 1: MBench results across ten evaluated dimensions. Best and second-best values within each group are shown in **bold** and *italics*, respectively.

Method	Entity ↑				Environment ↑				Causal ↑	
	HI	HA	OT	OG	Epi	Repr	Sty	Lit	Cam	Prm
LingBot-World 2.0 backbone										
Native	0.1971	0.4861	0.2322	0.0000	0.0344	0.0256	0.9295	0.9451	**0.4355**	0.3034
Full KV	0.2611	**0.5830**	0.3791	0.0000	0.0049	0.0177	0.9411	0.9297	0.3851	0.3014
Deep Forcing	**0.3265**	*0.5691*	0.4873	0.0000	0.0459	0.0645	0.9404	0.9405	0.4261	0.2979
MemRoPE	0.2275	0.4866	0.1546	0.0000	0.0416	0.0474	0.9374	0.9530	*0.4272*	0.3094
Infinity-RoPE	0.1741	0.4185	0.0000	0.0000	0.0041	0.0034	0.9236	0.9392	0.4000	**0.3184**
WorldKV	0.2500	0.5189	0.2992	**0.0318**	0.0322	0.0458	0.9516	0.9531	0.4172	0.3068
Context-as-Memory	*0.3094*	0.5568	*0.5559*	0.0000	**0.0970**	*0.0710*	*0.9580*	*0.9612*	0.4260	0.3101
SRAM (Ours)	0.2997	0.5548	**0.5704**	0.0000	*0.0798*	**0.0929**	**0.9656**	**0.9620**	0.4271	*0.3111*
Wan2.1-based systems										
Infinite-World	*0.3460*	**0.5704**	*0.4753*	0.0000	*0.0921*	**0.0991**	*0.9550*	*0.9402*	**0.8608**	0.0000
CaR	**0.3501**	*0.5448*	**0.5016**	0.0000	**0.0929**	*0.0907*	**0.9781**	**0.9781**	*0.4078*	**0.2884**

Table 2: PersistBench results. ΔP is peak storage relative to Native (Ref.). Best and second-best values are shown in **bold** and *italics*, respectively.

Method	Retention ↑			Degradation ↓	Memory Efficiency ↓
	1st	2nd	3rd	$\mathrm{MRD}_{\mathrm{curve}}$	ΔP (GiB)
Native	0.1665	0.1089	0.1112	0.0576	Ref.
Deep Forcing	0.0983	0.1088	0.0601	0.0487	218.96
MemRoPE	0.1999	0.1831	*0.2362*	**0.0168**	144.24
Infinity-RoPE	0.1878	0.1376	0.1036	0.0842	*133.04*
WorldKV	0.2301	0.1393	0.1195	0.1106	338.78
Context-as-Memory	*0.2384*	*0.2174*	0.2214	0.0210	242.27
SRAM (Ours)	**0.2605**	**0.2547**	**0.2428**	*0.0177*	**97.12**

The strongest method varies across dimensions. Deep Forcing leads Human Identity, Full KV leads Human Appearance, and Context-as-Memory leads Epipolar Consistency. Native and Infinity-RoPE have the highest Camera and Prompt Interaction scores, respectively. SRAM's gains are therefore concentrated in texture preservation and several environment dimensions, rather than a uniform improvement across all measures. Infinite-World and CaR are reported separately as system-level references. Additional causal and geometry evaluations are provided in the supplementary material.

5.3 Quantitative Results on PersistBench

Table 2 compares persistent memory under the common LingBot-World 2.0 backbone. SRAM achieves the highest retention at all three revisits. Its advantage over WorldKV becomes more pronounced after additional returns: their first-revisit scores are 0.2605 and 0.2301, whereas their third-revisit scores are 0.2428 and 0.1195. Context-as-Memory is the closest comparison on the first two returns, but remains below SRAM throughout. These results show that differences which are relatively small on the first return can become substantial after further exploration and memory updates.

SRAM also achieves the second-lowest $\mathrm{MRD}_{\mathrm{curve}}$ at 0.0177, compared with 0.0576 for Native, 0.1106 for WorldKV, and 0.0210 for Context-as-Memory. MemRoPE obtains a slightly lower value of 0.0168, but has lower retention at every return. Its third-return score of 0.2362 follows a recovery from 0.1831 on the second return, whereas SRAM remains above 0.24 throughout. Taken together, the retention and degradation scores show that SRAM maintains a higher level of scene recall with limited decline over the evaluated revisits.

SRAM has the lowest reported additional peak storage among the memory methods: 97.12 GiB, compared with 242.27 GiB for Context-as-Memory and 338.78 GiB for WorldKV. These are reduc-

