GRAPHMEM: A TRAINING-FREE GRAPH RETRIEVAL AND MEMORY CACHING FRAMEWORK FOR CONSISTENT WORLD GENERATION

Anonymous authors
Paper under double-blind review



Figure 1: **Preserving scene consistency under long-horizon trajectory revisits.** *Top:* The camera travels a closed loop and returns to its starting viewpoint while our method preserves scene consistency throughout. *Down:* The camera sweeps back and forth left then right, or right then left, and our method retains the regions that leave the field of view.

ABSTRACT

We present **GraphMem**, a training-free, graph-structured visual memory framework for long-horizon autoregressive video world models. Rolling KV caches enable efficient chunk-wise generation but discard earlier scene evidence once it leaves the recent window, causing inconsistency when the camera revisits a previously observed region. Full-history conditioning retains this evidence but makes the active context and inference cost grow with video length. Our key idea is to dynamically retrieve relevant history based on the current state. GraphMem organizes pixel-space history into a Vision Memory Graph. Conditioned on the current generation state, dynamic graph retrieval activates a compact set of relevant and complementary historical views. The retrieved history is encoded with timestep-aligned noise and prefilled through the frozen backbone, producing layer-wise history KV caches in its native attention space. Dynamic retrieval keeps the active context bounded, while strided prefill amortizes the additional computation. Across diverse backbones, long-horizon memory performance increases by about 2% on average and shows better revisit consistency.

1 INTRODUCTION

Recent advances in video diffusion models have substantially improved visual fidelity, motion realism, and camera controllability, enabling applications such as interactive world exploration (Tong et al., 2026; Chen et al., 2026a), embodied simulation, and real-time content creation (Wang et al.,

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.

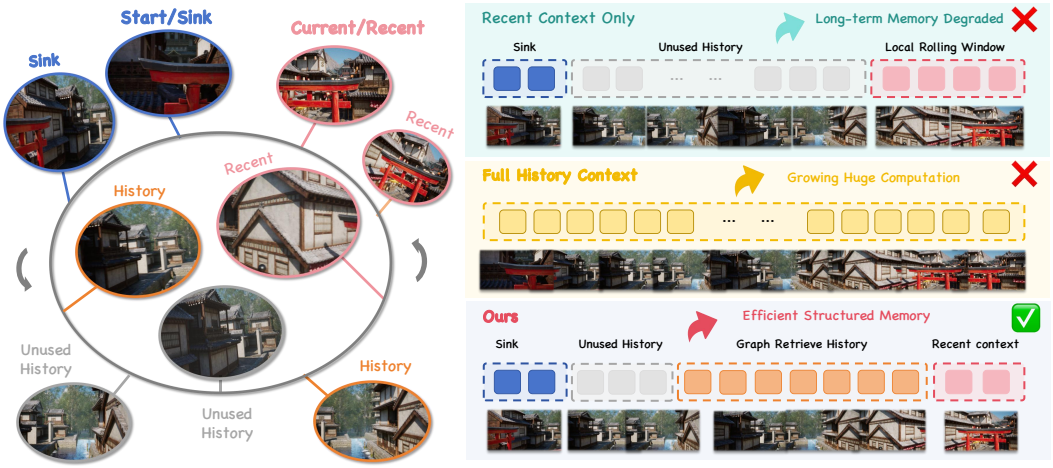


Figure 2: **Motivation for graph memory retrieval. (1) Recent context only.** Sink tokens and a rolling window keep attention bounded but lose intermediate views. **(2) Full history context.** All past views remain accessible, while the active cache and attention cost grow with video length. **(3) Ours.** GraphMem retrieves relevant keyframes into a KV Cache when needed.

2026b; Lin et al., 2026; Liu et al., 2026a; Zhu et al., 2026d). However, generating longer videos does not necessarily create a persistent world. Most long-horizon video world models operate autoregressively, causing prediction errors to propagate and accumulate. Moreover, as early observations leave the active context, previously established geometry, appearance, and object layouts may be forgotten or reconstructed inconsistently, particularly when the camera revisits a region after a long interval.

The underlying challenge is historical context selection. Rolling windows preserve recent observations for local continuity but may exclude earlier views that are more relevant to the current scene. This reveals a fundamental mismatch: **temporal proximity does not necessarily indicate scene relevance**. Without access to relevant historical evidence, the model must reconstruct established scenes from incomplete context, leading to progressive spatial and appearance drift. Persistent world generation therefore requires a memory that both preserves historical observations and retrieves the right evidence when needed. Fig. 2 contrasts two common context policies. Sink tokens and a rolling window bound attention cost but evict intermediate observations that may be needed during revisits; a larger window only delays their eviction. Full-history conditioning retains those views but makes the active KV cache and attention cost grow with video length, while introducing redundant context. The challenge is therefore to preserve detailed history outside the active context and retrieve only the views relevant to the current camera state.

Recent work on long-horizon memory follows four partly overlapping directions. EchoMemory (King et al., 2026) provides a useful insight: raw context offers strong memory capacity, compression may remove evidence needed for revisits, and preserved evidence remains ineffective unless it can be reliably accessed. **(1) Training-based methods.** Existing approaches enhance long-horizon generation by learning memory retrieval, compressing historical context, or introducing explicit spatial representations. Context-as-Memory (Yu et al., 2025; 2026a; Yang et al., 2025), RELIC (Hong et al., 2025), Matrix-Game 3.5 (Wang et al., 2026c), and AlayaWorld (Team et al., 2026a) integrate learned retrieval or memory conditioning into the generation process, enabling selective access to historical observations. FramePack (Zhang et al., 2026a; Gu et al., 2025; Ji et al., 2025), and Infinite-World (Wu et al., 2026a) control context growth through packing, sparse activation, or hierarchical compression. Other methods, such as SPMem (Wu et al., 2026b; Wang et al., 2026a) and AnchorWeave (Wang et al., 2026d) organize historical observations using explicit geometric representations to establish spatial correspondence. Despite their effectiveness, these methods generally require memory-specific training, backbone adaptation, or auxiliary geometric representations, limiting their applicability to pretrained video diffusion models. **(2) Training-free methods.** Recent approaches improve long-horizon consistency without additional model training. Warp-as-History (Wang & He, 2026), WorldKV (Yi et al., 2026), Closing the Loop (Ma et al., 2026), and WorldTrace (Wu et al., 2026c) reuse native historical pathways, cached internal states, or

virtual positional representations to incorporate previous observations during inference. Although these methods offer lightweight adaptation to pretrained generators, they typically represent memory as warped observations, independent temporal chunks, or compressed states. Consequently, they lack an explicit scene-level structure for organizing and retrieving spatially related visual evidence, particularly under long-range revisits.

Despite recent progress, existing training-free memory methods still face two key limitations. **(1) Flat similarity-based retrieval fails to capture relations among memories.** Many methods treat historical observations as independent entries and rank them solely according to their similarity to the current query. Such independent selection can repeatedly retrieve near-duplicate views while overlooking complementary observations connected through camera motion, spatial proximity, or previous scene revisits. **(2) A fixed retrieval budget cannot accommodate time-varying memory demands.** Existing methods often activate a preset amount of historical context at each generation step, despite substantial variation in the required memory. Local camera motion may be sufficiently supported by recent observations, whereas revisiting a previously observed region requires broader long-term evidence. Consequently, the retrieved context can be insufficient when long-range information is critical, yet redundant when the recent context is already adequate. These limitations expose a fundamental gap between preserving history and making it accessible: storing past observations alone does not constitute an effective memory. Long-term memory should instead organize observations according to their scene-level relationships and allocate its retrieval capacity based on the current generation state.

Although videos are generated sequentially, their observations are connected not only temporally but also through spatial proximity, camera motion, and scene revisits. GraphMem leverages these relations to enable structured and demand-adaptive memory retrieval, and then translates the retrieved evidence into representations directly usable by a frozen generator. **First, GraphMem organizes visual history using a Vision Memory Graph.** It stores pixel-space keyframes in scene-centric nodes and connects them through temporal, spatial, and revisit relations. Given the current generation state, GraphMem combines global candidate search with relation-aware graph traversal to retrieve complementary observations, while dynamically adjusting the number of activated memories within a bounded budget. This design replaces independent similarity matching and preset retrieval budgets with structured, demand-adaptive recall. **Second, GraphMem incorporates the retrieved evidence through dynamic History KV prefill.** The selected keyframes are encoded by the frozen VAE, perturbed with noise aligned to the current denoising timestep, and replayed through the frozen Diffusion Transformer to construct layer-wise KV states in its native attention space. Memory-specific rotary positions align historical states with current queries, while strided refresh amortizes the additional prefill cost across generation steps. Together, graph retrieval determines *what* and *how much* history to recall, whereas History KV prefill determines *how* the frozen generator consumes the retrieved evidence, all without parameter updates or modifications to the original denoising schedule.

