# W$^4$MEM: RELEVANCE, RELIABILITY, AND TIMING FOR LONG-HORIZON VIDEO WORLD MEMORY

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Video world models can generate realistic short sequences. However, they often lose scene identity as observations leave the local context and generation errors accumulate. We argue that effective long-horizon memory is not only a storage problem, since its utility depends on which observations are retrieved, how reliable the memory is, and when it is read during denoising. We introduce W$^4$Mem, a memory framework that collectively addresses these three factors through Role-factorized Retrieval, Diffused Memory Conditioning, and Denoising-Stage Routing. W$^4$Mem retrieves stable anchors, target-view revisits, and complementary views, and combines them with role and target-relative geometry. During training, our Diffused Memory Conditioning trains the model to use memories with query-dependent fidelity without expensive autoregressive rollouts. Denoising-Stage Routing allocates memory roles across denoising stages, reducing unnecessary computation. With the same backbone and memory budget as the baseline, our W$^4$Mem reduces MSE by 36.1%, LPIPS by 35.6%, and FID by 82.0%, while improving PSNR by 3.68 dB over Minecraft 500-frame rollouts. It also reduces inference time by 23.5% and peak memory of the GPU by 36.9%. On 300-frame forward-and-return RealEstate10K trajectories, W$^4$Mem also improves long-horizon scene consistency. Our results show that long-horizon world modeling benefits from jointly controlling memory relevance, reliability, and timing. Code and models will be released.

## 1 INTRODUCTION

Video world models aim to turn generative video models into interactive simulators. Conditioned on previous observations and control signals, they predict how the world should appear in the next time step (Bruce et al., 2024; Alonso et al., 2024; Valevski et al., 2025; Decart et al., 2024). Recent diffusion formulations make it possible to roll such models forward beyond a fixed prediction horizon (Chen et al., 2024), but generating a longer sequence does not necessarily preserve the world it depicts. When the camera returns to a visited location, the model should recover the established scene rather than generate a plausible replacement. Long-horizon world modeling, therefore, requires more than realistic successive frames. It requires scene appearance and geometry to remain consistent with observations made much earlier in the rollout.

Maintaining scene identity is difficult for two related reasons. First, the local context is limited, so observations that establish a scene's identity eventually leave the conditioning window, even though they may be needed again later (Xiao et al., 2025; Wu et al., 2025; Wei et al., 2026). Second, the model feeds its own predictions back into the context. Small errors in generated frames carry over to later predictions and build up over time (Chen et al., 2024; 2025; Huang et al., 2026). Meanwhile, earlier observations leave the context window, so the model has to rely more on its own imperfect predictions. Keeping older observations available can help limit this drift, but they need to be relevant to the current view and reliable enough to guide generation.

Recent memory-augmented methods use retrieved frames or persistent spatial representations as memory to retain information beyond the local window (Xiao et al., 2025; Wu et al., 2025; Hong et al., 2025; Yang et al., 2026b). However, access to a longer history does not by itself tell the model how to use a limited set of memory frames. An early observation gives the model a reference for scene identity. Revisits help recover content in the current view, whereas other viewpoints supply



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

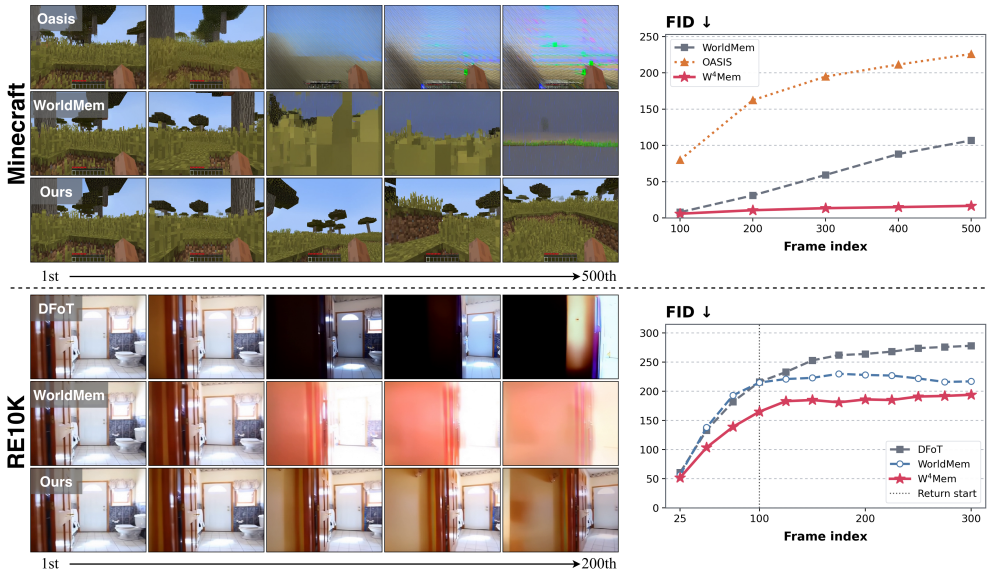


Figure 1: **W$^4$Mem preserves scene identity over long rollouts and revisiting.** In Minecraft (top), Oasis (Decart et al., 2024) and WorldMem (Xiao et al., 2025) tend to accumulate artifacts or lose the structure of previously seen scenes and objects, while our W$^4$Mem maintains consistency with earlier observations throughout the 500-frame rollout. On RealEstate10K (bottom), when the camera returns to previous viewpoints, W$^4$Mem recovers the scene's appearance and layout more accurately, whereas DFoT (Song et al., 2025) and WorldMem exhibit increasing drift. The FID curves on the right show that the quality gap expands as the rollout continues.

complementary geometric evidence. Selecting useful frames is not enough, a memory-based model trained on clean observations must later work with generated frames that may contain errors. A frame's usefulness can also change over the course of denoising. We treat these as three points, namely relevance, reliability, and timing. We ask *which frames to retrieve*, *what fidelity to use during training*, and *when to read them*. To address these limitations, we introduce W$^4$Mem to control how past frames are selected and used as memory. For relevance, we separate the memory into anchors, revisits, and complementary views. Anchors from the input observed prefix provide a stable reference for scene appearance and layout. For the target, *revisits* offer the best coverage of view, while *complementary views* supply more observations from other viewpoints that can reveal details the revisits miss. All memories are combined by a unified reader with role and geometry information. Learned role embeddings distinguish the memory roles, while target-relative camera geometry helps align each frame with the target view. For reliability, we propose diffused memory conditioning to enable the model to learn to use imperfectly generated memory. During training, we corrupt active memory latents through the model's forward diffusion process, with noise levels conditioned on the target query's denoising condition. The model learns to use memory of varying fidelity without autoregressive training rollouts. For timing, the denoising-stage routing controls when each role is read. It follows a coarse-to-fine prior that anchors are used at high noise, complementary views at low noise, and revisits throughout denoising. Training and inference use the same routing threshold. For each target set, we make a single routing decision based on the mean target timestep. At inference, the active memory roles depend on the current denoising timestep. This lets the model use each role at the intended stage and saves computation at inference by skipping inactive memory streams. These three parts correspond to which memory is relevant, what fidelity it has, and when it should be used.

Experiments on Minecraft and RealEstate10K, examples in Figure 1, demonstrate improved generation quality and scene consistency over long-horizon rollouts. With the same backbone and memory budget as baseline (Xiao et al., 2025), our method reduces MSE by 36.1%, LPIPS by 35.6%, FID by 82.0%, and improves PSNR by 3.68 dB on the Minecraft benchmark. W$^4$Mem also reduces 23.5% inference time and 36.9% peak GPU memory use. Our main contributions are as follows:

- **Memory relevance.** We show that historical observations play distinct functional roles: stable scene anchors, target-view revisits, and complementary geometric evidence. We thus introduce role-factorized retrieval and a unified role- and geometry-aware reader preserving these distinctions.

- **Memory reliability.** We identify a train–inference mismatch caused by clean training memories versus imperfect generated histories. We propose query-conditioned diffusion corruption to expose the denoiser to a controllable range of memory fidelity.
- **Memory timing.** We show that the utility of different memory roles varies across denoising stages, and exploit this structure through stage-aware routing that simultaneously reduces memory-attention computation and improves long-horizon quality.
- **Long-horizon evaluation.** We demonstrate improved scene consistency over 500-frame Minecraft rollouts and 300-frame forward-and-return RealEstate10K trajectories under matched backbone and memory budgets.

## 2 Related Work

**Long-horizon video world models.** Interactive video generators have been developed through latent-action models, diffusion world models, and real-time neural game engines (Bruce et al., 2024; Alonso et al., 2024; Valevski et al., 2025; Decart et al., 2024). Diffusion Forcing samples a noise level independently for each sequence token during causal training. The model can then generate sequences longer than local context in training (Chen et al., 2024). Other recent methods change the generator or its training distribution to make long rollouts more stable. LongLive (Yang et al., 2026a) trains model with a teacher supervising each short clip on its own long-horizon rollouts. LIVE (Huang et al., 2026) enforces cycle consistency by reconstructing the starting frames from reversed trajectory. RELIC (Hong et al., 2025) uses memory-efficient self-forcing (Huang et al., 2025) to distill a long-horizon generation teacher model into a causal student model. AlayaWorld (Team et al., 2026) use autoregressive distillation with corrupted histories and rollout residuals. These methods close the train–inference gap by generating real rollouts during training. Our method instead corrupts ground-truth memory with the model's own forward-diffusion process, covering a similar fidelity range without the computation cost of actual rollouts.

**Memory for video world models.** Memory-based video world models differ in the representation of memory they maintain from past observations. Spatial-based methods produce persistent 3D structures or geometric caches (Wu et al., 2025; Wang et al., 2026), while frame-based methods (Yu et al., 2025) retain visual detail directly. WorldMem (Xiao et al., 2025) retrieves past frames based on view overlap and temporal distance. DecMem (Yang et al., 2026b) divides latent memory into sparse global and anchored local components. MemLearner (Yu et al., 2026) introduces learnable query tokens to extract relevant information from historical context. GIM-World (Wei et al., 2026) compresses variable-length history into a fixed number of geometry-aware memory tokens. Matrix-Game 3.5 (Qian et al., 2026) maintains a geometry-aware 3D patch memory, while ReWorld (Chen et al., 2026) uses a bounded KV cache together with a pose-indexed memory bank. These methods improve how long memory are represented or queried. W$^4$Mem keeps past frames as memory and instead focuses on how a fixed memory budget is used during diffusion generation. We separate memories by their role for the current target, vary memory fidelity during training, and control when each role is read during denoising. Our focus is on relevance, reliability and timing rather than introducing another persistent or compressed memory representation.

## 3 Method

Given a target query frame for denoising and its causal history, W$^4$Mem constructs a bounded frame memory and injects it into every DiT block through a shared target-to-memory reader, as shown in Figure 2. We organize the method around three factors that affect memory use under a fixed budget, namely relevance, reliability, and timing. Section 3.1 retrieves past frames according to the evidence they provide for the target. Section 3.2 varies the fidelity of memory seen during training, and Section 3.3 controls when each memory role is read during denoising. Section 3.4 then describes a unified memory reader that integrates all memories with role identity and target-relative geometry.