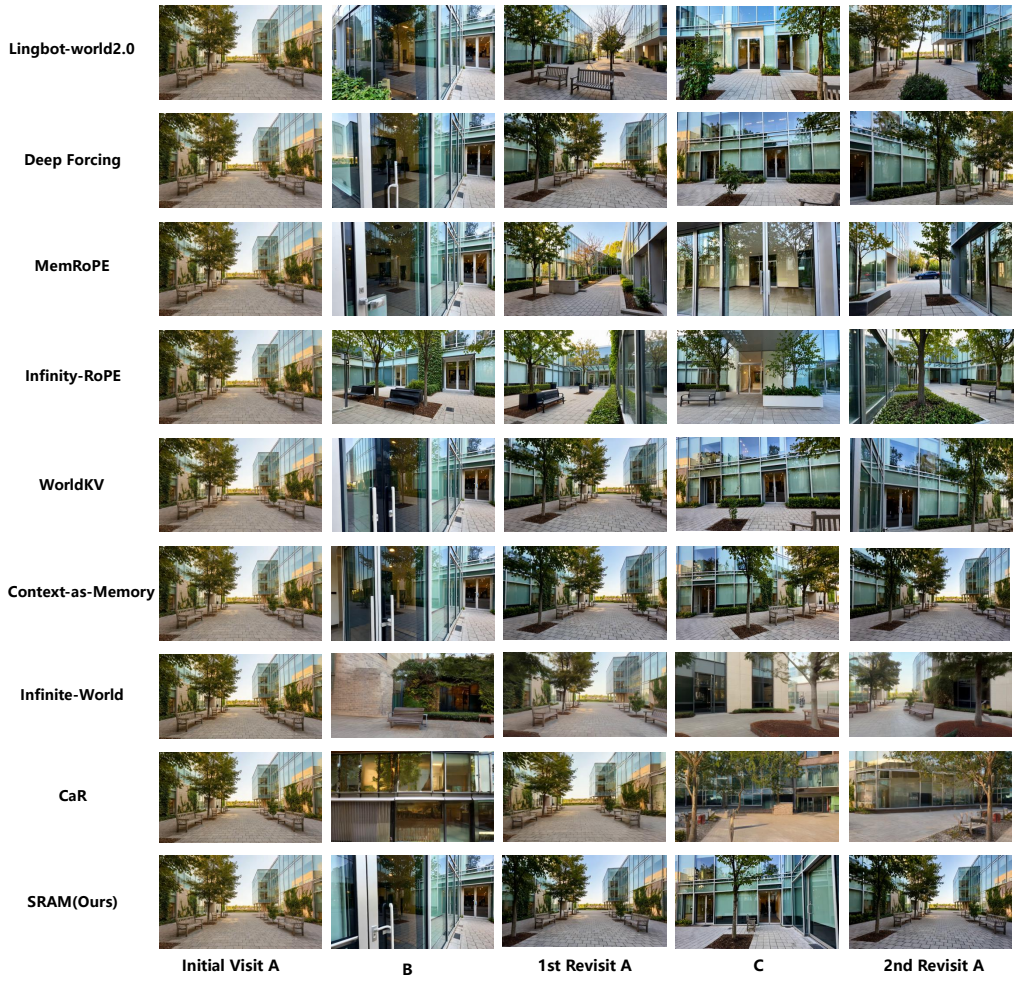

Figure 4: Qualitative comparison along an ABACA trajectory. The initial visit to A is compared with its first and second returns for each method.

tions of 59.9% and 71.3% in the peak increment over Native, rather than in total system memory. Under this accounting, SRAM combines higher revisit retention with lower memory overhead. The component ablations examine the effects of spatial organization and correspondence-guided compression.

5.4 Qualitative Analysis

Figure 4 compares repeated visits to A along an ABACA trajectory. The first return follows exploration of B, whereas the second follows further exploration of C. Each method is compared with its own initial observation, focusing on the preservation of scene appearance and local structure rather than differences between the scenes initially generated by different models. This view complements the quantitative analysis by showing which parts of an established scene are retained or changed after intervening generation. Additional qualitative results and complete generated videos are provided in the supplementary material.

5.5 Ablation Study

We evaluate two module-removal variants: one removes the Global Spatial Memory Bank but retains correspondence-guided compression; the other keeps spatial storage and retrieval but disables compression. Native provides the local-context reference. All ablation variants use the same backbone and the complete set of long-horizon trajectories supporting three returns, as specified in Section C.2. The evaluation cases and metric definitions match the main three-return PersistBench comparison.

Table 3: Ablation results on PersistBench. ΔP follows Table 2. Best and second-best values are shown in **bold** and *italics*, respectively.

Configuration	Retention ↑			Degradation ↓	Memory Efficiency ↓
	1st	2nd	3rd	$\mathrm{MRD}_{\mathrm{curve}}$	ΔP (GiB)
Native	0.1665	0.1089	0.1112	0.0576	Ref.
w/o Spatial Bank	0.2001	0.1773	0.1359	0.0642	*401.56*
w/o Memory Compression	*0.2593*	*0.2434*	*0.2300*	*0.0293*	402.04
SRAM (Ours)	**0.2605**	**0.2547**	**0.2428**	**0.0177**	**97.12**

Table 3 shows that removing the Spatial Bank reduces retention at every revisit. The gap to SRAM grows from 0.0604 on the first return to 0.1069 on the third, where retention falls from 0.2428 to 0.1359. $\mathrm{MRD}_{\mathrm{curve}}$ increases from 0.0177 to 0.0642, and additional peak storage rises from 97.12 to 401.56 GiB. Correspondence-guided compression alone does not reproduce the full method's recall or memory-cost results. The widening gap across returns supports the role of place-level organization in maintaining historical evidence through repeated memory use.

Removing Memory Compression has a smaller effect on retention but a substantial effect on additional peak storage. The first-return score remains close to SRAM, at 0.2593 versus 0.2605, while the second- and third-return scores decrease to 0.2434 and 0.2300. The corresponding $\mathrm{MRD}_{\mathrm{curve}}$ rises to 0.0293. Additional peak storage increases to 402.04 GiB, so the full method reduces this increment by 75.8% while retaining higher scores at all three returns. Disabling compression therefore increases measured memory overhead without improving recall in this comparison. Spatial organization provides the shared place record, while correspondence-guided selection reduces the measured memory cost without lowering retention.

6 Limitations

SRAM relies on camera poses in a shared coordinate system. Pose errors or unsuitable spatial thresholds can assign observations of the same place to different records or group unrelated views together. Correspondence errors may also discard useful observations or retain redundant content, further corrupting the spatial memory and degrading the consistency of scene representations. The memory bank has bounded capacity: accepted evidence remains available only while its place record is retained, and eviction removes that reference entirely from active memory. Compression reduces redundant storage but does not guarantee access to every previously visited place over arbitrarily long exploration; distant or rarely revisited locations may be pruned even if they could support future localization and scene reasoning.

PersistBench focuses on static environments and camera revisits. It does not address dynamic scene states. Such changes require distinguishing legitimate state updates from inconsistent regenerated observations, rather than treating earlier evidence as a fixed reference. This introduces additional challenges for memory management, as outdated records must be selectively invalidated without erasing stable background geometry. The controlled comparisons also use LingBot-World 2.0 as the common backbone. Robustness to estimated poses, dynamic environments, and other video-generation backbones remains to be evaluated, alongside long-term memory stability under noisy pose streams and frequent scene modifications.

7 Conclusion

We presented Spatially Registered Adaptive Memory (SRAM), which equips implicit historical memory with explicit spatial organization for retrieval and compression. The Global Spatial Memory Bank associates repeated visits with shared place records, while Correspondence-Guided Memory Compression uses spatial relationships learned through the SCD objective to identify redundant evidence. The method reuses the generator's native memory without maintaining an explicit 3D scene representation. We also introduced PersistBench to evaluate scene recall, memory stability across repeated revisits, and storage efficiency. Under the evaluated configurations, experiments on MBench and PersistBench show improved scene retention and reduced memory overhead relative to the compared memory-bank baselines. These results support maintaining shared spatial records for both retrieving established evidence and deciding which new observations to retain.

AI USE STATEMENT

Generative AI tools assisted literature research, manuscript drafting and polishing, figure preparation, and discussions of the method, evaluation design, and results. The authors reviewed and revised the resulting drafts and take responsibility for the final content of this work.