We evaluate GraphMem on long-horizon revisit trajectories and integrate it into five autoregressive video/audio world-model backbones. Across all five, GraphMem improves long-horizon memory performance without parameter updates. Qualitative comparisons show that previously observed structures are more faithfully recovered when the camera returns after a long temporal gap.

Our contributions are summarized as follows:

- We introduce **GraphMem**, a training-free memory framework that indexes visual history with a scene-centric Vision Memory Graph, recasting long-horizon conditioning from unbounded context accumulation as bounded scene retrieval.
- Technically, we propose **dynamic history KV prefill**, a training-free mechanism that makes retrieved visual evidence directly usable by a frozen backbone. It addresses the representation mismatch caused by projecting external memory into the backbone's attention space. Retrieved history is therefore encoded with timestep-aligned noise and prefilled through the frozen backbone to construct layer-wise KV states in its native attention space.
- We demonstrate **cross-backbone gains in long-horizon memory** across diverse autoregressive video world models. Experimental results and attention analysis collectively show improved scene persistence without memory-specific training or changes to the original denoising procedure.

2 Related Work

2.1 Few-Step Autoregressive Video Generation

Few-step autoregressive (AR) video diffusion enables low-latency streaming by distilling bidirectional models into causal generators, as in DMD-based CausVid (Yin et al., 2023; Liu et al., 2025; Yin et al., 2025). Self-Forcing (Huang et al., 2025a) reduces exposure bias through self-generated rollouts, while Causal Forcing (Zhu et al., 2026c) and Causal Forcing++ (Zhao et al., 2026b) improve few-step student initialization. Rolling Forcing (Liu et al., 2026b), Reward Forcing (Lu et al., 2026), and OPSD-V (Liu et al., 2026a) further strengthen long-rollout stability. Yet smoother generation cannot recover a previously observed scene once its views leave the active context. This motivates training-free retrieval of scene-relevant history for existing few-step AR generators.

2.2 Retrieval-Augmented Memory for Interactive Video World Models

Interactive video world models (Ma et al., 2025) must recover earlier scenes when the camera returns. RAGME (Peruzzo et al., 2025) and MotionRAG (Zhu et al., 2026a) retrieve external motion references, whereas persistent worlds require access to their own generated history. Context-as-Memory (Yu et al., 2025) and WorldMem (Xiao et al., 2026) retrieve historical frames; RELIC (Hong et al., 2025) uses compact camera-aware tokens; and ReWorld (Chen et al., 2026b) and Wonder (Xu et al., 2026) improve access through pose-indexed memory or sparse attention. WorldStereo (Zhang et al., 2026c) and MosaicMem (Yu et al., 2026b) provide stronger spatial alignment through 3D structure. These approaches establish the value of long-term retrieval, but often rely on learned memory interfaces, compressed evidence, or geometric reconstruction. GraphMem instead indexes pixel-space keyframes by scene and prefills selected views into a frozen generator's native KV cache, enabling training-free memory augmentation for few-step AR models.

3 Method

We study a pretrained video diffusion model that generates a video one chunk at a time. Let $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, where each chunk $\mathbf{x}_n \in \mathbb{R}^{F\times H\times W\times 3}$ contains F frames of resolution $H \times W$. Before generating chunk n, all earlier chunks $\mathbf{X}_{<n}$ have been produced, but the model can attend to only the history placed in its active KV cache $\mathcal{C}_n$. Given a text prompt $\mathbf{c}$ and the camera specification $\mathbf{p}_n$ for the next chunk, generation follows

$$\mathbf{x}_n \sim p_\theta\left(\mathbf{x}_n \mid \mathbf{c}, \mathbf{p}_n, \mathcal{C}_n\right), \tag{1}$$

where θ denotes the frozen model parameters.

GraphMem addresses unstructured and fixed-count memory retrieval through structured recall and native KV-cache integration. Sec. 3.1 constructs a Vision Memory Graph that connects historical keyframes through temporal, spatial, and revisit relations, enabling relation-aware and demand-adaptive retrieval. Sec. 3.2 converts the retrieved keyframes into timestep-aligned KV caches that can be directly consumed by the frozen diffusion model. Finally, Sec. 3.3 presents the inference procedure, including autoregressive generation, online graph updates, and efficiency analysis.

3.1 Stage 1: Memory Retrieval

A generated video is ordered in time, but the views needed for future generation need not be temporally close. When the camera revisits a region, an observation from many chunks earlier may be more useful than the latest frames. We therefore index historical keyframes by scene relationships as they are generated, rather than relying on their positions in a chronological queue.

Graph construction. After generating chunk n, we maintain

$$\mathcal{G}_n = \left(\mathcal{V}_n, \mathcal{E}_n^{\text{temp}}, \mathcal{E}_n^{\text{spat}}, \mathcal{E}_n^{\text{rev}}\right), \tag{2}$$

where $\mathcal{V}_n$ is the node set and the three edge sets record temporal transitions, spatial proximity, and revisits. Each node v_i stores a set of pixel-space keyframes $\mathcal{I}_i$, their camera poses $\mathcal{P}_i$ and generation indices $\mathcal{T}_i$, and a representative visual descriptor $\bar{\mathbf{f}}_i$.

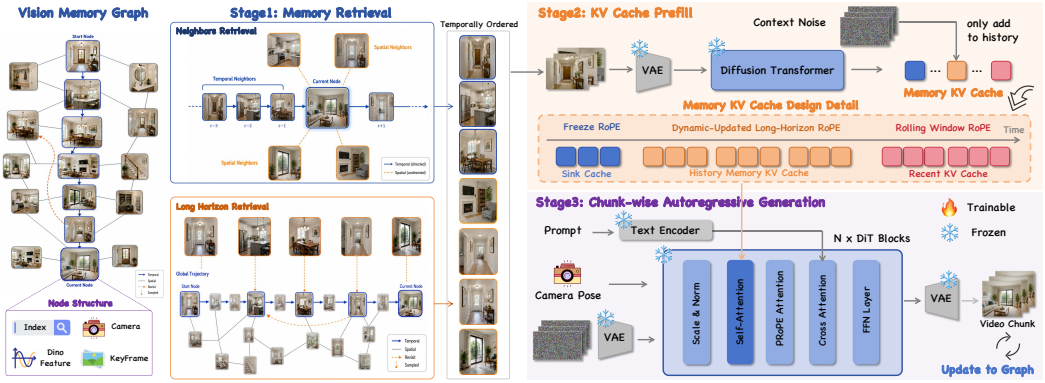


Figure 3: **Overview of GraphMem. Stage 1:** Generated keyframes are incrementally organized into scene-centric nodes connected by temporal, spatial, and revisit edges. Before generating each chunk, query-adaptive retrieval combines local graph traversal with global search to select relevant, non-redundant views and restore their temporal order. **Stage 2:** The retrieved keyframes are encoded into latents, perturbed with timestep-aligned context noise, and prefilled through the frozen DiT to construct a layer-wise History KV Cache in its native attention space. Memory-specific RoPE positions make this distant evidence accessible to current queries without training or modifying the denoising procedure. **Stage 3:** The model generates the current chunk and then asynchronously writes it back to the graph, closing the autoregressive memory loop.

For each new keyframe I at camera pose $\mathbf{p}$, a frozen DINO encoder extracts $\mathbf{f} = \Phi(I)$. We match the keyframe only against nodes that satisfy the pose threshold, then choose the most visually similar eligible node:

$$\mathcal{J}(\mathbf{p}) = \{j : v_j \in \mathcal{V}_{n-1},\ d_{\text{pose}}(\mathbf{p}, \mathcal{P}_j) \le \tau_{\text{pose}}\}, \qquad j^\star = \arg\max_{j \in \mathcal{J}(\mathbf{p})} \cos\big(\mathbf{f}, \bar{\mathbf{f}}_j\big). \tag{3}$$

Here, $\mathcal{V}_{n-1}$ is the node set of $\mathcal{G}_{n-1}$ before inserting I; $d_{\text{pose}}(\mathbf{p}, \mathcal{P}_j)$ measures the distance to the closest pose stored in v_j; and cos denotes cosine similarity. If $\mathcal{J}(\mathbf{p})$ is nonempty and the similarity to $v_{j^\star}$ exceeds τ_{vis}, we add I, its pose, and its generation index to that node and update its descriptor. Otherwise, we create a new node. Pose lookup and DINO encoder search are accelerated with a KD-Tree Ram & Sinha (2019) and FAISS Johnson et al. (2019), respectively. The two thresholds prevent visually similar but spatially separate observations from being merged solely on appearance. Directed temporal edges connect successive node assignments, while spatial edges connect nodes with nearby camera poses. When a new keyframe matches a node last visited more than Δ_{rev} chunks ago, a revisit edge links the preceding trajectory node to that historical node. Together, these edges capture exploration order, spatial proximity, and long-range returns.