**Preliminaries.** Let $o_i$, $a_i$, and $p_i$ denote the observation, action, and known camera pose at frame index $i$, and let $z_i = E(o_i)$ be its frozen-VAE latent. We use $L_{\text{loc}}$ throughout to denote the *local context window* length. A training target set is $\mathcal{T} = \{n, \dots, n + L_{\text{loc}} - 1\}$, while inference uses a sliding window of $L_{\text{loc}}$ frames (Xiao et al., 2025). Under the Diffusion-Forcing objective (Chen et al., 2024), each target receives an independent timestep $t_i \in \{0, \dots, T-1\}$ and noise $\epsilon_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$,

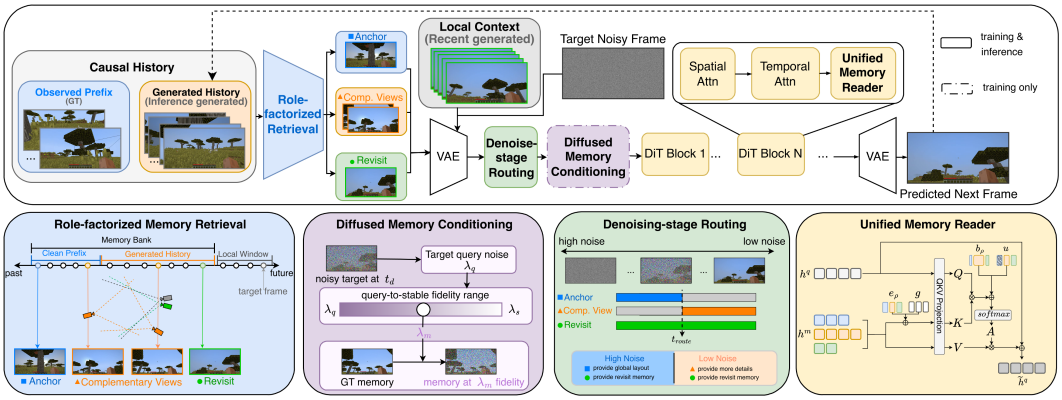


Figure 2: **Overview of W$^4$Mem.** W$^4$Mem organizes memory use around relevance, reliability, and timing. *Role-factorized retrieval* first retrieves anchors, complementary views, and revisits from the causal history. At each denoising step, *denoise-stage routing* selects the active memory roles, while *diffused memory conditioning* varies their fidelity during training according to the target noise level. In each DiT block, target tokens attend to active memory tokens through a *unified memory reader* whose keys combine role identity and target-relative geometry.

giving $z_i^{(t_i)} = \sqrt{\bar{\alpha}_{t_i}} z_i + \sqrt{1-\bar{\alpha}_{t_i}} \epsilon_i$ and velocity target $\nu_i = \sqrt{\bar{\alpha}_{t_i}} \epsilon_i - \sqrt{1-\bar{\alpha}_{t_i}} z_i$. The history before the target set is $\mathcal{H}_n = \{(\hat{z}_j, p_j, j) : 0 \le j < n\}$, where $\hat{z}_j$ is an encoded ground-truth observation during training or a completed prediction during inference. The retriever selects $\mathcal{M}_n = \mathcal{R}(\mathcal{H}_n; p_{i i \in \mathcal{T}})$ from this history and provides it as additional conditioning to the velocity predictor:

$$\mathcal{L}(\theta) = \frac{1}{L_{\text{loc}}} \sum_{i \in \mathcal{T}} w(t_i)\ \text{MSE}\big(\hat{\nu}_\theta(z_i^{(t_i)}, t_i, a_i, p_i, \mathcal{M}_n), \nu_i\big). \tag{1}$$

### 3.1 Relevance: Role-Factorized Retrieval

For a target set $\mathcal{T}$, W$^4$Mem retrieves the external memory $M_n$ into three role-specific groups. Anchors $\mathcal{M}_n^{\text{anc}}$ are selected from ground truth observed history and provide a stable reference for scene identity. Revisits $M_n^{\text{rev}}$ cover most of the query view, while complementary views $\mathcal{M}_n^{\text{cv}}$ provide additional evidence from a different camera pose. The complete memory is:

$$\mathcal{M}_n = \mathcal{M}_n^{\text{anc}} \uplus \mathcal{M}_n^{\text{cv}} \uplus \mathcal{M}_n^{\text{rev}}, \tag{2}$$

where $\uplus$ denotes a role-preserving disjoint union. During training, all three sets are selected from ground-truth history before Diffused Memory Conditioning is applied. At inference, anchors come only from the observed prefix, whereas revisits and complementary views may also come from generated history. The query is the final target frame in $\mathcal{T}$ during training and the frame currently being generated at inference. Appendix B, Algorithm 1 and Algorithm 2 give the detailed selection rules at inference and during training.

**Anchors provide a clean reference for scene identity.** As the rollout grows, recent history increasingly contains generated frames that can drift from the observed scene. We therefore keep a small set of anchors from clean input frames. They provide a stable reference for the world's appearance and global layout. They are selected to cover different camera poses and are fixed during inference, so the model can continue to access the same scene reference throughout the rollout.

**Revisits recover content needed by the target view.** Revisits aim to provide most of the revisiting information for the target; they are selected from past frames outside the local context for views that are close to the current camera and overlap the target field of view. We first shortlist candidates by pose similarity and discard those that overlap with the target below a threshold. This follows the view-matching intuition of WorldMem (Xiao et al., 2025).

**Complementary views add pose-diverse evidence.** A high-overlap revisit is useful for recovering visible content, but it can still contain limited information when the target region was only partially covered or viewed from a similar direction. We therefore reserve part of the memory budget for nearby views with different camera poses. These frames are required to overlap the target or a selected revisit, while pose diversity encourages them to contribute different geometric evidence. Exact overlap tests, sampling grids, and thresholds are given in Appendix B.

### 3.2 Reliability: Diffused Memory Conditioning

Training uses only ground-truth history, whereas inference may retrieve imperfectly generated frames. DMC narrows this gap by varying the fidelity of active memory latents with the same forward diffusion process used for the targets. A noisy target pack is paired with a broad range of memory qualities; as the targets become cleaner, this range narrows toward higher-fidelity memory.

Let $\lambda(t) = \log \bar{\alpha}_t - \log(1 - \bar{\alpha}_t)$ denote the log-SNR schedule. We define $\lambda_q = \max_{i \in \mathcal{T}} \lambda(t_i)$ as the log-SNR of the cleanest target in the pack and $\lambda_s = \lambda(t_s)$ as the stabilization level used at inference. For each active memory frame $j$,

$$\widetilde{\lambda}_j \sim \mathcal{U}[\lambda_q,\, \max(\lambda_q, \lambda_s)]\,, \qquad \widetilde{t}_j = \arg\min_{t \in \{0,\dots,T-1\}} \left|\lambda(t) - \widetilde{\lambda}_j\right|,$$

$$(z_j^{\text{in}}, t_j^{\text{in}}) = \begin{cases} (\sqrt{\bar{\alpha}_{t_s}}\, \hat{z}_j,\ t_s)\,, & \xi_j = 1, \\ \left(\sqrt{\bar{\alpha}_{\widetilde{t}_j}}\, \hat{z}_j + \sqrt{1 - \bar{\alpha}_{\widetilde{t}_j}}\, \epsilon_j,\ \widetilde{t}_j\right), & \xi_j = 0, \end{cases} \tag{3}$$

where $\epsilon_j \sim \mathcal{N}(0, I)$ and $\xi_j \sim \text{Bernoulli}(p_{\text{stab}})$. When $\lambda_q > \lambda_s$, the sampling interval collapses to $\lambda_q$. The stabilized branch matches the deterministic memory transformation used during inference. The diffused branch provides a simple way to simulate the generated imperfect memory during training time, rather than a heavy model to generate autoregressive rollouts. The frame's camera pose, temporal index, and role label remain unchanged. After this transformation, each active memory in $M_n$ contains $(z_j^{\text{in}}, t_j^{\text{in}})$ together with those unchanged attributes, and the resulting $M_n$ is passed to the DiT in Eq. (1). Memory frames receive no separate denoising loss, so the reader is trained only through its contribution to the target prediction $\hat{\nu}_\theta$. At inference, every active memory uses the stabilized branch.

### 3.3 Timing: Denoising-Stage Routing

Denoising provides a natural coarse-to-fine axis for allocating memory. At high noise levels, anchors provide clean observations that establish the world's appearance and global layout. As denoising progresses, complementary views contribute additional geometric evidence and refine viewpoint-dependent details. Revisits remain active throughout because their overlap with the target view makes them directly relevant at every stage of prediction. Let $t_{\text{route}} \in [0, T)$ be the stage boundary and let $\bar{t}$ denote the query noise level, the mean target timestep during training and the current DDIM timestep at inference. Routing depends on this query level, not on the sampled memory corruption in Eq. (3):

$$c = \mathbf{1}[\bar{t} \geq t_{\text{route}}], \qquad (\gamma_{\text{anc}}(c), \gamma_{\text{cv}}(c), \gamma_{\text{rev}}(c)) = (c, 1 - c, 1). \tag{4}$$

Thus $c = 1$ activates anchors and revisits at high noise, whereas $c = 0$ activates complementary views and revisits at low noise. Training and inference use the same gates. At inference, inactive memory streams are excluded before the DiT forward pass, so their key and value are never computed. The denoising-stage routing therefore controls both the memory evidence accessible to the model and the computational cost of memory attention at each denoising stage.

### 3.4 A Shared Role- and Geometry-Aware Reader

Within each DiT block, the three memory roles share one reader. For each active memory frame $j$, the memory stream is initialized from $z_j^{\text{in}}$ in Eq. (3), while $t_j^{\text{in}}$ conditions its AdaLN layers. We denote memory token $\ell$ at the current block by $h_{j\ell}^m$ and target token $u$ of frame $i$ by $h_{iu}^q$. Learned role embeddings distinguish why each memory frame was selected, while target-relative camera geometry describes its pose with respect to the target. Memory tokens provide conditioning only, they are neither denoised nor directly included in the prediction loss.