ETHICS STATEMENT

PersistBench is intended for research evaluation. The use and redistribution of its source datasets and scene assets are subject to their original licenses and applicable usage restrictions.

REPRODUCIBILITY STATEMENT

Sections 3.3–3.5 describe the memory mechanism, and Section 5.1 summarizes the experimental setup. Appendix A collects the notation and read–generate–write sequence. Appendix B describes the training inputs, loss decomposition, and optimization settings. Appendix C specifies the scene and trajectory inventories, full evaluation coverage, frame pairing, metric aggregation, and peak-storage comparison. Appendix D records the baseline adaptations and additional results, followed by visual examples in Appendices E and F.

REFERENCES

Benedikt Bitterli. Rendering resources, 2016. https://benedikt-bitterli.me/resources/.

Blender Foundation. Blender demo files. `https://www.blender.org/download/demo-files/`, n.d. Accessed: 2026-09-26.

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

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pp. 1–16. PMLR, 2017.

Zelin Gao, Qiuyu Wang, Jiapeng Zhu, Jingye Chen, Zichen Liu, Qingyan Bai, Jiahao Wang, Yufeng Yuan, Hanlin Wang, Yichong Lu, et al. Infinite worlds with versatile interactions. *arXiv preprint arXiv:2607.07534*, 2026.

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

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2568–2577. IEEE, 2025.

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

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024.

Youngrae Kim, Qixin Hu, C-C Jay Kuo, and Peter Beerel. Memrope: Training-free infinite video generation via evolving memory tokens. In *European Conference on Computer Vision*, pp. 244–261. Springer, 2026.

Runjia Li, Philip Torr, Andrea Vedaldi, and Tomas Jakab. Vmem: Consistent interactive video scene generation with surfel-indexed view memory. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25690–25699. IEEE, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

Amazon Lumberyard. Amazon lumberyard bistro, open research content archive (orca), July 2017. URL `http://developer.nvidia.com/orca/amazon-lumberyard-bistro`.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Zhan Peng, Jie Ma, Huiqiang Sun, Chong Gao, Zhijie Xue, Zhiyu Pan, Zhiguo Cao, Jun Liang, and Jing Li. Compression and retrieval: Implicit memory retrieval for video world models. *arXiv preprint arXiv:2606.23105*, 2026.

Runjia Qian, Zile Wang, Jihai Zhang, Kai Zou, Wei Yu, Jiaxing Li, Zexiang Liu, Yaokun Li, Fei Kang, Kaichen Huang, et al. Matrix-game 3.5: Enhancing real-time streaming interactive world models with patch memory. *arXiv preprint arXiv:2608.29910*, 2026.

Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6121–6132. IEEE, 2025.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

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

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

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025a.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10510–10522. IEEE, 2025b.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709. IEEE, 2024.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Ruiqi Wu, Xuanhua He, Meng Cheng, Tianyu Yang, Yong Zhang, Zhuoliang Kang, Xunliang Cai, Xiaoming Wei, Chunle Guo, Chongyi Li, et al. Infinite-world: Scaling interactive world models to 1000-frame horizons via pose-free hierarchical memory. *arXiv preprint arXiv:2602.02393*, 2026a.

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

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

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

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

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.

Hidir Yesiltepe, Tuna Meral, Adil Kaan Akan, Kaan Oktay, and Pinar Yanardag. Infinity-rope: Action-controllable infinite video generation emerges from autoregressive self-rollout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40256–40265, 2026.

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

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

Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T Freeman, and Jiajun Wu. Wonderworld: Interactive 3d scene generation from a single image. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5916–5926. IEEE, 2025a.

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. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025b.

Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024.

Wei Yu, Runjia Qian, Yumeng Li, Liquan Wang, Songheng Yin, Dennis Anthony, Yang Ye, Yidi Li, Weiwei Wan, Animesh Garg, et al. Mosaicmem: Hybrid spatial memory for controllable video world models. *arXiv preprint arXiv:2603.17117*, 2026.

Lvmin Zhang and Maneesh Agrawala. Packing input frame context in next-frame prediction models for video generation. *arXiv e-prints*, pp. arXiv–2504, 2025.

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

Shengjun Zhang, Zhang Zhang, Simin Huang, Zhenyu Tang, Hanyang Wang, Chensheng Dai, Min Chen, Yifan Li, Yuxin Li, Yingjie Chen, et al. Mbench: A comprehensive benchmark on memory capability for video world models. *arXiv preprint arXiv:2606.00793*, 2026.

Jinjing Zhao, Fangyun Wei, Zhening Liu, Hongyang Zhang, Chang Xu, and Yan Lu. Spatia: Video generation with updatable spatial memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4245–4257, 2026.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Lulu Gu, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, et al. Vbench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

SUPPLEMENTARY MATERIAL

This supplement provides additional method details, training settings, and experimental results. Sections A and B describe the memory mechanism and correspondence supervision. Section C presents the construction and evaluation protocol of PersistBench. Section D provides experimental settings, baseline adaptations, and detailed results on MBench and PersistBench. Section E presents visual comparisons with baseline methods, and Section F provides further results across scenes and trajectories.

CONTENTS

A. Additional Method Details

B. Training and Spatial Correspondence Distillation

C. PersistBench: Construction and Evaluation Protocol

D. Additional Experimental Details

E. Qualitative Comparisons

F. Additional Visual Results

A ADDITIONAL METHOD DETAILS

A.1 NOTATION AND MEMORY REPRESENTATION

Table 4 collects the notation introduced in Section 3. Observation memory M_t is recorded after I_t is generated, whereas $\mathcal{B}_t$ is the bank available before that step.

Table 4: Notation used by SRAM.

Symbol	Meaning
I_t, a_t	Generated observation at step t and the action leading to it.
$p_t = (R_t, \mathbf{c}_t)$	Camera orientation and position in the shared world coordinate system.
$\mathcal{C}_t$	The generator's recent visual context.
M_t, u_t	Native historical memory for the completed observation and its aligned clean latent window.
$\mathcal{B}_t$	Persistent memory bank available before step t.
z_t, $\bar{p}_z$	Addressed place index and a record's representative camera pose.
$\mathcal{D}_z$, $\mathcal{M}_z$	Accepted local descriptors and the corresponding stored memory in place z.
C_θ, (D_t, m_t)	Correspondence predictor and its descriptor and matchability outputs.
S_t, $\widehat{M}_t$	Accepted candidate indices and the selected historical memory.
M_t^{ret}	Historical memory retrieved to condition the current generation step.

The predictor reads u_t and returns descriptors aligned with entries in M_t. Selecting descriptor indices therefore selects the corresponding native memory entries; the descriptors do not replace their contents.