Graph retrieval. Before generating chunk n, we query $\mathcal{G}_{n-1}$ with the descriptor $\mathbf{f}_{n-1}$ of the latest keyframe and the target camera pose $\mathbf{p}_n$. Local retrieval collects temporal neighbors of the latest node and pose neighbors near $\mathbf{p}_n$. Global retrieval uses FAISS to find visually related nodes across the graph and samples representative key nodes to cover regions beyond the local neighborhood. Together, these routes provide recent, spatially adjacent, and temporally distant candidates.

Let $\mathcal{R}_n$ be the union of these candidate nodes. We select keyframes through both Top-K and MMR metrics to balance similarity and diversity while suppressing near-duplicate views, then restore their original generation order:

$$\mathcal{M}_n = \text{TemporalSort}\left(\text{Select}(\mathcal{R}_n; \mathbf{f}_{n-1}, \mathbf{p}_n)\right) = \left(I_{m_1}, \ldots, I_{m_{K_n}}\right), \qquad K_n \le K_{\max}. \tag{4}$$

Here, m_i specifies the original generation order of keyframe I_{m_i}, K_n is the number of selected keyframes, and $K_{\max}$ caps the active history budget. A relevance threshold allows K_n to vary with the current view: a straightforward continuation may require little distant history, whereas a revisit can activate more. Descriptors and poses make the graph searchable, but the memory evidence itself remains in pixel space. The retrieved keyframes retain visual detail and can be encoded by different video backbones when constructing History KV states. Only these selected frames enter the active attention context; the graph retains the remaining history for later queries.

3.2 Stage 2: KV Cache Prefill

Graph retrieval identifies which earlier views are relevant to the next chunk. The remaining question is how a frozen video diffusion model can use them. Projecting retrieved features into a separate memory branch would require learning a new conditioning interface. GraphMem instead passes the selected keyframes through the pretrained backbone itself, constructing historical keys and values in the attention space that the model already uses.

Native history KV prefill. Let $\mathcal{M}_n = (I_{m_1}, \ldots, I_{m_{K_n}})$ be the temporally ordered keyframes retrieved for chunk n. A frozen VAE encoder $\mathcal{E}$ converts each frame into a clean latent $\mathbf{z}_{m_i} = \mathcal{E}(I_{m_i})$. At diffusion timestep t, we apply the backbone's noise schedule to the retrieved latents:

$$\mathbf{z}^t_{m_i} = \alpha_t \mathbf{z}_{m_i} + \sigma_t \boldsymbol{\epsilon}_{m_i}, \qquad \boldsymbol{\epsilon}_{m_i} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{5}$$

where α_t and σ_t are the signal and noise coefficients. This makes the prefetched history compatible with the noise level used to construct the current denoising state; the sink and recent caches retain their original treatment.

Memory-specific RoPE. The retrieved history is assigned dynamic relative positions rather than its original absolute temporal positions, serving as supplementary history context to the recent rolling cache. Specifically, we order the retrieved keyframes chronologically and assign their rotary positions immediately after the last position of the active recent cache. The sink positions remain fixed, the recent positions follow the rolling generation window, and the history positions are recomputed whenever the retrieved set changes. This design preserves the internal order of retrieved observations while allowing them to complement the recent context without positional overlap.

Dynamic history injection. Let $\mathbf{z}^t_{\text{hist}} = (\mathbf{z}^t_{m_1}, \ldots, \mathbf{z}^t_{m_{K_n}})$ be the stacked noisy latents of the retrieved keyframes. At each refresh step, they are passed through the frozen DiT:

$$\left(\mathbf{K}^t_{\text{hist}}, \mathbf{V}^t_{\text{hist}}\right) = \text{Prefill}_\theta\left(\mathbf{z}^t_{\text{hist}}, \mathbf{c}_{\text{hist}}\right), \tag{6}$$

where $\mathbf{c}_{\text{hist}}$ includes the corresponding camera conditions and rotary positions. This produces layer-wise history KV states using the backbone's native attention projections, without introducing additional projectors or learned fusion modules. During generation, the history states are appended to the active KV cache:

$$\mathbf{K}^t = \left[\mathbf{K}^t_{\text{sink}}; \mathbf{K}^t_{\text{recent}}; \mathbf{K}^t_{\text{hist}}; \mathbf{K}^t_{\text{cur}}\right], \qquad \mathbf{V}^t = \left[\mathbf{V}^t_{\text{sink}}; \mathbf{V}^t_{\text{recent}}; \mathbf{V}^t_{\text{hist}}; \mathbf{V}^t_{\text{cur}}\right]. \tag{7}$$

Only the current-chunk tokens produce queries, while the retrieved history supplies additional keys and values. The current chunk can therefore access long-term visual evidence through the original self-attention operation, without increasing the query length or modifying the denoising procedure.

3.3 Stage 3: Chunk-wise Autoregressive Generation

Generation. For each chunk n, GraphMem initializes a noisy latent and denoises it using the frozen diffusion transformer backbone. At each autoregressive denoising step, the current-chunk queries attend jointly to the sink cache, retrieved history cache, and recent KV cache.

Computational efficiency and online updates. After generating each chunk, GraphMem extracts its keyframes and inserts them into the Vision Memory Graph. Temporal edges connect consecutive chunks, while spatial and revisit edges are established according to their visual and camera relationships. These graph updates are executed asynchronously before the next retrieval. The sink and recent caches have fixed sizes, while graph retrieval activates at most $K_{\max}$ keyframes. Let L_{cur} denote the number of current-chunk tokens and L^n_{active} the number of active context tokens. The attention cost for generating one chunk over T denoising steps is

$$\mathcal{O}\left(T L_{\text{cur}}\left(L_{\text{cur}} + L^n_{\text{active}}\right)\right), \qquad L^n_{\text{active}} = L_{\text{sink}} + L_{\text{recent}} + L^n_{\text{hist}}. \tag{8}$$

Because L^n_{active} is bounded by the retrieval budget, the per-chunk attention cost does not increase with video length, even as the external graph continues to accumulate keyframes. To further reduce prefill overhead, we refresh the History KV Cache every s_{pf} denoising steps, reducing the number of frozen-DiT prefill passes from T to $N_{\text{pf}} = \left\lceil \frac{T}{s_{\text{pf}}} \right\rceil$. The stride s_{pf} therefore provides a controllable trade-off between timestep alignment and prefill efficiency. More retrieval efficiency comparisons and latency analysis are provided in Appendix 6.2.

Table 1: Quantitative comparison results among diverse baselines.

Method	Quality					Memory			
	Temporal Flickering ↑	Subject Consistency ↑	Dynamic Degree ↑	Motion Smoothness ↑	Background Consistency ↑	I2V Score ↑	Long-Horizon Score ↑	Memory Score ↑	Loop Closure ↑
External baselines									
minWM (Zhao et al., 2026a)	0.926	0.940	0.726	0.943	0.972	0.904	0.531	0.527	0.415
SANA-WM (Zhu et al., 2026b)	**0.988**	0.926	0.778	0.988	0.966	0.842	0.781	0.755	0.681
MatrixGame 3.0 (Wang et al., 2026c)	0.971	**0.975**	0.575	0.988	0.966	0.902	**0.812**	0.787	0.761
WorldKV (Yi et al., 2026)	0.926	0.935	0.990	0.954	0.948	0.894	0.768	0.742	0.642
AlayaWorld (Team et al., 2026a)	0.976	0.953	0.412	0.993	0.977	**0.943**	0.799	**0.791**	0.742
Ours: base world model + GraphMem									
HY-WorldPlay-1.5 (HunyuanWorld, 2025)	0.962	0.966	0.900	0.958	0.936	0.937	0.781	0.779	0.764
HY-WorldPlay-1.5 + GraphMem	0.977	0.952	0.877	0.974	0.957	0.930	0.788	0.786	**0.771**
Lingbot-World-1.5 (Team et al., 2026c)	0.922	0.922	0.985	0.949	0.944	0.855	0.749	0.715	0.616
Lingbot-World-1.5 + GraphMem	0.984	0.916	0.961	**0.985**	0.965	0.853	0.762	0.727	0.628
DreamX-World-1.0 (Team et al., 2026b)	0.970	0.934	0.861	0.976	0.968	0.917	0.755	0.734	0.650
DreamX-World-1.0 + GraphMem	0.973	0.939	0.892	0.978	0.969	0.914	0.783	0.775	0.692
Lingbot-World-v2 (Gao et al., 2026)	0.973	0.919	0.881	0.979	0.965	0.898	0.752	0.733	0.654
Lingbot-World-v2 + GraphMem	0.978	0.918	0.888	0.981	**0.974**	0.911	0.771	0.758	0.669
EchoWM (Zhang et al., 2026b)	0.919	0.910	0.987	0.948	0.947	0.847	0.754	0.738	0.625
EchoWM + GraphMem	0.928	0.921	**0.993**	0.951	0.957	0.862	0.772	0.743	0.652