Camera geometry is encoded relative to each target frame. Given the camera-to-world transformation matrices $\mathbf{C}_i$ and $\mathbf{C}_j$, we define the relative transform from memory camera $j$ to target camera $i$ as $\mathbf{C}_{i \leftarrow j} = \mathbf{C}_i^{-1} \mathbf{C}_j$, with rotation $\mathbf{R}_{i \leftarrow j}$ and translation $\mathbf{c}_{i \leftarrow j}$. Let $d_\ell^{\text{cam}}$ be the unit camera ray through the center of memory patch $\ell$, computed from the camera intrinsics. Its direction in the target-camera coordinate system and its six-dimensional Plücker feature are

$$d_{ij\ell} = \text{normalize}(\mathbf{R}_{i \leftarrow j} d_\ell^{\text{cam}}), \qquad g_{ij\ell} = [\, c_{i \leftarrow j} \times d_{ij\ell}; d_{ij\ell}\,] \in \mathbb{R}^6. \tag{5}$$

For target token $\mathbf{h}^{\mathrm{q}}_{iu}$ and memory token $\mathbf{h}^{\mathrm{m}}_{j\ell}$ with role $\rho(j)$, each attention head forms

$$Q_{iu} = W_Q h^q_{iu}, \qquad K_{ij\ell} = W_K\left(h^m_{j\ell} + e_{\rho(j)} + W_g g_{ij\ell}\right), \qquad V_{j\ell} = W_V h^m_{j\ell},$$
$$A_{iu,j\ell} = \mathrm{softmax}_{j,\ell}\left(\frac{Q_{iu}^\top K_{ij\ell}}{\sqrt{d_h}} + b_{\rho(j)} + \mu_j\right), \qquad O_{iu} = \sum_{j,\ell} A_{iu,j\ell} V_{j\ell}. \tag{6}$$

where $e_\rho$ is learned role embedding, $W_g$ projects the geometry. We use $b_\rho = -\log(\max(n_\rho, 1))$ to correct for the number $n_\rho$ of active frames assigned to role $\rho$, while $\mu = -\infty$ masks an unfilled memory slot. The head outputs are concatenated, projected, and used to update only the target stream.

### 3.5 Training and Autoregressive Inference

During training, W$^4$Mem retrieves a role-factorized memory set for each target set, applies denoising-stage routing, and presents active memory through one of the two branches in Eq. (3). The unified memory reader then uses these memories to condition target denoising, and the model is optimized with the target velocity loss. As a training augmentation, with probability $p_{\mathrm{sel}}$, the selected frames are replaced by randomly sampled causal frames while retaining their role embeddings. This encourages the reader to rely on frame content and camera geometry rather than the role embeddings alone.

At inference, the observed prefix initializes the memory bank, and anchors are selected once and fixed throughout the rollout. Revisit and complementary-view memories are retrieved from the updated bank before generating each frame, then held fixed throughout that frame's denoising trajectory. Completed predictions become part of the history for subsequent retrieval.

## 4 Experiments

### 4.1 Experimental Setup

**Benchmarks.** Following the baseline method WorldMem (Xiao et al., 2025), we evaluate 300 samples from the MineDojo test set (Fan et al., 2022). Each sample provides 600 observed frames as context, followed by a 500-frame autoregressive rollout. For the RealEstate10K dataset (Zhou et al., 2018), we follow the baseline setting and build a new long-horizon loop from each of the 200-frame test trajectories, yielding 100 samples. The first 100 ground-truth frames are used as context. The model then generates 100 frames forward along the remaining camera trajectory and another 200 frames along the reversed trajectory, returning to the initial camera position. This produces a 300-frame loop that tests whether the model can recover earlier scenes after leaving the local context.

**Baselines and metrics.** All methods use the same dataset-specific evaluation protocol, using the reproduction settings in Appendix A, with results averaged over two evaluation seeds. WorldMem (Xiao et al., 2025) is our baseline on both datasets. For comparison, we also evaluate Oasis (Decart et al., 2024), Matrix-Game 2.0 (He et al., 2025), and DecMem (Yang et al., 2026b) on Minecraft, and DFoT (Song et al., 2025), Geometry Forcing (Wu et al., 2026), and VMem (Li et al., 2025) on RealEstate10K. We report MSE, PSNR, and LPIPS (Zhang et al., 2018) on both datasets, pooled FID (Heusel et al., 2017) for distributional quality, and SSIM (Wang et al., 2004) on RealEstate10K. Minecraft FID pools all 150,000 frames.

**Implementation details.** For a fair comparison, W$^4$Mem and WorldMem use the same backbone, Oasis on Minecraft and DFoT on RealEstate10K, and share the frozen VAE and an 8-frame local context window. We retrain WorldMem on both datasets, on Minecraft with the authors' official code, and on RealEstate10K with our reimplementation, since no official implementation is released. We also match the memory budget and DDIM step count: 8 memory frames and 20 steps on Minecraft, and 4 memory frames and 50 steps on RealEstate10K. W$^4$Mem divides each budget among anchors, complementary views, and revisits: $(2, 4, 2)$ on Minecraft and $(1, 2, 1)$ on RealEstate10K, full configuration shown in Table 8. On each dataset, both methods are initialized from the same pretrained backbone weights and trained with 200k steps with global batch size 128 on 8 H200 GPUs, with the optimizer settings in Appendix D.

### 4.2 Long-Horizon Results on Minecraft

**Main Results.** Table 1 reports results over 500 generated Minecraft frames. W$^4$Mem achieves the best values across all four metrics. Relative to WorldMem, it reduces MSE by 36.1%, LPIPS by 35.6%, and FID by 82.0%, and increases PSNR by 3.68 dB. Compared with DecMem, the corresponding reductions are 6.4%, 7.6%, and 61.8%, with a 1.99 dB PSNR gain. The FID reduction

Table 1: **Comparison results on Minecraft**. Over 500 generated frames, averaged over 300 test samples with 2 seeds. FID is pooled over all 150,000 generated frames. Best values are bolded.

| Method | MSE ↓ | PSNR ↑ | LPIPS ↓ | FID ↓ |
|---|---|---|---|---|
| Oasis (Decart et al., 2024) | 0.1333 | 10.83 | 0.6884 | 148.90 |
| Matrix-Game 2.0 (He et al., 2025) | 0.2069 | 8.32 | 0.6726 | 123.02 |
| DecMem (Yang et al., 2026b) | 0.0327 | 20.66 | 0.2839 | 23.64 |
| WorldMem (Xiao et al., 2025) | 0.0479 | 18.97 | 0.4074 | 50.25 |
| **W$^4$Mem (Ours)** | **0.0306** | **22.65** | **0.2624** | **9.02** |




Figure 3: **Qualitative Comparison Results on Minecraft.** Starting from the same 600-frame observed context, WorldMem and W$^4$Mem generate the next 500 frames. WorldMem progressively loses the established terrain, tree line, and block geometry, whereas W$^4$Mem remains consistent with the ground-truth scene throughout the rollout. Additional results are presented in Appendix F.

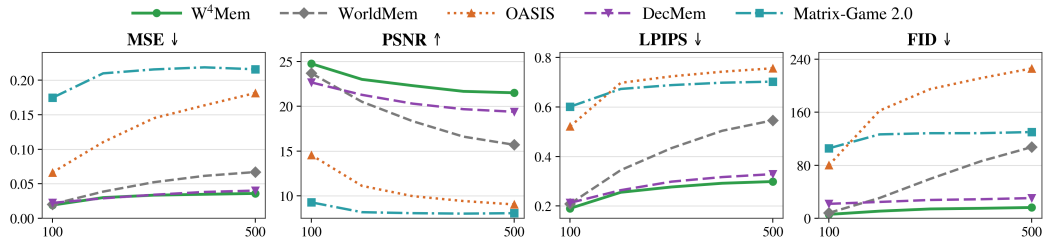


Figure 4: **Long-horizon Comparison Results on Minecraft.** We evaluate five consecutive 100-frame bins over a 500-frame rollout. Lower is better for all metrics except PSNR.

indicates better frame-level distributional quality across the long rollout. With the same backbone, local context window, DDIM schedule, and memory budget, the improvements mainly came from the design of W$^4$Mem's selection, presentation, and memory reads. We also halve the memory budget to four frames for both methods in Table 13. The FID of WorldMem increases from 50.25 to 84.57, while that of W$^4$Mem only increases from 9.02 to 10.78. With four frames, W$^4$Mem even outperforms WorldMem with eight frames on all four metrics. W$^4$Mem also improves inference efficiency. It generates each frame in **1.37 s** with **6.18 GB** of peak VRAM, compared with 1.79 s and 9.79 GB for WorldMem, corresponding to reductions of **23.5%** in inference time and **36.9%** in VRAM. By reading only the active memory roles at denoising stage, routing reduces memory-attention computation. W$^4$Mem therefore achieves higher generation quality with lower computation.

**Long-horizon stability.** Figure 4 shows that W$^4$Mem's advantage becomes increasingly pronounced later in the rollout. From the first to the final 100-frame bin, its MSE only increases from 0.019 to 0.035 (**1.9**×), while its FID increases from 5.87 to 16.27 (**2.8**×). Over the same horizon, WorldMem's MSE and FID increase by 3.3× and 13.3×, respectively. Although the two methods have similar MSE at the beginning of long-horizon generation, their errors diverge substantially at the end, reaching 0.035 for W$^4$Mem and 0.066 for WorldMem. The widening gap shows that W$^4$Mem limits degradation over the long rollout, rather than only improving the first generated frames.

Table 2: **Comparison results on RealEstate10K.** Over the complete 300-frame generated loop, averaged over 2 evaluation seeds. Best values are bolded.

| Method | MSE ↓ | PSNR ↑ | LPIPS ↓ | SSIM ↑ | FID ↓ |
|---|---|---|---|---|---|
| DFoT (Song et al., 2025) | 0.1367 | 9.79 | 0.7210 | 0.2361 | 170.0 |
| Geometry Forcing (Wu et al., 2026) | 0.0911 | 10.34 | 0.6276 | 0.2528 | 135.9 |
| VMem (Li et al., 2025) | 0.1098 | 10.15 | 0.7202 | 0.2413 | 208.9 |
| WorldMem (Xiao et al., 2025) | 0.1198 | 10.24 | 0.7829 | **0.2781** | 149.0 |
| **W$^4$Mem (Ours)** | **0.0901** | **11.50** | **0.5955** | 0.2728 | **114.5** |

**Revisit consistency.** We further test whether W$^4$Mem recovers the same content when the camera returns to a viewpoint previously seen. Each generated frame is paired with an earlier frame of high FOV overlap, taken from the generation or the ground truth, and we compare each pair with Inception-feature cosine similarity and LPIPS in Appendix E. At an FOV overlap threshold of 0.8, W$^4$Mem improves RC$_{\text{self}}$ from 0.758 to 0.892 and RC$_{\text{GT}}$ from 0.718 to 0.853 over WorldMem, and reduces LPIPS$_{\text{self}}$ from 0.417 to 0.253 and LPIPS$_{\text{GT}}$ from 0.491 to 0.317, shown in Table 14.

**Qualitative results on Minecraft.** Figure 3 and Appendix F shows the same trend as the quantitative results. W$^4$Mem preserves scenes and objects better than the baseline throughout the long-horizon 500-frame rollout. We also extend the rollout horizon to 1,000 frames, as illustrated in Figure 9. Even at twice the length, the generated scenes retain recognizable world structure.