A.2 SPATIAL REGISTRATION AND RECORD LIFETIME

Registration follows Eq. 3. The representative pose is a record address, not a reconstructed surface coordinate: observations of the same surface from sufficiently different camera poses may belong to different records (Wang et al., 2025b). Local correspondence is evaluated within the addressed record. The pose thresholds and memory capacities used in the experiments are listed in Section D.2.

The first visit initializes a record. Later visits can append selected entries while capacity remains, but do not overwrite accepted entries. Eviction removes the record and its reference evidence together. The non-overwriting rule therefore applies to resident records, rather than guaranteeing unlimited retention of every visited place.

A.3 Read–Generate–Write Sequence

Table 5 summarizes the execution order. Retrieval uses records available before generation. The newly generated observation is processed by the write path only after it has been completed.

Table 5: Execution sequence of SRAM at generation step t.

Step	Operation
Read	Retrieve M_t^{ret} from $\mathcal{B}_t$ using the target pose p_t.
Generate	Condition the frozen generator on recent context, the action, and retrieved history; obtain I_t, M_t, and u_t.
Address	Associate the completed observation with place z_t, allocating a new record when needed.
Compare	Apply C_θ to u_t and compare its local descriptors with $\mathcal{D}_{z_t}$.
Select	Reject redundant entries and apply the admission and remaining-capacity checks to obtain S_t.
Write	Append $\mathrm{Select}(M_t, S_t)$ and its descriptors to the addressed record, yielding $\mathcal{B}_{t+1}$.

A failed match does not by itself admit an entry: the candidate must also pass the admission and remaining-capacity checks in Section 3.4. If no candidate is accepted, the record contents remain unchanged. At retrieval, the temporal-position adjustment in Section 3.5 is applied separately from place addressing.

B Training and Spatial Correspondence Distillation

B.1 Training Data and Pair Preparation

The correspondence predictor is trained on the Context-as-Memory video dataset used in Section 5.1. For each paired observation, the clean video latent is the predictor input and the RGB frames provide the input to the frozen VGGT teacher. Confidence-filtered geometric predictions supply correspondence labels aligned with the predictor's output grid (Wang et al., 2024). The generator and teacher remain frozen; only C_θ is optimized.

B.2 Bidirectional SCD Objective

Equation 4 groups correspondence supervision in $\mathcal{L}_{\mathrm{corr}}$. For a query–reference pair, let $\ell_{\mathrm{cls}}^{q\to r}$ denote correspondence classification, $\ell_{\mathrm{rank}}^{q\to r}$ hard-negative ranking, and $\ell_{\mathrm{reject}}^{q\to r}$ rejection of invalid matches. Then

$$\begin{aligned}
\ell_{\mathrm{corr}}^{q\to r} &= \alpha_{\mathrm{cls}}\ell_{\mathrm{cls}}^{q\to r} + \alpha_{\mathrm{rank}}\ell_{\mathrm{rank}}^{q\to r} + \alpha_{\mathrm{reject}}\ell_{\mathrm{reject}}^{q\to r},\\
\ell_{\mathrm{SCD}}^{q\to r} &= \ell_{\mathrm{corr}}^{q\to r} + \lambda_{\mathrm{align}}\ell_{\mathrm{align}}^{q\to r} + \lambda_{\mathrm{match}}\ell_{\mathrm{match}}^{q\to r},\\
\mathcal{L}_{\mathrm{SCD}} &= \tfrac{1}{2}\left(\ell_{\mathrm{SCD}}^{q\to r} + \ell_{\mathrm{SCD}}^{r\to q}\right).
\end{aligned} \tag{6}$$

The α coefficients collect the weights absorbed into $\mathcal{L}_{\mathrm{corr}}$ in Eq. 4. The directed matchability term combines the supervision for the paired observations. Classification and ranking distinguish valid correspondences from competing locations; rejection handles locations without a valid match. Alignment compares positive descriptors, and matchability predicts whether cross-view correspondence is available. The reverse-direction loss uses the same terms with query and reference exchanged.

B.3 Optimization and Checkpoint Selection

Table 6 collects the training settings from Section 5.1. Four runs are optimized independently; the reported deployment checkpoint is chosen by validation correspondence performance.

Table 6: Correspondence-predictor training configuration.

Setting	Value
Training source	Context-as-Memory video dataset
Optimized module	Correspondence predictor C_θ
Frozen modules	Video generator and VGGT teacher
Optimizer	AdamW
Learning rate	2×10^{-4}
Weight decay	10^{-4}
Gradient-norm clipping	1.0
Independent training runs	4
Optimization steps per run	20,000
Checkpoint selection	Validation correspondence performance
GPU model	NVIDIA A800, 80 GB per GPU