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Evaluation Metrics.We assess video quality with VBench-I2V-Long (Huang et al., 2025b; 2024), reporting its motion, temporal-stability, background-consistency, and visual-quality metrics. To measure memory ability across temporal gaps, we additionally report three visual-similarity metrics: *Long-Horizon Score*,*Memory Score* and *Loop Closure Score* averages pose-matched comparisons between early and late chunks and compare the early and final chunks of trajectories that return to the initial region. More details are in Appendix 6.4.1. **Datasets.** We evaluate GraphMem on 100 revisit trajectories selected from the validation splits of Context-as-Memory (Yu et al., 2025) and DL3DV (Ling et al., 2024). The trajectories average approximately 30 seconds and include out-and-back motion, repeated lateral sweeps, and closed loops. All methods use the same prompts, reference images, camera trajectories, and output lengths. Experiments are conducted on the same memory budget across diverse backbones and inference with a batch size of one.

4.2 EXPERIMENT RESULTS

Quantitative Results. The upper block of Table 1 compares GraphMem with representative world-generation models. We include minWM, the official implementation of Causal Forcing Zhu et al. (2026c), as a standard few-step autoregressive baseline. The remaining methods represent three distinct memory paradigms. Matrix-Game 3.0 Wang et al. (2026c) combines camera-aware retrieval with long-horizon training and achieves strong Long-Horizon and Memory scores. WorldKV Yi et al. (2026) retrieves and compresses evicted KV chunks and achieves the highest Dynamic Degree among external baselines. SANA-WM Zhu et al. (2026b) achieves the best performance in terms of temporal consistency. AlayaWorld Team et al. (2026a) integrates compressed temporal context with spatial memory and obtains the strongest I2V Score among the baselines. These results provide a broad comparison across existing memory designs. However, differences in backbones, training data, and conditioning mechanisms prevent them from isolating the contribution of memory alone.

The lower block therefore reports controlled comparisons on the same pretrained backbones. For each pair, GraphMem preserves the backbone weights and uses identical prompts and camera trajectories. Adding GraphMem consistently improves Temporal Flickering, Motion Smoothness, and Background Consistency across all evaluated backbones. More importantly, it improves all three memory-oriented metrics. The average absolute gains are 0.017 in Long-Horizon Score, 0.018 in Memory Score, and 0.021 in Loop Closure Score. The consistent Loop Closure improvement is particularly relevant because it directly measures whether an early scene can be recovered after a long autoregressive rollout. The gains are concentrated on temporal stability and long-term scene recall, while the overall I2V Score remains largely unchanged. This result matches the intended role of GraphMem. Overall, the paired comparisons demonstrate that GraphMem provides a consistent memory improvement across different video generators.

Table 2: **Ablation study of memory organization and graph relations.**

Method	Temporal Flickering ↑	Subject Consistency ↑	Motion Smoothness ↑	Background Consistency ↑	I2V Score ↑	Long-Horizon Score ↑	Memory Score ↑	Loop Closure ↑
Queue	0.959	0.931	0.976	0.958	0.898	0.733	0.733	0.651
Bank(Temporal)	0.931	0.925	0.967	0.948	0.891	0.724	0.734	0.646
Bank(Spatial)	0.930	0.923	0.967	0.951	0.906	0.756	0.734	0.644
Ours (Graph)	**0.973**	**0.939**	**0.978**	**0.969**	**0.914**	**0.783**	**0.775**	**0.692**
w/o Temporal Edges	0.971	0.935	0.972	0.964	0.912	0.764	0.738	0.667
w/o Spatial Edges	0.962	0.921	0.975	0.963	0.906	0.767	0.743	0.651
w/o Revisit Edges	0.973	0.935	0.975	0.969	0.899	0.762	0.732	0.641
w/o Global Sampling	0.962	0.939	0.973	0.958	0.888	0.764	0.745	0.656

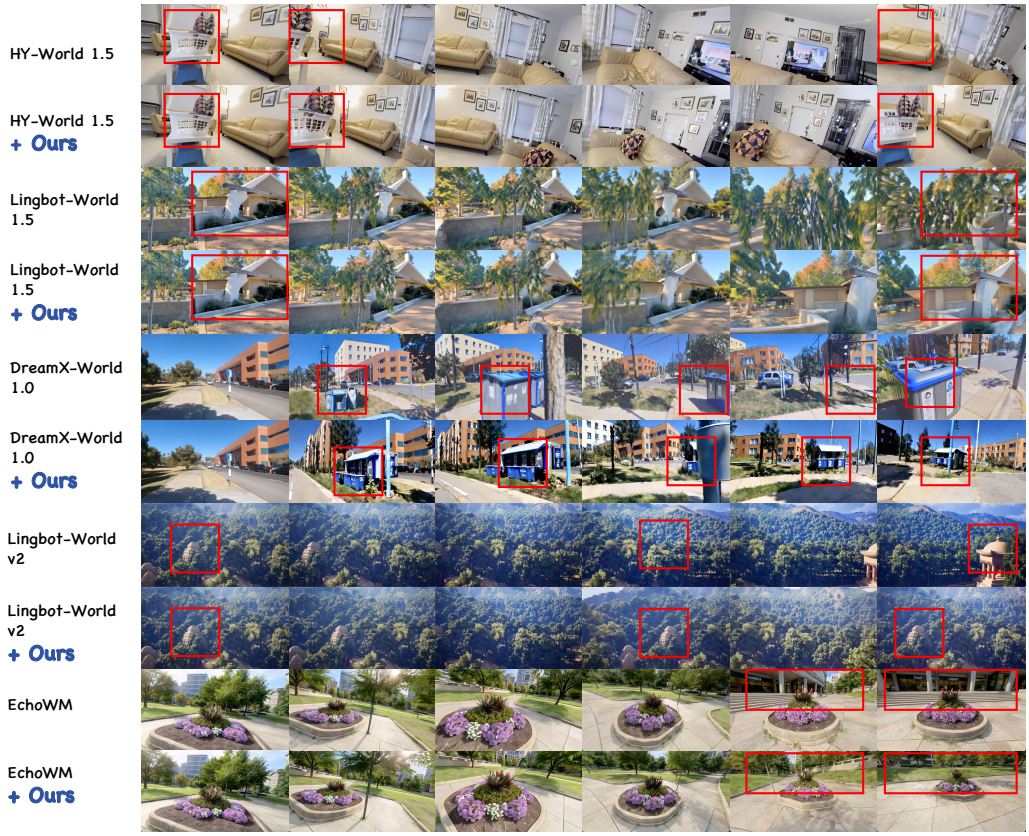
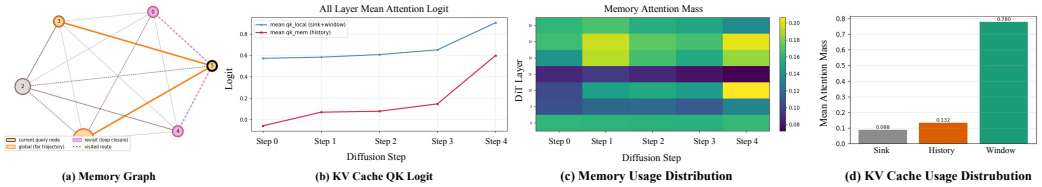


Figure 4: **Cross-backbone comparisons under camera revisits.** Each pair shows an original backbone (upper row) and the same backbone equipped with GraphMem (lower row) along the same trajectory. When the camera returns after a long temporal gap, GraphMem better preserves previously observed scene elements and layouts.