### 4.3 Results on RealEstate10K

**Main Results.** On the 300-frame RealEstate10K loops benchmark, W$^4$Mem achieves the lowest MSE, LPIPS, and FID and the highest PSNR among the compared methods in Table 2. Compared with baseline WorldMem, MSE drops by 24.8%, LPIPS by 23.9%, and FID by 23.2%, while PSNR rises by 1.26 dB. These results show consistent improvements over the complete 300-frame generation.

**Long-horizon return consistency.** Figure 5 further shows the performance over the full loop rollout. The model first generates 100 frames forward along the original camera trajectory, and then generates another 200 frames along the reversed trajectory toward the initial camera position. The dotted line marks the beginning of this return phase. After the return begins, W$^4$Mem remains largely stable across all four metrics, with only small changes through the remaining 200 frames. In contrast, the other methods continue to degrade as the rollout proceeds. This suggests that W$^4$Mem can reuse earlier scene information when revisiting previous viewpoints, rather than accumulating additional error during the return trajectory. The qualitative results on RealEstate10K are shown in Appendix F.

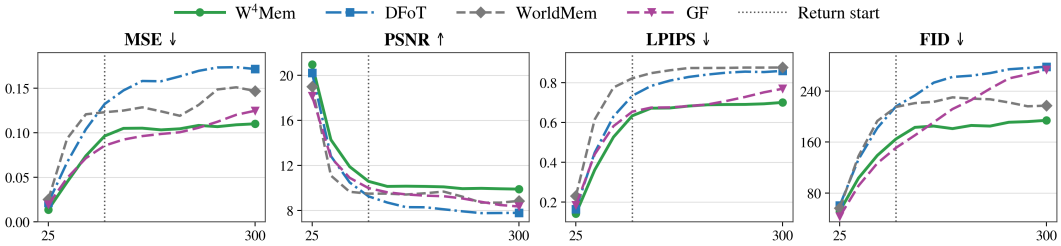


Figure 5: **Long-horizon Comparison Results on RE10K**, metrics over twelve 25-frame bins. The dotted line marks the return phase at generated frame 100; lower is better except PSNR.

### 4.4 Ablations

We start from the baseline and ablate each different component combination to see where the gains come from in Table 3. We then examine memory roles, corruption strategies, and routing schedules in more detail, as shown in Tables 4–6.

Table 3: **Components ablation on Minecraft.**

| Configuration | MSE ↓ | PSNR ↑ | LPIPS ↓ | FID ↓ |
|---|---|---|---|---|
| Baseline | 0.04793 | 18.976 | 0.4074 | 50.25 |
| + anchors | 0.04380 | 19.350 | 0.4180 | 51.80 |
| + complementary views | 0.05050 | 18.700 | 0.3480 | 32.80 |
| + RFR + unified reader | 0.04200 | 20.732 | 0.2930 | 16.20 |
| + RFR + unified reader + DMC | 0.03906 | 21.886 | 0.2745 | 11.05 |
| + RFR + unified reader + DMC + routing | **0.03063** | **22.651** | **0.2624** | **9.02** |

Table 4: **Memory design ablation on Minecraft.** Best are bolded.

| Variant | MSE ↓ | PSNR ↑ | LPIPS ↓ | FID ↓ |
|---|---|---|---|---|
| Revisit only | 0.04364 | 20.776 | 0.3389 | 33.53 |
| Anc.→rev. | 0.03929 | 21.478 | 0.3243 | 12.32 |
| C.v.→rev. | 0.04153 | 21.326 | 0.3017 | 12.38 |
| w/o role emb. | 0.03559 | 21.774 | 0.2787 | 9.21 |
| **W$^4$Mem (Ours)** | **0.03063** | **22.651** | **0.2624** | **9.02** |

Table 5: **Memory corruption ablation on Minecraft.** Best are bolded.

| Strategy | MSE ↓ | PSNR ↑ | LPIPS ↓ | FID ↓ |
|---|---|---|---|---|
| Shuffled query | 0.03899 | 21.958 | 0.2791 | 9.80 |
| Uniform sample | 0.03874 | 21.465 | 0.3265 | 19.29 |
| w/o DMC | 0.04327 | 21.146 | 0.3683 | 19.74 |
| **W$^4$Mem (Ours)** | **0.03063** | **22.651** | **0.2624** | **9.02** |

Table 6: **Denoising-stage routing ablation and inference cost on Minecraft.** Best are bolded.

| Variant | Route (anc./cv/rev.) | MSE ↓ | PSNR ↑ | LPIPS ↓ | FID ↓ | Memory Frames | s/f ↓ |
|---|---|---|---|---|---|---|---|
| Reversed routing | low/high/all | 0.03568 | 21.252 | 0.3089 | 13.31 | 5 | **1.37** |
| No-route | all/all/all | 0.03906 | 21.886 | 0.2745 | 11.05 | 8 | 1.70 |
| **W$^4$Mem (Ours)** | high/low/all | **0.03063** | **22.651** | **0.2624** | **9.02** | 5 | **1.37** |

**Component ablations.** In Table 3, adding anchors or complementary views to WorldMem improves some metrics. Anchors lower MSE but slightly worsen LPIPS and FID, while complementary views improve LPIPS and FID at the cost of MSE and PSNR. WorldMem reads every memory frame in the same way, regardless of why it was retrieved, so frames chosen for scene identity or viewpoint diversity can introduce content that does not match the target view. With the unified reader, which introduces role embeddings and target-relative geometry, both roles help: replacing either one with revisits worsens all four metrics as shown in Table 4. The RFR + unified reader row changes retrieval, role embeddings, geometry keys and count correction together; Table 4 separates part of this effect. DMC and routing then reduce FID from 16.20 to 11.05 and 9.02. Figure 6 also shows the rollout performance of each component ablation.

**Role-factorized memory selection.** We replace anchors or complementary views with revisits in W$^4$Mem. All variants in Table 4 keep same memory budget. We replace the two anchors or four complementary views with revisit frames. Replacing anchors increases MSE by 28.3% and FID by 36.6%; replacing complementary views increases MSE by 35.6% and FID by 37.3%. Removing role embeddings also worsens performance. The *Revisit-only* variant fills all memory with the WorldMem-style selector. Its MSE and FID increase to 0.043 and 33.53. It retains our unified reader and DMC, but removes role embeddings, count correction, and denoising-stage routing.

**Query-conditioned memory corruption.** As shown in Table 5, we ablate different memory corruption strategies here. Without DMC, performance increases MSE by 41.3%, LPIPS by 40.4% and FID by 118.8%. We also test two more different noise memory sampling strategies. Shuffling the query bound raises MSE by 27.3%; keeping the bound but sampling timesteps uniformly raises FID to 19.29. These show that adding noise helps, but we get better results by setting the corruption range relative to the current targets and sampling noise levels in log-SNR space.

**Denoising-stage memory routing.** In Table 6, we compare our routing schedule with a reversed schedule and a no-routing variant. Both schedules read five memory frames per denoising step on average. Reading anchors at high noise and complementary views at low noise lowers FID from 13.31 to 9.02 compared with the reversed schedule. Removing routing raises MSE by **27.5%** and FID by **22.5%**, while PSNR drops by 0.77 dB. It requires reading all eight memory frames at every step, increasing runtime from 1.37 to 1.70 s/frame.

## 5 Conclusion

We proposed W$^4$Mem to improve long-horizon generation in interactive video world models by considering relevance, reliability, and timing in memory use. We select anchors, revisits, and complementary views through role-factorized memory retrieval, then use Denoising-Stage Routing to decide when to read them. The model is trained with Diffused Memory Conditioning, which ties memory corruption to the query's denoising state. On Minecraft and RealEstate10K, W$^4$Mem achieves better results than baseline with the same backbone and memory budget. Dynamic memory routing at different denoising steps also cuts inference time and GPU memory use. The ablations consistently show gains across all three designs: which memory to use, how to present it during training, and when to read it. These results show that relevance, reliability, and timing all matter when using memory for long-horizon generation.

## Reproducibility Statement

We describe W$^4$Mem in Section 3, with further details of memory selection and corruption in Appendices B and C. Section 4 and Appendix A describe the datasets, evaluation protocols, and baseline configurations. Architecture and optimization settings are provided in Appendix D, with memory, diffusion, and sampling configurations in Table 8. Appendix E explains metric aggregation and reports Minecraft results for individual evaluation seeds. We will publicly release the code and model checkpoints.

## Ethics Statement

This study uses Minecraft data from MineDojo (Fan et al., 2022) and RealEstate10K (Zhou et al., 2018) to investigate long-horizon scene consistency in video world models. Our experiments focus on scene generation and do not involve person-specific synthesis. The videos generated in this study are intended for academic research and evaluation.

## AI Use Statement

We used large language models to check for typographical and grammatical errors and to make minor language edits. The authors take full responsibility for the content of the paper.

## References


Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.

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.

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.

Taiye Chen, Xun Hu, Zihan Ding, and Chi Jin. Learning world models for interactive video generation. *Advances in Neural Information Processing Systems*, 38:154456–154483, 2025.

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*, 2026.

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

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35: 18343–18362, 2022.

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.

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

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

Junchao Huang, Ziyang Ye, Xinting Hu, Tianyu He, Guiyu Zhang, Shaoshuai Shi, Jiang Bian, and Li Jiang. Live: Long-horizon interactive video world modeling, 2026. URL https://arxiv.org/abs/2602.03747.

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, 2025.

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.

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.

Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. *arXiv preprint arXiv:2502.06764*, 2025.

AlayaWorld Team, Kaipeng Zhang, Chuanhao Li, Yifan Zhan, Yongtao Ge, Yuanyang Yin, Jiaming Tan, Kang He, Liaoyuan Fan, Mingliang Zhai, et al. Alayaworld: Interactive long-horizon world modeling–full technical report. *arXiv preprint arXiv:2607.18367*, 2026.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *International Conference on Learning Representations*, volume 2025, pp. 73754–73776, 2025.

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

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.

Zhengxuan Wei, Xu Guo, Xinghui Li, Xunzhi Xiang, Min Wei, Yiran Zhu, Qiulin Wang, Xintao Wang, Pengfei Wan, Xiangwang Hou, et al. Geometry-aware implicit memory for video world models. *arXiv preprint arXiv:2606.02436*, 2026.

Haoyu Wu, Diankun Wu, Tianyu He, Junliang Guo, Yang Ye, Yueqi Duan, and Jiang Bian. Geometry forcing: Marrying video diffusion and 3d representation for consistent world modeling. In *International Conference on Learning Representations*, volume 2026, pp. 119186–119209, 2026.

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, 2025.

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, 2025.

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. In *International Conference on Learning Representations*, 2026a.

Zhenhao Yang, Xiaoshi Wu, Zhengyao Lv, Xiaoyu Shi, Xintao Wang, Pengfei Wan, Kun Gai, and Kwan-Yee K Wong. Decmem: Towards minute-long consistent world generation with decoupled memory. *arXiv preprint arXiv:2605.31336*, 2026b.

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, 2025.

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*, 2026.