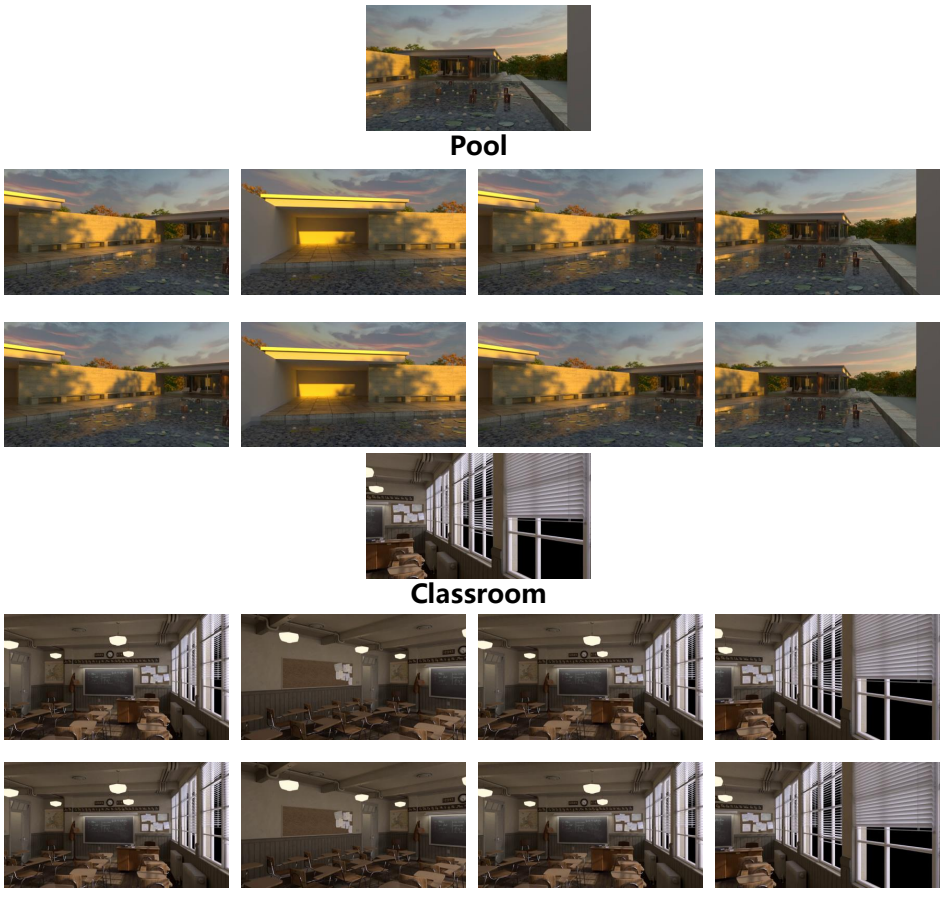

Figure 5: Scenes and corresponding video frames from PersistBench.

C PERSISTBENCH: CONSTRUCTION AND EVALUATION PROTOCOL

C.1 SCENES AND TRAJECTORIES

PersistBench contains 24 static scenes: 12 reconstructed real-world scenes from Replica and HM3D, and 12 game or synthetic scenes from CARLA, NVIDIA ORCA, Blender demonstration assets, and the Bitterli scene collection. Table 7 lists the sources and scene counts. Figure 5 shows scenes and frames from their corresponding videos.

Table 7: Scene sources in PersistBench.

Scene type	Source	Scenes
Reconstructed real-world	Replica	7
	HM3D	5
Game / synthetic	CARLA	1
	NVIDIA ORCA	4
	Blender demonstration scenes	3
	Bitterli scene collection	4
Total		24

Table 8: Trajectory patterns and reference-video lengths. Short routes are available for all 24 scenes; long routes use the 12 reconstructed scenes. "–" indicates that the route is not included in the long-horizon set.

Pattern	Returns to A	Short frames	Long frames
ABA	1	81	–
ABCBA	1	153	–
ABACA	2	153	1,845
ABABACA	3	225	2,745
ABCACBA	2	225	2,745
ABACADA	3	225	2,745

For each scene, we register camera viewpoints $A, B, C, \ldots$ and connect them with continuous camera trajectories. The camera remains stationary at each viewpoint for an observation interval. Repeated letters return to the same position and orientation. The initial visit supplies the reference frames; later occurrences are numbered as revisits. For example, ABACADA contains an initial visit to A and three revisits.

Reference videos. Each scene has six routes, listed in Table 8. The short reference release uses nine-frame observation intervals and 27-frame transitions. It contains 144 sequences and 25,488 frames at 832×464 resolution and 16 FPS (26.55 minutes). Of these sequences, 48 follow the single-return routes ABA and ABCBA; the remaining 96 contain multiple returns to A. The development and private-test partitions each contain 12 scenes and 72 sequences, with no shared scenes.

Each sequence includes a scene identifier, route identifier, initial RGB observation, rendered reference video, and source record. Camera records and additional modalities, such as depth or instance annotations, depend on the rendering backend.

Long-horizon videos. The long-horizon set uses the four multi-revisit routes with 45-frame observation intervals and 405-frame transitions. Across the 12 reconstructed scenes, this gives 48 sequences and 120,960 frames, or 126 minutes at 16 FPS. These videos use longer intervals than the short reference release; they are a separate set of sequences.

C.2 Evaluation Protocol

Evaluation coverage. We evaluate all cases in PersistBench, including every scene and registered route in the short reference release and the complete long-horizon set. The two trajectory-duration settings are evaluated separately. Within each setting, a K-return comparison includes every trajectory with at least K returns to place A. This complete eligible set remains fixed across all reported revisit orders and compared methods. The two-return analysis includes all routes with at least two returns; the three-return analysis includes all ABABACA and ABACADA routes. The main multi-revisit comparisons and module ablations use the long-horizon setting with the same eligibility rule.

Frame pairing. Each return is compared with the model's own initial visit to that place. The reference remains fixed throughout the video. We use the first nine frames of each observation interval and pair frames by their interval offset. In a long ABACA video, the three A intervals begin at frames 0, 900, and 1800. The evaluated pairs are $(i, 900 + i)$ and $(i, 1800 + i)$ for $i = 0, \ldots, 8$,

using zero-based frame indices. Pairing is set by the camera trajectory, without searching for similar frames in the generated output.

Revisit-order retention. For one method, let $I_{c,0,i}$ be frame i of the initial visit in case c, and $I_{c,k,i}$ the paired frame at revisit k. Let $\phi(I)$ denote the ℓ_2-normalized DINO (Caron et al., 2021; Oquab et al., 2023) feature of image I. We compute per-return and aggregate retention as

$$\begin{aligned} s_{c,k} &= \frac{1}{9}\sum_{i=0}^{8} \phi(I_{c,0,i})^\top \phi(I_{c,k,i}), \\ R_k &= \frac{1}{|\mathcal{C}_K|}\sum_{c\in\mathcal{C}_K} s_{c,k}, \qquad k = 1,\dots,K, \end{aligned} \tag{7}$$

where $\mathcal{C}_K$ contains all trajectories supporting at least K returns within the trajectory-duration setting being evaluated. The feature extractor and preprocessing are identical for all methods. The same complete eligible set is used for $R_1,\dots,R_K$ and for all compared methods, following Section C.2.

Maximum revisit degradation. We measure the largest decrease on the aggregate retention curve:

$$\mathrm{MRD}_{\mathrm{curve}} = \max_{1\le k<\ell\le K} [R_k - R_\ell]_+, \qquad [x]_+ = \max(x, 0). \tag{8}$$

MRD is computed after case averaging and uses a fixed return horizon $K \ge 2$. A later recovery does not cancel an earlier decline. For three returns, the metric is $\max(0, R_1 - R_2, R_1 - R_3, R_2 - R_3)$.

Additional retention summaries. We also report mean retention, endpoint change, and worst-return retention in the additional analyses:

$$\begin{aligned} \overline{R}_K &= \frac{1}{K}\sum_{k=1}^{K} R_k, \\ D_K^{\mathrm{end}} &= R_1 - R_K, \\ W_K &= \frac{1}{|\mathcal{C}_K|}\sum_{c\in\mathcal{C}_K} \min_{1\le k\le K} s_{c,k}. \end{aligned} \tag{9}$$

Endpoint change can be negative when final-return retention improves. Worst-return retention takes the minimum within each case before averaging.