Qualitative Results. Figure 4 presents five paired comparisons under camera-revisit trajectories. For each pair, the original backbone and its GraphMem-augmented counterpart follow the same trajectory, allowing us to examine what happens when the camera returns to a region whose earlier views are no longer in the recent context window. The distinction is clearest at these returns: the original generators may produce plausible adjacent frames yet alter previously established furniture, architecture, objects, or background layouts when the scene reappears (red box in Figure 4). With GraphMem, these recognizable elements are more faithfully recovered while the camera continues to move. This pattern appears across five backbones, suggesting that access to relevant historical views improves scene recall across different generators and environments. Together, the paired results establish a training-free memory gain across different video generators.



(a) Memory Graph (b) KV Cache QK Logit (c) Memory Usage Distribution (d) KV Cache Usage Distrubution

Figure 6: **KV-cache utilization during inference.** (a) Vision Memory Graph and its temporal, spatial, and revisit relations. (b) Mean query-key logits for local context and retrieved history across denoising steps. (c) History attention mass distribution across DiT layers and denoising steps. (d) Attention allocation among caches. This shows that the history cache is used effectively.



Figure 5: **Ablation of memory organization and graph relations.** Rows compare the Queue, Bank, Graph, and variants without temporal (*w/o Temp*), spatial (*w/o Spa*), revisit (*w/o Revis*), and global sampling (*w/o Glo*) edges along the same trajectories. The full graph more consistently preserves scene structures and object appearance when previously observed regions reappear.

Ablation Study. We evaluate the contributions of graph organization, graph relations, and global sampling using the same backbone and evaluation protocol. Table 2 compares the full GraphMem configuration with a recent-context Queue, two bank-based baselines, and variants that remove individual graph components. GraphMem leads on all eight metrics. Relative to the Queue, its Long-Horizon, Memory, and Loop Closure scores increase from 0.733, 0.733, and 0.651 to 0.783, 0.775, and 0.692, respectively. Both Bank variants also trail the full graph, confirming that retaining history is less effective than organizing it for structured retrieval. **All graph relations contribute to these gains**. Removing temporal edges reduces the Long-Horizon and Memory scores to 0.764 and 0.738, indicating weaker trajectory continuity. Removing spatial edges decreases the Memory and Loop Closure scores to 0.743 and 0.651, reflecting poorer access to spatially relevant history. Revisit edges are particularly important: removing them yields the lowest Memory and Loop Closure scores of 0.732 and 0.641. Without global sampling, these scores decrease to 0.745 and 0.656, showing that local traversal alone cannot cover all relevant distant observations. Figure 5 illustrates the same trend: the full graph better preserves the industrial entrance and coastal boat across revisits, whereas the baselines and ablated variants exhibit larger changes in structure, placement, and appearance.

4.3 Analysis

How is the retrieved memory used? Figure 8 (a) and (d) show that the model does use graph-retrieved history: it receives 13.2% of the attention, with access concentrated in particular DiT layers and denoising steps. The Recent Cache still receives 78.0%, indicating that historical evidence complements the local context rather than replacing it. **Why native KV-cache prefill?** GraphMem uses the frozen DiT to produce historical keys and values in the backbone's native attention space. This gives retrieved frames a direct path to current queries without training, and Figure 8 (b) and (c) show that the history cache is effectively utilized. Further comparisons with alternative injection designs in Appendix 6.3 and latency analysis are provided in Appendix 6.1 and 6.2.

5 Conclusion

We presented **GraphMem**, a training-free visual memory graph framework that dynamically retrieves history and converts it into native history KV caches via Frozen-DiT prefill, improving long-horizon consistency across five backbones without changing any model parameters.

REFERENCES

Danze Chen, Zeqing Wang, Ziyue Lin, Xingyi Yang, and Yeying Jin. H3-world: Turning language understanding into world control, 2026a. URL https://arxiv.org/abs/2609.01560.

Zhifei Chen, Luozhou Wang, Guibao Shen, Dongyu Yan, Shuai Yang, Tianshuo Xu, Yihua Du, Wei Wang, Tianyi Gui, Lianghua Huang, et al. Reworld: An interactive world model with long-horizon memory. *arXiv preprint arXiv:2608.23565*, 2026b.

Zelin Gao, Qiuyu Wang, Jiapeng Zhu, Jingye Chen, Zichen Liu, Qingyan Bai, Jiahao Wang, Yufeng Yuan, Hanlin Wang, Yichong Lu, Ka Leong Cheng, Haojie Zhang, Jian Gao, Tianrui Feng, Yuzheng Liu, Yao Yao, Yinghao Xu, Xing Zhu, Yujun Shen, and Hao Ouyang. Infinite worlds with versatile interactions. *arXiv preprint arXiv:2607.07534*, 2026.

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

Yicong Hong, Yiqun Mei, Chongjian Ge, Yiran Xu, Yang Zhou, Sai Bi, Yannick Hold-Geoffroy, Mike Roberts, Matthew Fisher, Eli Shechtman, et al. Relic: Interactive video world model with long-horizon memory. *arXiv preprint arXiv:2512.04040*, 2025.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025a.

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

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, et al. Vbench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b.

Team HunyuanWorld. Hy-world 1.5: A systematic framework for interactive world modeling with real-time latency and geometric consistency. *arXiv preprint*, 2025.

Sihui Ji, Xi Chen, Shuai Yang, Xin Tao, Pengfei Wan, and Hengshuang Zhao. Memflow: Flowing adaptive memory for consistent and efficient long video narratives. *arXiv preprint arXiv:2512.14699*, 2025.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Wayne King, Zeyue Xue, Yuxuan Bian, Jie Huang, Haoran Li, Yaowei Li, Yaofeng Su, Yuming Li, Haoyu Wang, Shiyi Zhang, et al. Echo-memory: A controlled study of memory in action world models. *arXiv preprint arXiv:2606.09803*, 2026.

Zijun Lin, Zeqing Wang, Cheston Tan, Bihan Wen, and Yeying Jin. Stateplay: State-aware game world models for mechanics-consistent generation, 2026. URL https://arxiv.org/abs/2607.26754.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22160–22169. IEEE, 2024.

Dongyang Liu, Peng Gao, David Liu, Ruoyi Du, Zhen Li, Qilong Wu, Xin Jin, Sihan Cao, Shifeng Zhang, Hongsheng Li, and Steven Hoi. Decoupled dmd: Cfg augmentation as the spear, distribution matching as the shield. *arXiv preprint arXiv:2511.22677*, 2025.

Hongyu Liu, Chun Wang, Feng Gao, Xuanhua He, Yue Ma, Ziyu Wan, Yong Zhang, Xiaoming Wei, and Qifeng Chen. Opsd-v: On-policy self-distillation for post-training few-step autoregressive video generators. *arXiv preprint arXiv:2607.08766*, 2026a.

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

Yunhong Lu, Yanhong Zeng, Haobo Li, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Jiapeng Zhu, Hengyuan Cao, Zhipeng Zhang, Xing Zhu, et al. Reward forcing: Efficient streaming video generation with rewarded distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 34385–34397, 2026.

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

Yue Ma, Kunyu Feng, Zhongyuan Hu, Xinyu Wang, Yucheng Wang, Mingzhe Zheng, Xuanhua He, Chenyang Zhu, Hongyu Liu, Yingqing He, et al. Controllable video generation: A survey. *arXiv preprint arXiv:2507.16869*, 2025.

Elia Peruzzo, Dejia Xu, Xingqian Xu, Humphrey Shi, and Nicu Sebe. Ragme: Retrieval-augmented video generation for enhanced motion realism. In *Proceedings of the 2025 International Conference on Multimedia Retrieval*, pp. 1081–1090, 2025.

Parikshit Ram and Kaushik Sinha. Revisiting kd-tree for nearest neighbor search. In *Proceedings of the 25th acm sigkdd international conference on knowledge discovery & data mining*, pp. 1378–1388, 2019.

AlayaWorld Team, Kaipeng Zhang, Chuanhao Li, Yifan Zhan, Yongtao Ge, Yuanyang Yin, Jiaming Tan, Kang He, Liaoyuan Fan, Ruicong Liu, et al. Alayaworld: Long-horizon and playable video world generation. *arXiv preprint arXiv:2607.06291*, 2026a.