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.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

# APPENDIX

## A EXPERIMENTAL DETAILS

This appendix specifies the evaluation protocols, memory configuration, and nominal cost of denoising-stage routing. The selection rules, memory corruption, and architecture are detailed in Appendices B–D.

**Minecraft evaluation.** Following WorldMem, we evaluate 300 MineDojo test samples at $360 \times 640$ resolution. Each sample provides 600 observed frames followed by a 500-frame autoregressive rollout. We divide the generated segment into five consecutive 100-frame bins. MSE, PSNR, and LPIPS are evaluated over all 150,000 generated frames, and pooled FID is computed from the complete generated-frame population. All reported results are averaged over two evaluation seeds.

**RealEstate10K evaluation.** We construct a long-horizon loop from each of 100 test trajectories at $256 \times 256$ resolution. The first 100 ground-truth frames of each 200-frame trajectory provide the observed context. The model then generates 100 frames along the remaining forward path and 200 frames along the reversed path back to the initial camera position. Each rollout therefore contains 300 generated frames, divided into twelve 25-frame bins, with the return phase beginning at generated-frame offset 100. We report MSE, PSNR, LPIPS, SSIM, and pooled FID, averaged over two evaluation seeds. Table 7 summarizes the two protocols.

Table 7: Evaluation protocols. Results are averaged over two evaluation seeds.

| Protocol quantity | Minecraft | RealEstate10K |
|---|---|---|
| Evaluation units | 300 test samples | 100 test trajectories |
| Frame resolution | $360 \times 640$ | $256 \times 256$ |
| Observed context frames | 600 | 100 |
| Generated frames per unit | 500 | 300 (100 forward + 200 return) |
| Horizon bins | $5 \times 100$ frames | $12 \times 25$ frames |
| Paired metrics | MSE, PSNR, LPIPS | MSE, PSNR, LPIPS, SSIM |
| Distributional metric | pooled FID | pooled FID |

**Training and memory configuration.** Table 8 lists the settings for each backbone. The external-memory budget is eight frames on Minecraft and four on RealEstate10K, allocated as $(2, 4, 2)$ and $(1, 2, 1)$ anchors, complementary views, and revisits, respectively. Both datasets use a separate eight-frame local context window. Appendix D gives the optimization details.

**Reproduction of comparison methods.** All models are evaluated on the same test samples, observed prefixes, action or camera trajectories, rollout lengths, and metric aggregation protocol for each dataset. We use architecture-specific settings for local context and memory access, expressed in the units used by each model. For DecMem on Minecraft, we set the short-term memory (STM) window to 6 and retrieve the top-27 memory tiles. For Geometry Forcing on RealEstate10K, each prediction uses 11 past frames plus one target frame. $W^4$Mem and WorldMem use an eight-frame local window on both datasets, with external-memory budgets of eight frames on Minecraft and four frames on RealEstate10K. Matrix-Game 2.0 follows the same Minecraft evaluation protocol. For DFoT, Geometry Forcing, and VMem, we use their official RealEstate10K checkpoints and run them on our loop protocol with their default camera conditioning and sampler settings.

## B ROLE-FACTORIZED MEMORY SELECTION

This appendix provides the full algorithmic details of the three memory selectors introduced in Section 3.1. All retrieval is causal. For a target pack beginning at index $n$, the distant candidate pool is $\mathcal{D}_n = \{j : 0 \le j < n - L_{\text{loc}}\}$. Each role has a maximum capacity $M_s$, with $\sum_s M_s \le M$.

**Pose distances.** Write camera pose $p_i$ in terms of its translation $\mathbf{c}_i$, yaw–pitch coordinates $\boldsymbol{\phi}_i$, and unit forward direction $\mathbf{f}_i$. Anchor selection uses

$$d_{\text{anc}}(i, j) = \sqrt{\|\mathbf{c}_i - \mathbf{c}_j\|_2^2 + \|\text{wrap}(\boldsymbol{\phi}_i - \boldsymbol{\phi}_j)\|_2^2}, \tag{7}$$

Table 8: **Memory, diffusion, and sampling configurations.** Budgets specify the maximum number of frames assigned to each memory role, selection can leave slots unfilled.

| Symbol | Meaning | Oasis | DFoT |
|---|---|---|---|
| *Diffusion schedule* | | | |
| $T$ | discrete noise levels | 1000 | 1000 |
| $t_s$ | stabilization level for context and clean memory | 14 | 0 |
| $t_{\text{route}}$ | boundary between denoising stages | $T/2$ | $T/2$ |
| *Memory capacity and routing* | | | |
| $L_{\text{loc}}$ | local context frames | 8 | 8 |
| $M_{\text{anc}}, M_{\text{cv}}, M_{\text{rev}}$ | anchor, complementary, revisit budgets | 2, 4, 2 | 1, 2, 1 |
| $W_{\text{anc}}$ | training anchor window | 100 frames | 100 frames |
| *Frustum coverage* | | | |
| $R_{\max}$ | frustum truncation radius | $30\,\text{m}$ | $4S$ |
| $N_{\text{pts}}$ | query-frustum probes | 4096 | 256 |
| $\tau_{\text{rev}}$ | minimum revisit overlap | 0.6 | 0.6 |
| $\Omega_r^{\text{co}}, \Omega_r^{\text{fi}}$ | coarse / fine overlap probe grids | $4{\times}6{\times}2$ / $8{\times}14{\times}8$ | $4{\times}6{\times}2$ / $8{\times}14{\times}8$ |
| *Candidate selection and pose diversity* | | | |
| $N_{\text{pre}}$ | pose-preselected candidates | 32 | – |
| $\tau_{\text{cv}}$ | minimum shared area | 0.3 | 0.3 |
| $\delta_{\text{cv}}$ | minimum pose novelty | 0.1 | 0.1 |
| $s_p$ | translation scale in $d_{\text{pose}}$ | $30\,\text{m}$ | $S$ |
| *Training-time regularization* | | | |
| $p_{\text{sel}}$ | selection-dropout probability | 0.1 | 0 |
| $p_{\text{stab}}$ | stabilized-memory branch probability | 0.25 | 0.25 |
| $\gamma_{\text{snr}}$ | inherited loss-weight decay in $w(t_i)$ | 0.96 | – |
| $\text{SNR}_{\max}$ | inherited SNR clip in $w(t_i)$ | 5 | – |
| *Sampler and architecture* | | | |
| $N_{\text{DDIM}}, \eta$ | DDIM sampling steps, stochasticity | 20, 0 | 50, 0 |
| $P$ | patches per frame | 144 | 1024 / 256 |
| $D$ | backbone hidden width | 1024 | 128 / 256 / 576 / 1152 |
| $N_h$ | attention heads | 16 | 9 |
| *Trainable parameters* | | | |
| - | W$^4$Mem | 910.6M | 554.4M |
| - | WorldMem | 945.2M | 587.1M |

where wrap returns the shortest signed angular difference in the stored angle units. This distance combines the raw translation and yaw–pitch coordinates without rescaling. Complementary-view selection and revisit preselection instead use the dimensionless distance

$$d_{\text{pose}}(i,j) = \frac{\|\mathbf{c}_i - \mathbf{c}_j\|_2}{s_p} + \frac{\arccos(\text{clip}(\mathbf{f}_i^\top \mathbf{f}_j, -1, 1))}{\pi}, \tag{8}$$

where $s_p$ normalizes translation and clip restricts the inverse-cosine argument to $[-1, 1]$. On Minecraft, $s_p = R_{\max} = 30\,\text{m}$. RealEstate10K poses have no metric scale, so we set $s_p = S$ and $R_{\max} = 4S$, where $S$ is the maximum camera-center distance between valid candidate frames and the active reference views in the current retrieval call, with $S = 1$ when this maximum is below $10^{-6}$.

**Frustum overlap and union coverage.** We estimate the overlap between two camera frusta using fixed 3D samples. For a reference camera $r$, let $\Omega_r^g$ be a deterministic set of points inside its frustum, truncated at $R_{\max}$, and let $\chi_j(\mathbf{x}) \in \{0, 1\}$ indicate whether $\mathbf{x}$ also lies inside the truncated frustum of candidate camera $j$. The superscript $g \in \{\text{rev}, \text{co}, \text{fi}\}$ denotes the sampling pattern used for revisit selection and the coarse and fine complementary-view passes. In particular, $|\Omega_q^{\text{rev}}| = N_{\text{pts}}$. We define the directed overlap as

$$\text{ov}_g(j \mid r) = \frac{1}{|\Omega_r^g|} \sum_{\mathbf{x} \in \Omega_r^g} \chi_j(\mathbf{x}). \tag{9}$$

Because the samples are defined in the reference frustum, $\text{ov}_g(j \mid r)$ need not equal $\text{ov}_g(r \mid j)$. The score captures geometric overlap and does not model occlusion.

For revisit selection, we measure how much of the query frustum is covered by a set of candidate cameras:

$$\mathrm{cov}(S \mid q) = \frac{1}{|\Omega_q^{\mathrm{rev}}|} \sum_{\mathbf{x} \in \Omega_q^{\mathrm{rev}}} \mathbf{1}\left[\max_{j \in S} \chi_j(\mathbf{x}) = 1\right], \tag{10}$$

where $\mathrm{cov}(\varnothing \mid q) = 0$. Each query sample is counted once, even when it is covered by multiple cameras, so repeatedly covering the same region gives no additional gain.

**Anchor selection.** During training, anchors are selected from the $W_{\mathrm{anc}}$ observations immediately preceding the target pack:

$$\mathcal{A}_n = \{j : \max(0, n - W_{\mathrm{anc}}) \le j < n\}.$$

At inference, the anchor pool is the full observed prefix with 600 frames on Minecraft and 100 frames on RealEstate10K. We initialize the set with the oldest available observation, $S_1^{\mathrm{anc}} = \{\min \mathcal{A}_n\}$, and apply farthest-point sampling:

$$\begin{aligned} j_k^{\mathrm{anc}} &= \operatorname*{arg\,max}_{j \in \mathcal{A}_n \setminus S_{k-1}^{\mathrm{anc}}} \min_{\ell \in S_{k-1}^{\mathrm{anc}}} d_{\mathrm{anc}}(j, \ell), \\ S_k^{\mathrm{anc}} &= S_{k-1}^{\mathrm{anc}} \cup \{j_k^{\mathrm{anc}}\}. \end{aligned} \tag{11}$$

At inference, anchors are selected once and kept fixed throughout the rollout. During training, they are reselected for every target pack, with $W_{\mathrm{anc}} = 100$.