Additional metrics and aggregation. LPIPS (Zhang et al., 2018) and return SSIM (Wang et al., 2004) use the same initial-visit frame pairs. GT SSIM instead compares generated frames with the rendered reference at the registered pose. PSNR is reported in dB. Histogram correlation and SIFT (Lowe, 2004) matching compare appearance across paired observations. Action and Coverage denote optical-flow direction agreement with the reference video and valid-motion coverage, respectively.

Scores are averaged first over frame pairs and then over cases. Since each scene contributes the same route patterns within a reported cohort, case averaging gives each scene equal weight. Aggregation uses unrounded scores. Missing runs, out-of-memory failures, and invalid metric inputs are recorded separately; missing values are not assigned zero scores. Case identifiers, video hashes, frame-pair indices, and per-return scores are retained with the evaluation records.

Additional peak storage. For each trajectory c, we measure the peak storage of Native and the peak of method m on the same LingBot-World 2.0 backbone, hardware, parallel configuration, and generation settings. Let these peaks be P_c^{Native} and P_c^m, in GiB. The per-trajectory increment is

$$\Delta P_c^m = P_c^m - P_c^{\mathrm{Native}}. \tag{10}$$

The tables report the mean of these paired differences over the relevant evaluation set. The storage counter and device aggregation are identical within each pair. This is a difference of run-level peaks, not a direct count of memory-bank tensors or the peak of a pointwise difference. Native is marked

Table 9: Shared-backbone generation settings.

Parameter	Setting
Backbone	LingBot-World 2.0, 14B causal-fast
Generation seed	42
Latent chunk size	3 frames
Sampling shift	10.0
Denoising evaluations	4
Scheduler indices	$[0, 179, 358, 679]$
Nominal size argument	`480*832`
Shared-backbone hardware	8 NVIDIA A800 GPUs, 80 GB per GPU
Model sharding	DiT/T5 FSDP
Sequence parallelism	8-way Ulysses
Module-ablation parallelism	4-way Ulysses
MBench output length	393 decoded frames
PersistBench output resolution	832×464
PersistBench frame rate	16 FPS
PersistBench long-horizon lengths	1,845 or 2,745 decoded frames

Ref. in the tables and has zero increment by definition. A negative increment would indicate a lower peak than Native.

Percentage reductions compare the reported increments ΔP; they do not express reductions in total system memory. Configured cache capacities in Section D.2 describe the memory policies and are separate from these measured increments.

D Additional Experimental Details

D.1 Shared-Backbone Experimental Configuration

Evaluation setting. The shared-backbone comparisons in Section 5.1 use the frozen LingBot-World 2.0 causal-fast checkpoint. Memory policies are implemented through its native cache and historical-memory interfaces. Within each comparison, the checkpoint, initial observation, prompt, camera controls, generation seed, and sampling settings are shared. The method-specific capacities in Section D.2 are not constrained to a common byte or token budget.

Generation configuration. We use the lingbot-world-v2-14b-causal-fast checkpoint with the settings in Table 9. Each chunk uses four denoising evaluations at scheduler indices $[0, 179, 358, 679]$. These indices refer to the shifted scheduler, not directly specified noise levels.

Table 9 distinguishes the shared-backbone and module-ablation parallel configurations. Results and Native peak references from different configurations are not pooled.

The nominal size argument specifies the generator's size setting; the decoded PersistBench resolution is reported separately.

Evaluation coverage. All cases in MBench and PersistBench are evaluated. Each of the ten reported MBench dimensions is computed over its complete applicable case set. Category aggregation and the separate auxiliary diagnostic checks are described in Section D.3. PersistBench covers all scenes and registered routes, with short-reference and long-horizon cases evaluated separately. Each revisit comparison uses all trajectories supporting its return horizon, as defined in Section C.2.

D.2 Baseline Adaptations

We implement the shared-backbone methods through LingBot-World 2.0's native historical-memory interface. The adaptations retain each method's memory policy while using the same frozen generator. Table 10 summarizes the method-specific settings. Unless stated otherwise, frame-based capacities refer to latent frames, not decoded RGB frames.

Table 10: Memory settings of the shared-backbone adaptations. Frame counts refer to latent frames. Capacities are method-specific.

Method	Memory configuration
Native	Configured sink and six recent frames; no external retrieval.
Full KV	Complete temporal cache; finite local-attention restriction disabled.
Deep Forcing	Physical cache: 21 frames; sink: 10; compression capacity: 16; query window: 4; maximum Top-C reuse: 7.
MemRoPE	Cache: 12 frames; sink: 3; long EMA: 1; short EMA: 1; recent: 4; current: 3; EMA rates: 0.01 and 0.1.
Infinity-RoPE	Rolling cache: 6 frames; frozen sink: 1; generation block: 3.
WorldKV	Append-only bank; retrieval: 18 frames; pooled compression keep ratio: 0.5.
Context-as-Memory	First-observation storage; retrieval: 18 frames; RoPE correction; pose thresholds: 0.25 and 3°; minimum revisit gap: 3 chunks.
SRAM	Place-indexed bank; retrieval: 18 frames; learned selection; 64 places; 181,440 tokens per layer; 2,835 tokens per place; pose thresholds: 0.25 and 3°; initial-write fraction: 1.0; CPU storage.

Native and Full KV. Native uses the generator's sink and recent-history cache without an external bank. Full KV disables the finite local-attention restriction and retains the complete temporal cache (Xiao et al., 2024). It is an additional full-history reference in Table 1, not the module-removal variant without spatial organization.

Deep Forcing. Sink retention and attention-informed cache selection operate on LingBot's projected native keys and values. Sink and recent tokens are handled separately from the attention-ranked historical candidates. Cache capacity, query window, and reuse settings are listed in Table 10.

MemRoPE. The cache combines sink, long-term EMA, short-term EMA, recent, and current frames. Updates use an unrotated cache representation with dynamic block-relative rotary positions. The table gives the allocation and both EMA rates.

Infinity-RoPE. Rolling-cache updates and block-relative positional handling are applied through LingBot's attention path. A frozen sink is retained as the cache is refreshed between generation blocks.

WorldKV. The adaptation retrieves native history by pose from an append-only bank. Store-time pooled compression applies to non-anchor history, and retrieved entries are supplied directly to the generator's attention interface.

Context-as-Memory. The adaptation stores first observations, retrieves native memory by camera pose, and applies RoPE correction before reuse. It transfers the retrieval policy rather than reproducing the original RGB-context and field-of-view pipeline. In this implementation, the first-write path can retain more records than the configured place-limit argument; that argument does not enforce SRAM's effective 64-place bound.