DreamX Team, Yancheng Bai, Rui Chen, Xiangxiang Chu, Rujing Dang, Hao Dou, Bingjie Gao, Qiwen Gu, Siyu Hong, Jiachen Lei, et al. Dreamx-world 1.0: A general-purpose interactive world model. *arXiv preprint arXiv:2606.16993*, 2026b.

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*, 2026c.

Zizhao Tong, Yeying Jin, Hongfeng Lai, Zeqing Wang, Zhaohu Xing, Kexu Cheng, Haoran Xu, Zhao Pu, Shangwen Zhu, Ruili Feng, Jian Zhao, Yan Zhang, Hao Tang, and Ling Shao. Scope: Simulating cross-game operations in playable environments for fps world models, 2026. URL https://arxiv.org/abs/2605.23345.

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

Yifan Wang and Tong He. Warp-as-history: Generalizable camera-controlled video generation from one training video. *arXiv preprint arXiv:2605.15182*, 2026.

Zeqing Wang, Danze Chen, Zhaohu Xing, Zizhao Tong, Yinhan Zhang, Xingyi Yang, and Yeying Jin. Reactivegwm: Steering npc in reactive game world models, 2026b. URL https://arxiv.org/abs/2605.15256.

Zile Wang, Zexiang Liu, Jiaxing Li, Kaichen Huang, Baixin Xu, Fei Kang, Mengyin An, Peiyu Wang, Biao Jiang, Yichen Wei, et al. Matrix-game 3.0: Real-time and streaming interactive world model with long-horizon memory. *arXiv preprint arXiv:2604.08995*, 2026c.

Zun Wang, Han Lin, Jaehong Yoon, Jaemin Cho, Yue Zhang, and Mohit Bansal. Anchorweave: World-consistent video generation with retrieved local spatial memories. *arXiv preprint arXiv:2602.14941*, 2026d.

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.

Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory. *Advances in Neural Information Processing Systems*, 38:49371–49393, 2026b.

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*, 2026c.

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.

Jiacong Xu, Hanwen Jiang, Zhixin Shu, Kalyan Sunkavalli, Vishal M Patel, and Yiqun Mei. Wonder: Video world model done better. *arXiv preprint arXiv:2607.26037*, 2026.

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

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

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. *arXiv preprint arXiv:2311.18828*, 2023.

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

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*, SA Conference Papers '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400721373. doi: 10.1145/3757377.3763833. URL https://doi.org/10.1145/3757377.3763833.

Jiwen Yu, Jianxiong Gao, Jianhong Bai, Yiran Qin, Kaiyi Huang, Quande Liu, Xintao Wang, Pengfei Wan, Kun Gai, and Xihui Liu. Memlearner: Learning to query context memory for video world models. *arXiv preprint arXiv:2606.31734*, 2026a.

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*, 2026b.

Lvmin Zhang, Shengqu Cai, Muyang Li, Gordon Wetzstein, and Maneesh Agrawala. Frame context packing and drift prevention in next-frame-prediction video diffusion models. *Advances in Neural Information Processing Systems*, 38:30546–30566, 2026a.

Songchun Zhang, Yaowei Li, Junhao Zhuang, Weiyang Jin, Haoyu Wang, Xin Lu, Yilang Sun, Shiyi Zhang, Haoran Li, Xiaoxiao Ma, et al. Echowm: Open and enterable omnimodal world models. *arXiv preprint arXiv:2608.23189*, 2026b.

Yisu Zhang, Chenjie Cao, Tengfei Wang, Xuhui Zuo, Junta Wu, Jianke Zhu, and Chunchao Guo. Worldstereo: Bridging camera-guided video generation and scene reconstruction via 3d geometric memories. *arXiv preprint arXiv:2603.02049*, 2026c.

Min Zhao, Hongzhou Zhu, Bokai Yan, Zihan Zhou, Yimin Chen, Wenqiang Sun, Kaiwen Zheng, Guande He, Xiao Yang, Chongxuan Li, et al. minwm: A full-stack open-source framework for real-time interactive video world models. *arXiv preprint arXiv:2605.30263*, 2026a.

Min Zhao, Hongzhou Zhu, Kaiwen Zheng, Zihan Zhou, Bokai Yan, Xinyuan Li, Xiao Yang, Chongxuan Li, and Jun Zhu. Causal forcing++: Scalable few-step autoregressive diffusion distillation for real-time interactive video generation. *arXiv preprint arXiv:2605.15141*, 2026b.

Chenhui Zhu, Yilu Wu, Shuai Wang, Gangshan Wu, and Limin Wang. Motionrag: Motion retrieval-augmented image-to-video generation. *Advances in Neural Information Processing Systems*, 38: 173392–173415, 2026a.

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*, 2026b.

Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. *arXiv preprint arXiv:2602.02214*, 2026c.

Shangwen Zhu, Qianyu Peng, Zhao Pu, Zhilei Shu, Xiangrui Ke, Zhaohu Xing, Zizhao Tong, Zeqing Wang, Xinyu Cui, Zian Zheng, Huangji Wang, Jian Zhao, Yeying Jin, Fan Cheng, and Ruili Feng. Incantation: Natural language as the action interface for multi-entity video world models, 2026d. URL https://arxiv.org/abs/2605.18601.

6 Appendix

6.1 Memory Retention over Temporal Gaps

We first examine whether GraphMem's benefit persists as the recalled observation becomes more distant. We group pose-matched chunk pairs by their temporal gap and measure their visual similarity. As shown in Fig. 7(b), all methods perform similarly over short gaps, where the relevant scene often remains accessible through recent context. Their behavior diverges as the gap increases. The Base model exhibits the fastest decay, while Queue and Bank provide only limited improvements. At the longest evaluated gap, their similarities decrease by 25%, 22%, and 23%, respectively, compared with 21% for GraphMem.

This difference reflects how each method organizes historical evidence. Queue retrieval remains biased toward temporal recency and eventually loses the required view. Bank preserves more history, but its flat organization makes relevant observations compete with redundant or unrelated entries. GraphMem instead groups observations by scene and indexes them through visual, spatial, and trajectory relations. A revisited region can therefore reactivate its corresponding history regardless of when it was observed. The widening margin at longer gaps confirms that GraphMem primarily improves long-range scene recall rather than short-term temporal smoothness.

6.2 Retrieval Latency

Long-term memory introduces additional costs from retrieval, history prefill, and memory maintenance. Figure 7(a) therefore reports end-to-end latency for the complete rollout. The Base model requires 65.6 seconds, while Queue and Bank increase the latency to 125.4 and 105.2 seconds, corresponding to overheads of 91% and 60%. GraphMem reduces the total latency to 100.7 seconds, limiting the overhead to 53%, the lowest among the evaluated memory mechanisms.

The efficiency gain again follows from structured memory access. By consolidating related observations into scene nodes, GraphMem avoids repeatedly activating redundant history. Its bounded retrieval set also reduces both history-prefill and denoising costs compared with Queue and Bank. Strided cache refresh further amortizes prefill, while graph construction and updates are overlapped with autoregressive generation, leaving only 4.1 seconds of exposed update latency. GraphMem therefore does not eliminate the cost of long-term memory, but provides a better memory–latency trade-off: it preserves scene consistency over longer gaps while introducing less end-to-end overhead than sequential or unstructured memory.

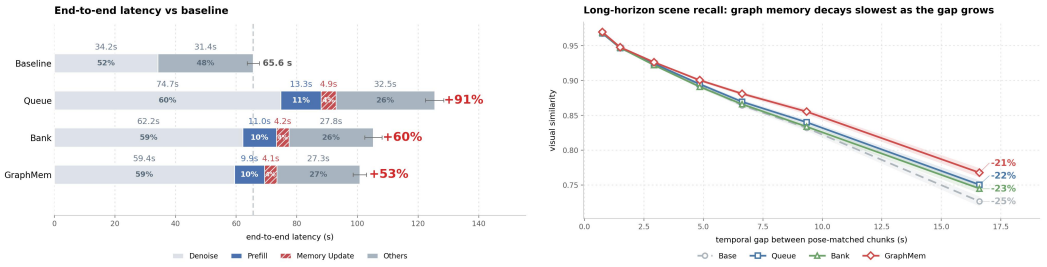


Figure 7: **Long-horizon memory retention and inference efficiency.** (a) End-to-end latency and its breakdown for the Base model and different memory organizations. GraphMem incurs the lowest overhead among the evaluated memory mechanisms. (b) Visual similarity between pose-matched chunks as their temporal gap increases. GraphMem exhibits the slowest memory decay, with its advantage becoming more pronounced over longer gaps.