**Revisit selection.** The query is the final target frame $q = \max \mathcal{T}$ during training and the frame currently being generated at inference. From the distant pool $\mathcal{D}_n$, we first select up to $N_{\mathrm{pre}}$ candidates nearest to the query under the pose distance in Eq. (8). We discard candidates with $\mathrm{ov}_{\mathrm{rev}}(j \mid q) < \tau_{\mathrm{rev}}$ and denote the surviving set by $\mathcal{V}_n^{\mathrm{rev}}$. Starting from $S_0^{\mathrm{rev}} = \varnothing$, we greedily select the remaining candidate with the largest increase in query-frustum union coverage:

$$\begin{aligned} j_k^{\mathrm{rev}} &= \operatorname*{arg\,max}_{j \in \mathcal{V}_n^{\mathrm{rev}} \setminus S_{k-1}^{\mathrm{rev}}} \left[\mathrm{cov}(S_{k-1}^{\mathrm{rev}} \cup \{j\} \mid q) - \mathrm{cov}(S_{k-1}^{\mathrm{rev}} \mid q)\right], \\ S_k^{\mathrm{rev}} &= S_{k-1}^{\mathrm{rev}} \cup \{j_k^{\mathrm{rev}}\}. \end{aligned} \tag{12}$$

Ties are broken by higher query overlap and then by recency. Selection ends after $M_{\mathrm{rev}}$ frames have been chosen or the candidate set is exhausted. We denote the final selected set by $S_\star^{\mathrm{rev}}$.

**Complementary-view selection.** Complementary views are selected relative to both the target views and the chosen revisits. The reference set is $\mathcal{B}_n = \mathcal{T} \cup S_\star^{\mathrm{rev}}$ during training and $\mathcal{B}_n = \{n\} \cup S_\star^{\mathrm{rev}}$ during inference. Selected revisits are removed from the distant candidate pool.

For each reference, the coarse pass keeps an equal number of candidates from two rankings, overlap alone and overlap weighted by pose diversity. We merge the resulting shortlists across all references. The fine pass then keeps candidates that overlap at least one reference:

$$\max_{r \in \mathcal{B}_n} \mathrm{ov}_{\mathrm{fi}}(j \mid r) \ge \tau_{\mathrm{cv}},$$

forming the filtered pool $\widetilde{\mathcal{D}}_n$. Starting from $S_0^{\mathrm{cv}} = \varnothing$, we select complementary views by

$$\begin{aligned} j_k^{\mathrm{cv}} &= \operatorname*{arg\,max}_{j \in \widetilde{\mathcal{D}}_n \setminus S_{k-1}^{\mathrm{cv}}} \min_{\ell \in \mathcal{B}_n \cup S_{k-1}^{\mathrm{cv}}} d_{\mathrm{pose}}(j, \ell), \\ S_k^{\mathrm{cv}} &= S_{k-1}^{\mathrm{cv}} \cup \{j_k^{\mathrm{cv}}\}. \end{aligned} \tag{13}$$

The distance is measured against both the reference set and previously selected complementary views, so each new frame must add a distinct viewpoint beyond the evidence already available. Selection stops after $M_{\mathrm{cv}}$ frames or when the best remaining pose novelty falls below $\delta_{\mathrm{cv}}$. Ties are broken by higher fine overlap and then by recency.

Finally, the indices in $S_\star^{\mathrm{anc}}$, $S_\star^{\mathrm{rev}}$, and $S_\star^{\mathrm{cv}}$ are converted to records $(\hat{z}_j, p_j, j)$ and assigned to their corresponding memory streams.

**Frustum membership.** To evaluate $\chi_j(\mathbf{x})$, we first transform point $\mathbf{x}$ into the coordinate system of camera $j$:

$$\mathbf{x}^{\mathrm{cam}} = \mathbf{R}_j^\top (\mathbf{x} - \mathbf{c}_j).$$

**Algorithm 1** Role-factorized memory selection at inference.

**Require:** Observed-prefix poses $\mathcal{P}^{\mathrm{obs}}$, history poses $\mathcal{P}_{<n}$, and query pose $p_n$
**if** $S_\star^{\mathrm{anc}}$ is not initialized **then**
  Select $M_{\mathrm{anc}}$ anchors from $\mathcal{P}^{\mathrm{obs}}$ by Eq. (11), seeded by the oldest frame
$\mathcal{D}_n \leftarrow \{j : 0 \le j < n - L_{\mathrm{loc}}\}$
From the $N_{\mathrm{pre}}$ candidates in $\mathcal{D}_n$ nearest to $p_n$, keep those with $\mathrm{ov}_{\mathrm{rev}}(j \mid n) \ge \tau_{\mathrm{rev}}$
Select up to $M_{\mathrm{rev}}$ revisits $S_\star^{\mathrm{rev}}$ by Eq. (12)
$\mathcal{B}_n \leftarrow \{n\} \cup S_\star^{\mathrm{rev}}$
Form $\widetilde{\mathcal{D}}_n$ from $\mathcal{D}_n \setminus S_\star^{\mathrm{rev}}$ using the coarse shortlist and fine-overlap threshold $\tau_{\mathrm{cv}}$
Select up to $M_{\mathrm{cv}}$ complementary views $S_\star^{\mathrm{cv}}$ by Eq. (13), stopping below $\delta_{\mathrm{cv}}$
**return** $\mathcal{M}_n = \mathcal{M}_n^{\mathrm{anc}} \uplus \mathcal{M}_n^{\mathrm{cv}} \uplus \mathcal{M}_n^{\mathrm{rev}}$

**Algorithm 2** Role-factorized memory selection during training.

**Require:** Target set $\mathcal{T}$, where $n = \min \mathcal{T}$ and $q = \max \mathcal{T}$; history poses $\mathcal{P}_{<n}$
$\mathcal{A}_n \leftarrow \{j : \max(0, n - W_{\mathrm{anc}}) \le j < n\}$
$\mathcal{D}_n \leftarrow \{j : 0 \le j < n - L_{\mathrm{loc}}\}$
**if** $\mathrm{Bernoulli}(p_{\mathrm{sel}}) = 1$ **then**
  Randomly sample $S_\star^{\mathrm{anc}} \subset \mathcal{A}_n$ and disjoint $S_\star^{\mathrm{rev}}, S_\star^{\mathrm{cv}} \subset \mathcal{D}_n$ at their respective budgets
  **return** the corresponding role-tagged memory streams
Select $M_{\mathrm{anc}}$ anchors $S_\star^{\mathrm{anc}}$ by Eq. (11), seeded by $\min \mathcal{A}_n$
From the $N_{\mathrm{pre}}$ candidates in $\mathcal{D}_n$ nearest to $p_q$, keep those with $\mathrm{ov}_{\mathrm{rev}}(j \mid q) \ge \tau_{\mathrm{rev}}$
Select up to $M_{\mathrm{rev}}$ revisits $S_\star^{\mathrm{rev}}$ by Eq. (12)
$\mathcal{B}_n \leftarrow \mathcal{T} \cup S_\star^{\mathrm{rev}}$
Form $\widetilde{\mathcal{D}}_n$ from $\mathcal{D}_n \setminus S_\star^{\mathrm{rev}}$ using the coarse shortlist and fine-overlap threshold $\tau_{\mathrm{cv}}$
Select up to $M_{\mathrm{cv}}$ complementary views $S_\star^{\mathrm{cv}}$ by Eq. (13), stopping below $\delta_{\mathrm{cv}}$
**return** $\mathcal{M}_n = \mathcal{M}_n^{\mathrm{anc}} \uplus \mathcal{M}_n^{\mathrm{cv}} \uplus \mathcal{M}_n^{\mathrm{rev}}$

For an image of size $W_{\mathrm{img}} \times H_{\mathrm{img}}$, focal lengths $(f_x, f_y)$, and a centered principal point, $\mathbf{x}$ lies inside the truncated frustum when

$$\begin{aligned} x_z^{\mathrm{cam}} > 0, \qquad & |x_x^{\mathrm{cam}}| \le x_z^{\mathrm{cam}} \frac{W_{\mathrm{img}}}{2f_x}, \\ \|\mathbf{x} - \mathbf{c}_j\|_2 \le R_{\max}, \quad & |x_y^{\mathrm{cam}}| \le x_z^{\mathrm{cam}} \frac{H_{\mathrm{img}}}{2f_y}. \end{aligned} \tag{14}$$

These conditions place the point in front of the camera, within the image bounds, and no farther than $R_{\max}$. This is a geometric frustum test and does not account for occlusion.

**Frustum sampling.** Revisit overlap and union coverage using fixed samples inside the truncated query frustum, with $N_{\mathrm{pts}} = 4096$ on Minecraft and $N_{\mathrm{pts}} = 256$ on RealEstate10K. The same samples are reused for every candidate compared with a given query. Complementary-view filtering uses deterministic grids over the two image-plane axes and depth, with $4 \times 6 \times 2$ for the coarse pass and $8 \times 14 \times 8$ for the fine pass. Keeping these samples fixed makes candidate–reference comparisons consistent without requiring scene rendering.

**Training-time selection.** With probability $p_{\mathrm{sel}}$, training replaces all structured selectors with sampling without replacement from their respective causal pools, retaining budgets and role labels. The reader must therefore use content and geometry rather than rely on role identity alone. Algorithms 1 and 2 summarize inference and training; selection dropout is used only during training.

**Memory ablation settings.** The memory-design ablations retain our unified reader and DMC. In *Anc.→rev.*, the two anchor slots are filled with revisit frames. In *C.v.→rev.*, the four complementary-view slots are filled with revisit frames. Both tests retain eight memory slots. The *Revisit-only* variant selects eight frames using the WorldMem-style selector without an overlap threshold. These frames form one stream and remain active throughout denoising. Target-relative geometry and DMC are retained, while role embeddings and count correction are removed. This is an ablation of W$^4$Mem, not the WorldMem baseline reported in Table 1.

## C Diffusion Objective and Memory Corruption

**Target-only objective.** Training uses the velocity objective in Eq. (1). We retain the backbone's cumulative-SNR weighting, with decay $\gamma_{\mathrm{snr}} = 0.96$ and clipping threshold $\mathrm{SNR}_{\max} = 5$. Because

the weight accumulates along the causal frame chain, it depends on preceding noise levels as well as the current target timestep. Memory frames provide conditioning only, they receive no direct denoising loss, and the memory pathway is trained through its contribution to target predictions.

**Query-conditioned corruption interval.** For the inherited $T = 1000$-level diffusion process, the signal coefficient and log-SNR are

$$\bar{\alpha}_t = \prod_{s \le t}(1 - \beta_s), \qquad \lambda(t) = \log \bar{\alpha}_t - \log(1 - \bar{\alpha}_t), \tag{15}$$

where $\beta_s$ is the backbone's variance schedule. Larger timesteps correspond to lower log-SNR and noisier latents. The endpoints used by DMC are

$$\lambda_q = \max_{i \in \mathcal{T}} \lambda(t_i), \qquad \lambda_s = \lambda(t_s), \tag{16}$$

where $\lambda_q$ is the log-SNR of the cleanest target in the pack and $\lambda_s$ corresponds to the stabilization timestep used at inference. Given the target timesteps, each active memory frame $j$ independently samples a log-SNR level:

$$\widetilde{\lambda}_j \sim \mathcal{U}[\lambda_q, \max(\lambda_q, \lambda_s)]\,. \tag{17}$$

When $\lambda_q \ge \lambda_s$, the interval collapses and $\widetilde{\lambda}_j = \lambda_q$. Otherwise, sampling is uniform in log-SNR between the cleanest target and the stabilization level. We map the sampled log-SNR to the nearest discrete timestep:

$$\widetilde{t}_j = \operatorname*{arg\,min}_{t \in \{0,\dots,T-1\}} \left|\lambda(t) - \widetilde{\lambda}_j\right|. \tag{18}$$

In the diffused branch, $\widetilde{t}_j$ never exceeds the minimum target timestep.