SRAM. SRAM uses the same pose-association thresholds and nominal retrieval budget as the spatial-retrieval adaptation. A learned correspondence predictor selects entries before they are written to the bounded CPU-resident bank. Its place and token capacities, and the configured initial-write fraction, are listed below. The configured fraction alone does not specify the number of entries accepted on a particular first visit.

D.3 Detailed MBench Results

Category-level summary. Table 11 gives the compact category summary for the nine methods listed, with each component evaluated over all applicable MBench cases. The ten component scores, including the additional Full KV reference, appear in Table 1. Entity averages Human Identity, Human Appearance, Object Texture, and Object Geometry; Environment averages Epipolar, Reprojec-

Table 11: MBench category scores. Best and second-best values within each group are shown in **bold** and *italics*, respectively.

Base model	LingBot-World 2.0							Wan2.1-based systems	
Method	Native	Deep Forcing	MemRoPE	Infinity-RoPE	WorldKV	Context-as-Memory	SRAM (Ours)	Infinite-World	CaR
Entity ↑	0.2289	0.3457	0.2172	0.1482	0.2750	*0.3555*	**0.3562**	*0.3479*	**0.3491**
Environment ↑	0.4836	0.4978	0.4948	0.4676	0.4957	*0.5218*	**0.5251**	*0.5216*	**0.5350**
Causal ↑	**0.3695**	0.3620	0.3683	0.3592	0.3620	0.3680	*0.3691*	**0.4304**	*0.3481*

Table 12: Additional MBench results. Best and second-best SP/PC scores within each group are shown in **bold** and *italics*, respectively.

Setting	LingBot-World 2.0 backbone								External systems	
Metric	Native	Full KV	Deep Forcing	MemRoPE	Infinity-RoPE	WorldKV	Context-as-Memory	SRAM (Ours)	Infinite-World	CaR
SP ↑	0.5000	*0.5313*	*0.5313*	0.4688	0.5000	0.4063	*0.5313*	**0.6563**	**0.2500**	*0.2188*
PC ↑	0.7000	*0.7750*	0.7250	0.7000	0.6750	0.7250	0.7250	**0.8250**	**0.4750**	*0.4500*
OG_7	0.0301	0.0183	0.0284	0.0279	0.0191	0.0218	0.0298	0.0278	0.1074	0.0616

tion, Style, and Lighting. The reported Causal summary averages Camera and Prompt Interaction. Category aggregation uses unrounded component scores.

Additional causal and geometry evaluations. The following targeted checks are reported separately from the full-benchmark evaluations above. The State Progress (SP) and Progress Correctness (PC) diagnostics use the same DeepSeek Flash judge for all methods, with eight judgments per dimension and method. An ordinal score $s \in \{1, \ldots, 5\}$ is normalized as $(s-1)/4$ for SP and $s/5$ for PC.

We also re-evaluate Object Geometry with corrected temporal alignment on seven common cases with valid measurements, denoted by OG_7. These results are reported separately in Table 12. SP and PC are not included in the Camera–Prompt category summary in Table 11. Likewise, OG_7 does not replace OG in Table 1. These diagnostics have their own judge and valid-case selection and are not merged into the ten-dimensional results.

Table 12 reports SP and PC scores of 0.6563 and 0.8250 for SRAM, respectively, placing it first on both measures within the shared-backbone group. Full KV, Deep Forcing, and Context-as-Memory tie for second on SP at 0.5313, while Full KV ranks second on PC at 0.7750. These evaluations characterize state progression and its correctness separately from the Camera and Prompt Interaction scores used in the main Causal aggregate.

D.4 Detailed PersistBench Results

Retention across three revisits. Table 13 adds LPIPS to the three-return comparison for the six listed methods. Their DINO and $\mathrm{MRD}_{\mathrm{curve}}$ values match Table 2. The comparison uses all long-horizon trajectories supporting three returns, with the fixed initial-visit pairing in Section C.2. LPIPS is reported separately from the DINO-based degradation measure.

The additional LPIPS results differ from the DINO ranking. WorldKV has the lowest first-return distance, 0.7253, while SRAM has the lowest second- and third-return distances, 0.7304 and 0.7192. SRAM's LPIPS improves across these returns even though its DINO similarity decreases slightly. The two metrics are therefore reported separately.

Two-return analysis. Table 14 uses all long-horizon trajectories supporting at least two returns, following Section C.2. This includes routes that do not provide a third return, so its case set differs from the three-return comparison. Mean, Worst, and Endpoint Drop follow Eq. 9. Endpoint Drop is the signed endpoint change, not $\mathrm{MRD}_{\mathrm{curve}}$.

SRAM has the highest mean and worst-return DINO scores, at 0.2624 and 0.1983, respectively. Its aggregate DINO similarity rises from 0.2507 on the first return to 0.2741 on the second. MemRoPE

Table 13: Three-return results on PersistBench. Best and second-best scores are shown in **bold** and *italics*, respectively.

Method	DINO ↑			LPIPS ↓			$\mathrm{MRD}_{\mathrm{curve}}$ ↓
	1st	2nd	3rd	1st	2nd	3rd	
Native	0.1665	0.1089	0.1112	0.7385	*0.7472*	*0.7420*	0.0576
Deep Forcing	0.0983	0.1088	0.0601	0.7593	0.7607	0.7798	0.0487
MemRoPE	0.1999	0.1831	*0.2362*	*0.7263*	0.7510	0.7592	**0.0168**
WorldKV	0.2301	0.1393	0.1195	**0.7253**	0.7536	0.7729	0.1106
Context-as-Memory	*0.2384*	*0.2174*	0.2214	0.7344	0.7480	0.7549	0.0210
SRAM (Ours)	**0.2605**	**0.2547**	**0.2428**	0.7403	**0.7304**	**0.7192**	*0.0177*

Table 14: Two-return diagnostics on PersistBench. Best and second-best scores are shown in **bold** and *italics*, respectively.

Method	DINO ↑				Endpoint Drop ↓	Mean LPIPS ↓
	1st	2nd	Mean	Worst		
Native	0.1679	0.1034	0.1356	0.0907	+0.0645	0.7403
Deep Forcing	0.1053	0.1049	0.1051	0.0678	+0.0004	0.7625
MemRoPE	*0.2092*	*0.1982*	*0.2037*	*0.1524*	+0.0110	**0.7266**
WorldKV	0.2020	0.1432	0.1726	0.1161	+0.0588	0.7426
Context-as-Memory	0.1918	0.1940	0.1929	0.1418	*−0.0021*	0.7500
SRAM (Ours)	**0.2507**	**0.2741**	**0.2624**	**0.1983**	**−0.0234**	*0.7339*

ranks second in mean and worst-return DINO and has the lowest mean LPIPS, 0.7266. SRAM's mean LPIPS of 0.7339 is second-best and lower than Native's 0.7403. By contrast, Deep Forcing's near-zero endpoint change accompanies the lowest mean retention, illustrating why a small change alone does not imply strong recall.