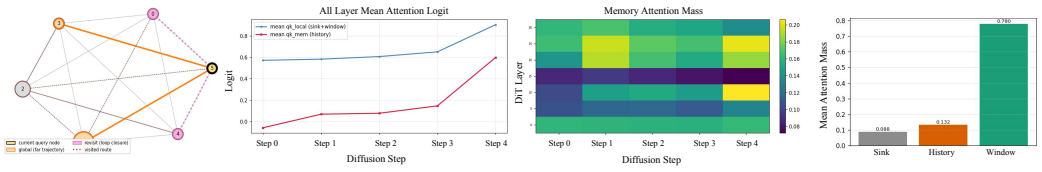


Figure 8: **KV-cache utilization during a representative camera revisit.** (a) The Vision Memory Graph associated with the current query. (b) Mean query-key logits for local context and graph-retrieved history across denoising steps. (c) History attention mass across DiT layers and denoising steps. (d) Aggregate attention allocation among the Sink, History, and Window caches. Retrieved history is accessed selectively, while the local window remains the dominant context source.

6.3 Memory Utilization and Native KV-Cache Prefill

Our analysis yields two main findings. First, the frozen backbone does not ignore graph-retrieved history. It accesses historical evidence selectively across denoising steps and DiT layers, while retaining the local window as its primary context. Second, this behavior depends on constructing history in the backbone's native KV space. An independently projected memory exhibits a large query-key logit mismatch and receives little attention, whereas native prefill produces well-calibrated historical states that participate consistently in the original self-attention operation. Together, these findings explain why GraphMem requires both components introduced in the main method: graph retrieval determines *what* should be recalled, while native KV prefill determines whether the frozen generator can effectively *use* it.

6.3.1 How Does the Model Use Retrieved Memory?

Main finding. Graph-retrieved history complements rather than replaces local context. On the representative revisit trajectory in Fig. 8, the Window Cache receives 78.0% of the total attention and remains responsible for local motion and appearance continuity. The History Cache nevertheless receives a meaningful 13.2%, with its use concentrated in particular layers and denoising steps. This selective pattern shows that the pretrained backbone can access recalled observations when they become relevant without being forced to attend to them uniformly.

Analysis protocol. We record attention statistics along a long-horizon trajectory containing a camera revisit. The keys available to each current-chunk query are partitioned into

$$\mathcal{K} = \mathcal{K}_{\text{sink}} \cup \mathcal{K}_{\text{hist}} \cup \mathcal{K}_{\text{win}}, \tag{9}$$

where the three subsets contain persistent sink keys, graph-retrieved history keys, and keys from the current and recent chunks, respectively. Let $\mathcal{Q}$ denote the current-chunk queries. At DiT layer l,

denoising step τ, and attention head h, the scaled query-key logit is

$$S^{l,\tau}_{h,q,k} = \frac{\left(\mathbf{Q}^{l,\tau}_{h,q}\right)^{\top} \mathbf{K}^{l,\tau}_{h,k}}{\sqrt{d_h}}, \tag{10}$$

where d_h is the feature dimension of one attention head. For a cache region r, its mean query-key compatibility is

$$L^{l,\tau}_{r} = \frac{1}{N_h|\mathcal{Q}||\mathcal{K}_r|} \sum_{h=1}^{N_h} \sum_{q\in\mathcal{Q}} \sum_{k\in\mathcal{K}_r} S^{l,\tau}_{h,q,k}, \tag{11}$$

where N_h is the number of attention heads. We further define the attention probability

$$P^{l,\tau}_{h,q,k} = \frac{\exp\left(S^{l,\tau}_{h,q,k}\right)}{\sum_{k'\in\mathcal{K}} \exp\left(S^{l,\tau}_{h,q,k'}\right)}, \tag{12}$$

and the attention mass allocated to region r as

$$A^{l,\tau}_{r} = \frac{1}{N_h|\mathcal{Q}|} \sum_{h=1}^{N_h} \sum_{q\in\mathcal{Q}} \sum_{k\in\mathcal{K}_r} P^{l,\tau}_{h,q,k}. \tag{13}$$

The first statistic measures representational compatibility before softmax, whereas the second measures the fraction of attention actually assigned to each cache region.

Progressive activation during denoising. Figure 8(a) shows the graph state for the analyzed revisit. The current query recovers temporally distant observations through the graph, which are then converted into the active History Cache. At the beginning of denoising, the mean history logit is slightly below zero, compared with approximately 0.57 for local context. As the current latent develops coherent visual structure, the history logit rises to approximately 0.59 (Fig. 8b). This trend does not alone determine the final attention allocation, since all keys compete within the same softmax. It nevertheless shows that retrieved observations become increasingly compatible with the evolving current representation rather than remaining uniformly irrelevant.

Layer- and timestep-selective access. The attention map in Fig. 8(c) provides more direct evidence of memory use. Historical attention is concentrated in selected DiT layers and denoising steps, while other layers continue to rely predominantly on local context. This non-uniform pattern is important: the History Cache is not imposed as a global conditioning signal. Instead, each layer decides whether to use the recalled evidence through its native query-key compatibility.

The aggregate allocation in Fig. 8(d) supports the same interpretation. The Window Cache receives 78.0% of the attention, the History Cache receives 13.2%, and the Sink Cache receives the remaining 8.8%. These values depend partly on the number of tokens in each region and should not be interpreted as token-normalized importance. However, together with the layer-wise analysis, they establish that retrieved history participates meaningfully in generation without overwhelming the recent context. The corresponding improvements in long-horizon memory and loop closure are reported in Sec. 4.2, while the graph-relation ablations in Sec. 4.2 verify the functional contribution of the retrieved evidence.

6.3.2 Why Native KV-Cache Prefill?

Main finding. Retrieved observations are useful only if their attention states are compatible with those of the frozen backbone. Native prefill achieves this compatibility by processing historical keyframes through the pretrained DiT itself. The resulting keys and values share the same layer-wise projections and feature distribution as the backbone's original cache. By contrast, a separately learned memory pathway must either reproduce this distribution through independent projections or overcome a residual gate that can suppress the new pathway.

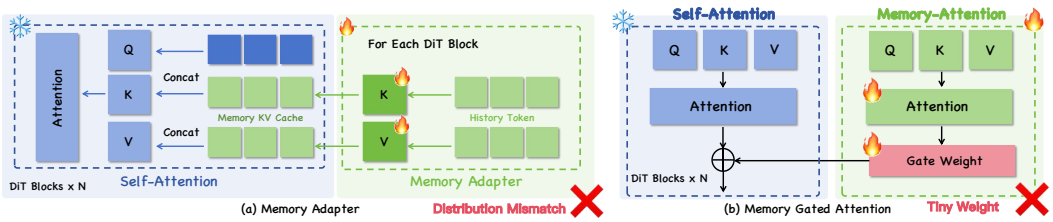


Figure 9: **Limitations of alternative memory-integration mechanisms.** (a) A memory adapter independently projects historical tokens before concatenating them with the native self-attention cache, creating a potential mismatch between the two KV spaces. (b) A separate memory-attention branch is controlled by a learned residual gate, which can suppress the memory pathway. GraphMem instead constructs history through the frozen DiT and inserts it directly into the native KV space. Fire and snowflake symbols denote trainable and frozen modules, respectively.

Why alternative interfaces fail. The memory adapter in Fig. 9(a) independently projects historical tokens into keys and values before concatenating them with the native cache. Historical and local keys must then compete inside the same softmax despite being produced by different projection functions. Smaller memory logits cause history to be ignored, whereas excessively large logits disturb the pretrained attention distribution. The adapter must therefore learn a properly calibrated KV space across all DiT layers and denoising states.

Let h_{self} be the output of the self-attention branch and h_{mem} that of the memory-attention branch. The gated design in Fig. 9(b) avoids mixing the two KV spaces directly, but creates a different optimization shortcut:

$$\mathbf{h}' = \mathbf{h}_{\text{self}} + \mathbf{g} \odot \mathbf{h}_{\text{mem}}, \tag{14}$$

where $\mathbf{g}$ controls the contribution of the memory branch. Because the pretrained self-attention pathway already provides a stable local-context solution, reducing $\mathbf{g}$ is an easy way to avoid perturbing the original denoising behavior. In our preliminary experiments, the gate consequently approaches a near-zero value, leaving the memory branch largely inactive.

Native prefill removes both failure modes. Retrieved keyframes are encoded at a timestep-aligned noise level and passed through the frozen DiT. Each layer therefore constructs its historical keys and values using the same projections that produce the native cache. The historical states enter the original self-attention operation directly, without an independently calibrated KV space or a suppressible residual gate. Native prefill does not force the backbone to use every retrieved frame; it makes the evidence available under the same compatibility test applied to local context.