**Memory presentation.** Let $\hat{z}_j$ denote the stored latent of memory frame $j$. During training, every $\hat{z}_j$ comes from a ground-truth frame. We apply the same forward diffusion process used for target latents:

$$\widetilde{z}_j = \sqrt{\bar{\alpha}_{\widetilde{t}_j}}\,\hat{z}_j + \sqrt{1 - \bar{\alpha}_{\widetilde{t}_j}}\,\epsilon_j, \qquad \epsilon_j \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{19}$$

This varies memory fidelity during training without reproducing all errors that occur in generated frames. Each active memory then uses either the stabilized or the diffused branch:

$$(z_j^{\text{in}}, t_j^{\text{in}}) = \begin{cases} (\sqrt{\bar{\alpha}_{t_s}}\,\hat{z}_j,\ t_s)\,, & \xi_j = 1, \\ (\widetilde{z}_j,\ \widetilde{t}_j), & \xi_j = 0, \end{cases} \qquad \xi_j \sim \text{Bernoulli}(p_{\text{stab}}). \tag{20}$$

The branch variable and corruption level are sampled independently per slot, allowing a pack to contain memories of different fidelities. Each memory frame receives the sampled timestep for a noisy slot and the dataset-specific stabilization timestep $t_s$ for a stabilized slot. These operations apply only to roles activated by Eq. (4); the routing stage is determined by the query, not by the sampled memory corruption.

**Stabilization at training and inference.** The stabilized branch uses $\text{stab}(z) = \sqrt{\bar{\alpha}_{t_s}}z$ without random noise, with $t_s = 14$ on Minecraft and $t_s = 0$ on RealEstate10K. It therefore differs from a forward-diffusion sample at the same timestep. At inference, this deterministic scaling is applied to completed context frames and all active memory frames. During training, the branch is selected with probability $p_{\text{stab}} = 0.25$, directly exposing the reader to the inference-time input transformation. Stabilization controls how a latent is presented; it does not make an imperfect generated frame equivalent to a ground-truth observation.

**Corruption ablation settings.** We use a zero-indexed 1000-step schedule with $t_s = 14$ (configured level 15). *Uniform sample* independently draws $t_m$ uniformly from $14, \dots, t_q$, setting $t_m = t_q$ when $t_q \le 14$. Here, $t_q$ is the minimum valid target timestep. *Shuffled query* takes the query log-SNR bound $\lambda_d$ from the next example in cyclic per-GPU batch order within the same noise stratum (mean target timestep $\ge 500$ or $< 500$), leaving singletons unchanged. Only the bound is reassigned, it samples log-SNR uniformly from $[\lambda_d, \max \lambda_d, \lambda(14)]$ and maps it to the nearest timestep. These variants retain the stabilized branch independently per slot with probability 0.25 $\sqrt{\bar{\alpha}_{14}}z_m$, without additive noise and with timestep 14. *Without DMC* and all variants at evaluation use only stabilized memory.

## D Architecture and Optimization Details

**Minecraft Backbone and model structure.** The DiT backbone uses width $D = 1024$, depth 16, $N_h = 16$ heads, and $P = 144$ patches on a $9 \times 16$ grid. Each block applies spatial, causal temporal, and target-to-memory attention. The memory branch has separate normalization parameters and a post-attention MLP. Targets attend causally to earlier targets; memory frames attend only to themselves with zero action conditions. The reader alone transfers memory information to targets and returns zero when no memory is active.

**RealEstate10K backbone and memory reader.** We use DFoT with the configuration listed in Table 8. The memory reader is inserted after each of 32 Transformer blocks, six encoder blocks at $32 \times 32$ resolution, twenty bottleneck blocks at $16 \times 16$, and six decoder blocks at $32 \times 32$. Memory frames follow the same U-ViT hierarchy as the local frames, giving hidden widths of 576 and 1152 at the two insertion resolutions. Relative Plücker and role features are projected to the corresponding layer width. Each insertion point has separate Q/K/V/O, FFN, AdaLN, and output-gate parameters, while the camera-and-role conditioning module is shared across blocks with width-specific projection heads.

**Target-relative camera geometry.** For camera-to-world transforms $\mathbf{C}_i, \mathbf{C}_j$, the transform from memory camera $j$ to target camera $i$ is

$$\mathbf{C}_{i\leftarrow j} = \mathbf{C}_i^{-1}\mathbf{C}_j = \begin{bmatrix} \mathbf{R}_{i\leftarrow j} & \mathbf{c}_{i\leftarrow j} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \tag{21}$$

The translation and rotation are expressed in the target frame. For a token at pixel offset $(u_\ell, v_\ell)$ from the principal point, the camera ray, target-relative direction and moment, and Plücker features are

$$\mathbf{d}_\ell^{\mathrm{cam}} = \mathrm{normalize}\big((u_\ell/f_x,\, v_\ell/f_y,\, 1)^\top\big), \tag{22}$$

$$\mathbf{d}_{ij\ell} = \mathrm{normalize}(\mathbf{R}_{i\leftarrow j}\mathbf{d}_\ell^{\mathrm{cam}}), \qquad \mathbf{m}_{ij\ell} = \mathbf{c}_{i\leftarrow j} \times \mathbf{d}_{ij\ell}, \tag{23}$$

$$\mathbf{g}_{ij\ell} = [\,\mathbf{m}_{ij\ell};\, \mathbf{d}_{ij\ell}\,] \in \mathbb{R}^6, \qquad \mathbf{G}_{ij} = [\,\mathbf{g}_{ij1}; \ldots; \mathbf{g}_{ijP}\,] \in \mathbb{R}^{P\times 6}. \tag{24}$$

Camera-frame rays can be precomputed for a fixed camera model and token grid. All roles share these target-dependent features, which are invariant to a common rigid transformation of the trajectory.

**Unified Memory Reader.** In the unified memory reader attention, queries use target visual features, keys add role embedding and target-relative geometry to memory features, and values contain timestep-conditioned visual features, without explicit role, geometry:

$$\mathbf{q}_{iu} = \mathbf{W}_Q \mathbf{h}_{iu}^{\mathrm{q}}, \tag{25}$$

$$\mathbf{k}_{ij\ell} = \mathbf{W}_K\big(\mathbf{h}_{j\ell}^{\mathrm{m}} + \mathbf{e}_{\rho(j)} + \mathbf{W}_g \mathbf{g}_{ij\ell}\big), \tag{26}$$

$$\mathbf{v}_{j\ell} = \mathbf{W}_V \mathbf{h}_{j\ell}^{\mathrm{m}}. \tag{27}$$

Role and geometry guide attention without entering the values directly. The attention logits are

$$L_{iu,j\ell} = \frac{\mathbf{q}_{iu}^\top \mathbf{k}_{ij\ell}}{\sqrt{d_h}} + b_{\rho(j)} + \mu_j, \qquad b_s = -\log\big(\max(n_s, 1)\big), \tag{28}$$

where $\mu_j = -\infty$ masks an unfilled slot and $n_s$ counts active frames of role $s$. To make the count correction explicit, let $\mathcal{S}_+ = \{s \in \mathcal{S} : n_s > 0\}$ and $E_{ij\ell} = \exp(\mathbf{q}_{iu}^\top \mathbf{k}_{ij\ell}/\sqrt{d_h})$ for filled slots. When at least one memory frame is active,

$$\sum_{j,\ell} \mathrm{softmax}_{(j,\ell)}\big(L_{iu,j\ell}\big)\, \mathbf{v}_{j\ell} = \frac{\sum_{s\in\mathcal{S}_+} n_s^{-1} \sum_{j\in s} \sum_\ell E_{ij\ell}\, \mathbf{v}_{j\ell}}{\sum_{s\in\mathcal{S}_+} n_s^{-1} \sum_{j\in s} \sum_\ell E_{ij\ell}}. \tag{29}$$

Dividing by the active frame count offsets stream cardinality without imposing equal role weights; attention still depends on learned query–key similarities.

**Reader implementation.** The zero-initialized role table $\mathbf{e} \in \mathbb{R}^{4\times D}$ has one row for targets and one for each memory role. Head outputs are concatenated, passed through the memory branch's MLP, and gated into target residuals. A zero dummy key/value pair avoids an undefined softmax for fully masked packs; its update is gated to zero.

Table 9: **Comparison of memory designs for long-horizon video world models.** ✓ denotes an explicit design, △ denotes a related but non-equivalent mechanism, and – denotes that no corresponding mechanism is explicitly described.

| Method | Memory representation | Retrieval | Geometry | Noise-aware training | Stage-aware reading |
|---|---|---|---|---|---|
| WorldMem | Frames + states | View overlap + recency | ✓ | – | – |
| Context-as-Memory | Frames | FOV overlap | ✓ | – | – |
| Long-term Spatial Memory | 3D spatial + episodic memory | Geometry-grounded | ✓ | – | – |
| VMem | Surfel-indexed views | Visible top-$K$ views | ✓ | – | – |
| DecMem | Global + local latent memory | Sparse blocks | ✓ | – | – |
| MemLearner | Context memory | Learned query | △ | – | – |
| AlayaWorld | Sink + compressed history + depth-warped spatial memory | Max-coverage via depth warping | ✓ | △ | – |
| GIM-World | Implicit memory tokens | Geometry-aware compression | ✓ | – | – |
| **W$^4$Mem (Ours)** | **Role-factorized frames** | **Role + geometry** | ✓ | ✓ | ✓ |

**Nominal routing cost.** Let $c^{(\kappa)}$ be the stage indicator in Eq. (4) at DDIM step $\kappa$, and let $\pi_{\mathrm{H}} = N_{\mathrm{DDIM}}^{-1} \sum_{\kappa=1}^{N_{\mathrm{DDIM}}} c^{(\kappa)}$ denote the fraction of high-noise steps. When all selected slots are filled, the mean number of memory frames read per step and the reduction relative to reading all roles are

$$\begin{aligned} \overline{M} &= \pi_{\mathrm{H}}(M_{\mathrm{anc}} + M_{\mathrm{rev}}) + (1 - \pi_{\mathrm{H}})(M_{\mathrm{cv}} + M_{\mathrm{rev}}), \\ \rho_{\mathrm{KV}} &= 1 - \frac{\overline{M}}{M_{\mathrm{anc}} + M_{\mathrm{cv}} + M_{\mathrm{rev}}}, \end{aligned} \tag{30}$$

where $\overline{M}$ counts active memory frames and $\rho_{\mathrm{KV}}$ denotes the corresponding reduction in the memory key/value budget. With $\pi_{\mathrm{H}} = 1/2$, routing reads four frames at high noise and six at low noise, giving $\overline{M} = 5$ rather than eight. The resulting 37.5% reduction concerns memory tokens, not total model computation or history storage. Actual occupancy can be lower when too few revisit candidates pass the overlap threshold or complementary-view selection stops early.