Appearance and motion diagnostics. Table 15 reports the additional appearance and motion measures over the complete long-horizon evaluation set, using the definitions in Section C.2. The table separates comparison with the first generated visit from comparison with rendered references. Motion diagnostics are reported alongside, rather than combined with, the appearance scores.

MemRoPE leads histogram correlation, SIFT matching, and both rendered-reference metrics. Deep Forcing has the highest return SSIM, while Context-as-Memory has the highest return PSNR. SRAM has the highest Action and Coverage scores, 0.4240 and 0.8789, but does not lead the pixel-level or feature-matching measures. Its higher DINO retention therefore does not extend uniformly to all appearance diagnostics. The motion scores describe separate aspects of the generated trajectories and are not direct measures of geometric recovery.

E Qualitative Comparisons

Figures 6 and 7 provide additional qualitative comparisons between SRAM and the baseline methods. Each comparison shows the initial visit and subsequent returns to the same place. For each method, the initial generated observation serves as the reference for inspecting changes in scene appearance, object details, and local structure across revisits.

F Additional Visual Results

Figure 8 presents more SRAM outputs across scenes and camera trajectories. The examples include both revisited places and observations made during intervening exploration, showing more of each generated sequence beyond the frames selected for the baseline comparisons.

Table 15: Appearance and motion diagnostics on PersistBench. Best and second-best scores are shown in **bold** and *italics*, respectively.

Method	First-visit reference ↑				Rendered reference ↑		Motion ↑	
	SSIM	PSNR	Hist.	SIFT	SSIM	PSNR	Action	Coverage
Native	0.4588	8.5294	0.1611	0.3795	0.4736	*8.8182*	0.3988	0.8086
Deep Forcing	**0.4900**	8.6796	0.0955	*0.3876*	*0.4825*	8.4273	0.3426	0.7237
MemRoPE	*0.4826*	*9.1844*	**0.2590**	**0.4041**	**0.4859**	**9.1178**	0.3955	0.8135
WorldKV	0.4699	9.0630	0.1389	0.3766	0.4502	8.2227	0.3841	0.8001
Context-as-Memory	0.4501	**9.3075**	*0.2096*	0.3840	0.4220	8.0309	*0.4130*	*0.8610*
SRAM (Ours)	0.4413	8.9215	0.1958	0.3726	0.4263	8.2282	**0.4240**	**0.8789**

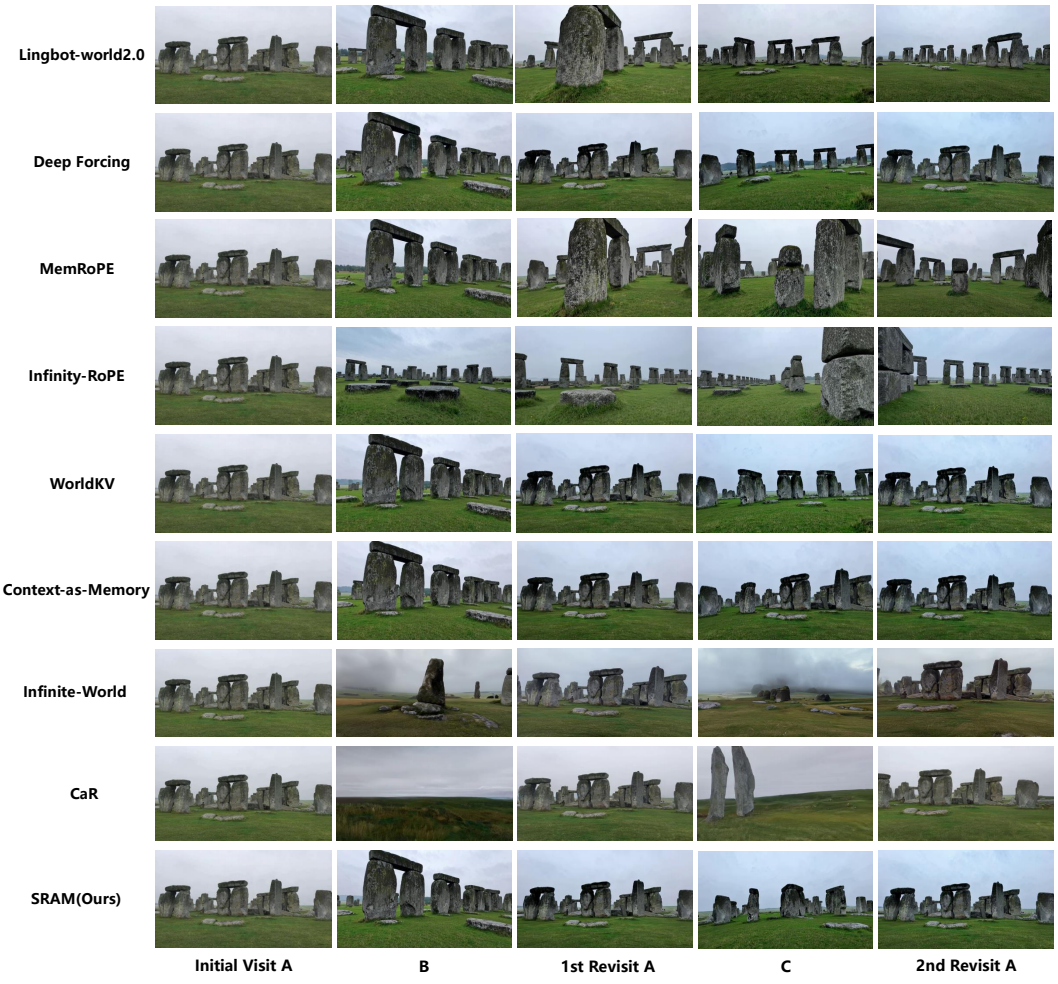

Figure 6: Further qualitative comparisons across scenes. Revisit frames are compared with each method's initial generated observation.



Figure 7: Further qualitative comparisons across scenes. Revisit frames are compared with each method's initial generated observation.



Figure 8: Additional visual results of SRAM across scenes and trajectories.