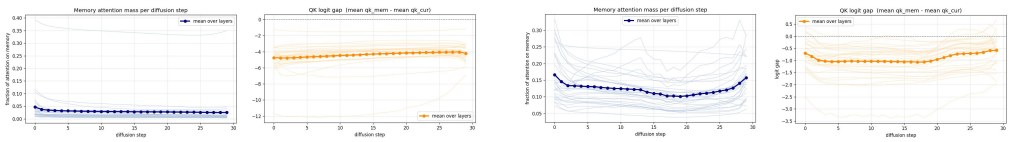
Quantitative evidence of KV calibration. We verify this explanation by comparing the memory adapter and native prefill using identical retrieved observations and the same generation trajectory. Figure 10 reports two complementary statistics: the history attention mass and the QK-logit gap between historical and current-context keys. We define the latter as

$$\Delta^{\tau}_{\text{QK}} = \frac{1}{L} \sum_{l=1}^{L} \left(L^{l,\tau}_{\text{hist}} - L^{l,\tau}_{\text{cur}} \right), \tag{15}$$

where L is the number of DiT layers and L_{cur} is current chunk noise latent. A strongly negative value indicates that historical keys are disadvantaged in the shared softmax competition.

The adapter exhibits a persistent QK-logit gap of approximately -4 to -5. Its layer-averaged history attention consequently remains mostly between 2% and 5%, showing that the independently projected states are present in the cache but rarely selected. Native prefill reduces the gap to approximately -0.5 to -1.1 and sustains a history attention mass of roughly 10% to 17%. The gap remains mildly negative rather than becoming strongly positive, indicating that historical keys are made competitive without dominating the local context.

These results establish the connection between KV calibration and memory use. Graph retrieval alone cannot guarantee that a frozen DiT will exploit the retrieved observations. By reconstructing history through the backbone's native projections, GraphMem converts external pixel-space evidence into attention states that the model can selectively access. This completes the memory loop



(a) Memory Adapter (b) Native KV Cache Prefill

Figure 10: **KV calibration and memory utilization.** For the memory adapter and native prefill, we report history attention mass and the QK-logit gap at each denoising step. Light curves represent individual DiT layers, and dark curves show the layer-wise mean. Native prefill substantially narrows the logit gap and maintains greater attention to retrieved history.

introduced in the main method: the graph makes distant evidence addressable, and native KV prefill makes that evidence usable.

6.4 Additional Experiment Details

6.4.1 Additional Experimental Setup

All experiments are conducted on an NVIDIA H200 GPU with an inference batch size of 1. We use three metrics to evaluate visual memory over long temporal gaps. The *Long-Horizon Score* measures whether the end of a video remains consistent with its beginning. The *Loop Closure Score* measures whether a previously visited scene is recovered when the camera returns. The *Memory Score* combines these signals into a single score. These metrics are computed separately from VBench and are not included in its overall score.

We use DINO features to measure subject similarity, and CLIP features to measure background similarity. For $e \in \{\text{sub}, \text{bg}\}$, the similarity between observations i and j is

$$s_e(i,j) = \max\left(0, \frac{\mathbf{z}_i^{e\top}\mathbf{z}_j^e}{\|\mathbf{z}_i^e\|_2\|\mathbf{z}_j^e\|_2}\right), \tag{16}$$

where $\mathbf{z}_i^e$ is the corresponding DINO or CLIP feature. Each score is first computed for individual videos and then averaged over the evaluation set.

Long-Horizon Score. We divide each video into its original autoregressive chunks. For each chunk, we sample up to four frames and average their features to obtain a chunk representation. Let $\mathcal{H}$ and $\mathcal{T}$ denote the chunks in the first and last quarters of the video. The Long-Horizon Score compares every head chunk with every tail chunk:

$$\text{LongHorizon} = \frac{1}{2}\sum_{e\in\text{sub,bg}} \frac{1}{|\mathcal{H}||\mathcal{T}|}\sum_{i\in\mathcal{H}}\sum_{j\in\mathcal{T}} s_e(i,j). \tag{17}$$

We exclude the middle half of the video so that intermediate frames do not dilute the long-term consistency signal.

Loop Closure Score. We uniformly sample 16 frames from the video and align their camera poses to the same timestamps. Let $\mathcal{I}_{16}$ denote the sampled frame indices, t_i the original frame index, $\mathbf{c}_i$ the camera center, and N the total number of frames. We first estimate the spatial scale of the sampled trajectory:

$$S = \left\| \max_{i\in\mathcal{I}_{16}} \mathbf{c}_i - \min_{i\in\mathcal{I}_{16}} \mathbf{c}_i \right\|_2, \tag{18}$$

where the maximum and minimum are computed independently for each spatial coordinate. A pair is considered a revisit when the frames are far apart in time but close in camera position:

$$\mathcal{R} = \left\{ (i,j) \;\middle|\; i,j \in \mathcal{I}_{16}, \quad t_j > t_i, \quad \frac{t_j - t_i}{N-1} \geq 0.5, \quad \|\mathbf{c}_j - \mathbf{c}_i\|_2 \leq 0.15S \right\}. \tag{19}$$

The Loop Closure Score averages the subject and background similarities over all valid revisit pairs:

$$\text{LoopClosure} = \frac{1}{2|\mathcal{R}|} \sum_{e \in \text{sub,bg}} \sum_{(i,j) \in \mathcal{R}} s_e(i,j), \qquad |\mathcal{R}| > 0. \tag{20}$$

Videos without a valid revisit pair are excluded from the dataset-level Loop Closure average.

Memory Score. We also measure similarity at the largest temporal gaps. Let $\mathcal{B}_{\text{far}}$ denote the evaluated frame pairs whose normalized temporal gaps fall within $[0.75, 1.0]$. Their average similarity is

$$\text{FarGap} = \frac{1}{2|\mathcal{B} * \text{far}|} \sum_{e \in \text{sub,bg}} \sum_{(i,j) \in \mathcal{B}_{\text{far}}} s_e(i,j). \tag{21}$$

The final Memory Score combines endpoint consistency, far-gap retention, and revisit consistency:

$$\text{Memory} = 0.4 \times \text{LongHorizon} + 0.3 \times \text{FarGap} + 0.3 \times \text{LoopClosure}. \tag{22}$$

If camera poses are unavailable, the Loop Closure term is omitted, and the remaining weights are renormalized.

6.4.2 Additional Qualitative Results

Figure 11 provides additional comparisons on two long-horizon generation scenarios: an East Asian architectural environment with tiled roofs, pagodas, and mountains, and a European urban environment with dense building facades and narrow streets. Frames are arranged chronologically from left to right, allowing us to examine both viewpoint progression and scene consistency over the rollout.

The compared methods exhibit different failure modes. HY-WorldPlay 1.5 and MatrixGames 3.0 largely preserve the initial appearance but produce limited scene evolution, suggesting insufficient response to the camera trajectory. minWM generates stronger motion but gradually develops severe texture artifacts and geometric corruption. SANA-World, WorldKV, and LingBot-World 1.5 produce more substantial viewpoint changes, yet progressively alter architectural styles, structural details, and background layouts. These results indicate that plausible individual frames and visible camera motion do not necessarily produce a persistent visual world.

GraphMem achieves a better balance between scene evolution and long-term consistency. In the East Asian scene, the camera advances while the roof geometry, architectural style, and mountainous background remain coherent. In the European scene, the sequence transitions from the initial facade into the surrounding street while preserving consistent materials, windows, columns, and architectural identity. The results further demonstrate that graph-retrieved history reduces structural and appearance drift without suppressing camera-induced scene evolution.



Figure 11: **Qualitative comparison with baseline methods.** Frames are sampled chronologically from two long-horizon trajectories. Existing methods exhibit limited scene evolution, structural distortion, texture corruption, or drift in architectural identity. GraphMem preserves coherent camera progression while maintaining scene appearance, geometry, and visual quality throughout the rollout.

6.5 AI Use Statement

We used AI-assisted tools to improve the manuscript's grammar and clarity. The authors reviewed the resulting text and take full responsibility for the paper's technical claims, citations, and conclusions.

6.6 Ethics and Reproducibility Statement

GraphMem augments pretrained video generators at inference time and requires no additional model training. Like the underlying generators, it can produce realistic synthetic scenes that may be mistaken for genuine footage; this work does not address that potential misuse. All results and qualitative examples are reproducible.