Relative to WorldMem, W$^4$Mem reduces peak GPU memory by 36.9% and inference time by 23.5%. This end-to-end measurement includes the local stream and backbone computation, which are not removed by memory routing.

**Optimization.** We train with AdamW, $\beta = (0.9, 0.99)$, weight decay $2 \times 10^{-3}$, and 500 warmup steps. Memory modules are trained from step zero at learning rate $8 \times 10^{-5}$. The DiT is unfrozen at step 10k and trained at $2 \times 10^{-5}$. Training runs for 200k steps with global batch size 128 on eight H200 GPUs.

**Comparison of Memory Designs** Table 9 compares recent memory-based video world models in terms of memory representation, retrieval, geometry, memory fidelity, and reading strategy. Most existing methods mainly differ in how memory is stored and retrieved, while W$^4$Mem additionally conditions memory fidelity on the query denoising state and changes memory usage across denoising stages. Here, ✓ indicates an explicit design, △ indicates a related but non-equivalent mechanism, and – indicates that no corresponding mechanism is explicitly described.

## E Additional Quantitative Results

This appendix provides additional quantitative results referenced in Section 4, including the first and final Minecraft bins, ablation results across the rollout, and the final RealEstate10K return bin.

**Minecraft bin endpoints.** Table 10 compares the first and final 100-frame bins for WorldMem and W$^4$Mem. Their MSE values are similar in the first bin, but the gap widens as the rollout progresses.

W$^4$Mem also achieves lower per-bin FID at both endpoints and shows substantially less degradation between them. These results complement the full rollout curves in Figure 4.

Table 10: **Minecraft first- and final-bin results.** Each bin contains 100 generated frames; growth is the ratio of bin 5 to bin 1. FID is computed separately for each bin, not pooled across the rollout. Lower values are better; the best values are bolded.

| Method | MSE ↓ | | | per-bin FID ↓ | | |
|---|---|---|---|---|---|---|
| | bin 1 | bin 5 | growth | bin 1 | bin 5 | growth |
| WorldMem | 0.02038 | 0.06696 | 3.3× | 8.10 | 107.40 | 13.3× |
| **W$^4$Mem (Ours)** | **0.01906** | **0.03593** | **1.9×** | **5.87** | **16.27** | **2.8×** |

**Pooled versus per-bin distributional metrics.** The FID in Table 1 is computed from all 150,000 generated frames, while Table 10 and Figure 4 evaluate each 100-frame bin against its corresponding reference population. Pooled FID is not the average of the bin-wise scores. The pooled statistic summarizes the complete rollout; the bin-wise scores reveal how distributional quality changes with generation horizon. The same distinction applies to pooled and per-bin FID on RealEstate10K.

**RealEstate10K final return bin.** Table 11 reports the final 25-frame bin, $[275, 300)$, in Figure 5. W$^4$Mem reduces MSE by 25.1% relative to WorldMem in this bin, close to the 24.8% reduction over the complete loop. It also improves PSNR, LPIPS, and FID, while its SSIM is lower than WorldMem's.

Table 11: **RealEstate10K quality in the final 25-frame bin.** Best values are bolded.

| Method | MSE ↓ | PSNR ↑ | LPIPS ↓ | SSIM ↑ | FID ↓ |
|---|---|---|---|---|---|
| DFoT | 0.1717 | 7.80 | 0.8587 | 0.1757 | 278.0 |
| WorldMem | 0.1470 | 8.85 | 0.8756 | **0.2604** | 217.0 |
| **W$^4$Mem (Ours)** | **0.1101** | **9.90** | **0.7000** | 0.2423 | **194.0** |

**Results across evaluation seeds.** Table 12 reports the individual results for evaluation seeds 42 and 43. W$^4$Mem outperforms WorldMem on all four metrics under both seeds. The variation between seeds is small relative to the performance gap. W$^4$Mem achieves FID values of 8.91 and 9.14, compared with 50.48 and 50.03 for WorldMem.

Table 12: **Minecraft results across evaluation seeds.**

| Method | Seed | MSE ↓ | PSNR ↑ | LPIPS ↓ | FID ↓ |
|---|---|---|---|---|---|
| WorldMem | 42 | 0.04905 | 18.896 | 0.4098 | 50.48 |
| WorldMem | 43 | 0.04681 | 19.049 | 0.4049 | 50.03 |
| W$^4$Mem (Ours) | 42 | 0.03017 | 22.691 | 0.2607 | 8.91 |
| W$^4$Mem (Ours) | 43 | 0.03109 | 22.612 | 0.2640 | 9.14 |

**Component ablations across the rollout.** Figure 6 compares component combinations starting from WorldMem. Anchors improve late-horizon MSE and PSNR, while complementary views improve LPIPS and FID. RFR with the unified reader gives more balanced gains, and adding DMC and routing further reduces degradation. The full model performs best on all four metrics in the final bin.

**Memory budget robustness.** We further test whether the gains of W$^4$Mem depend on the default eight-frame external-memory budget. We halve the budget to four frames while preserving the same 1:2:1 allocation across anchors, complementary views, and revisits. As shown in Table 13, W$^4$Mem consistently outperforms WorldMem under both memory budgets. With only four memory frames, W$^4$Mem remains close to its eight-frame performance across all four metrics, showing that the improvement does not rely on a larger memory capacity but on how the available memory is selected and used.

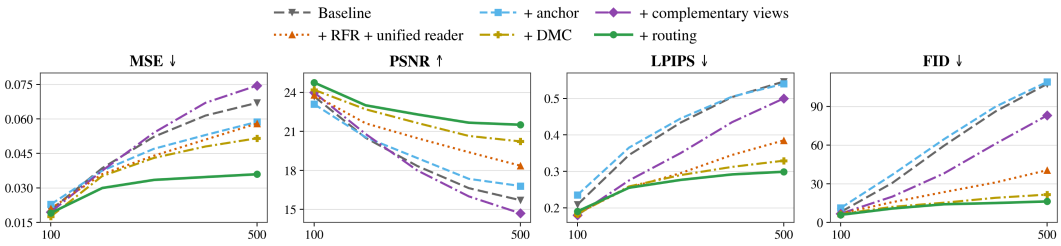


Figure 6: **Component ablations performance over the 500-frame rollout.** We compare component ablations starting from WorldMem over five consecutive 100-frame bins.

Table 13: **Memory-budget robustness on Minecraft.** We compare WorldMem and W$^4$Mem using four and eight external-memory frames. W$^4$Mem allocates the budgets as $(1, 2, 1)$ and $(2, 4, 2)$ anchors, complementary views, and revisits, respectively. Best values under each budget are bolded.

| Memory budget | Method | MSE ↓ | PSNR ↑ | LPIPS ↓ | FID ↓ |
|---|---|---|---|---|---|
| **4 frames** | WorldMem | 0.0662 | 16.91 | 0.5159 | 84.57 |
| | **W$^4$Mem (Ours)** | **0.0317** | **22.62** | **0.2606** | **10.78** |
| **8 frames** | WorldMem | 0.0479 | 18.97 | 0.4074 | 50.25 |
| | **W$^4$Mem (Ours)** | **0.0306** | **22.65** | **0.2624** | **9.02** |

Table 14: **Revisit consistency comparison on Minecraft.** Inception cosine similarity (RC) and LPIPS between each generated frame and an earlier matched view from the rollout (self) or the ground truth (GT). The FOV overlap thresholds 0.8 and 0.9 yield 191 and 113 eligible clips out of 300, comprising 4,129 and 1,301 frame pairs, respectively.

| | FoV overlap $> 0.8$ | | | | FoV overlap $> 0.9$ | | | |
|---|---|---|---|---|---|---|---|---|
| Method | $\text{RC}_{\text{self}}$ ↑ | $\text{RC}_{\text{GT}}$ ↑ | $\text{LPIPS}_{\text{self}}$ ↓ | $\text{LPIPS}_{\text{GT}}$ ↓ | $\text{RC}_{\text{self}}$ ↑ | $\text{RC}_{\text{GT}}$ ↑ | $\text{LPIPS}_{\text{self}}$ ↓ | $\text{LPIPS}_{\text{GT}}$ ↓ |
| WorldMem | 0.7576 | 0.7178 | 0.4169 | 0.4910 | 0.7645 | 0.7190 | 0.4078 | 0.4904 |
| **W$^4$Mem (Ours)** | **0.8922** | **0.8528** | **0.2527** | **0.3168** | **0.8985** | **0.8572** | **0.2343** | **0.3079** |

**Revisit consistency.** We evaluate whether generated content remains consistent when returning to a previously generated viewpoint. For each qualifying pair $(s, t)$, where $s < t$, we compute

$$\text{RC}_{\text{self}}(s,t) = \cos\Big(f(\hat{I}_t), f(\hat{I}_s)\Big), \qquad \text{RC}_{\text{GT}}(s,t) = \cos\Big(f(\hat{I}_t), f(I_s^{\text{GT}})\Big) \tag{31}$$

where $f$ extracts the 2048-dimensional Inception-v3 features used for FID, followed by $\ell_2$ normalization. The ground-truth reference is the original image at the earlier matched viewpoint, rather than the ground truth at the current frame. Pairs are selected from camera poses independently of either model, using a shared symmetric FOV frustum-coverage. We require a temporal separation of at least 100 frames and an intervening departure lasting at least 10 consecutive frames with overlap at most 0.2, excluding stationary-view continuation. Each target is paired with its highest-overlap eligible earlier frame. We also compute LPIPS on the same pairs,

$$\text{LPIPS}_{\text{self}}(s,t) = \text{LPIPS}\big(\hat{I}_t, \hat{I}_s\big), \qquad \text{LPIPS}_{\text{GT}}(s,t) = \text{LPIPS}\big(\hat{I}_t, I_s^{\text{GT}}\big), \tag{32}$$

which is more sensitive to local structure than Inception features. Matched views overlap strongly but are not identical, so LPIPS also reflects some viewpoint difference; the pairs are the same for both methods. As shown in Table 14, W$^4$Mem improves both self-consistency and agreement with earlier ground-truth observations. At overlap $> 0.8$, it raises $\text{RC}_{\text{self}}$ and $\text{RC}_{\text{GT}}$ by 0.135 and reduces $\text{LPIPS}_{\text{self}}$ and $\text{LPIPS}_{\text{GT}}$ by 39% and 35%. The gains hold under the stricter $> 0.9$ criterion. These results support that W$^4$Mem can better remember previously seen content when revisiting. Evaluating both scores is important because self-consistency alone can reward temporally coherent but inaccurate outputs.

## F Additional Qualitative Results

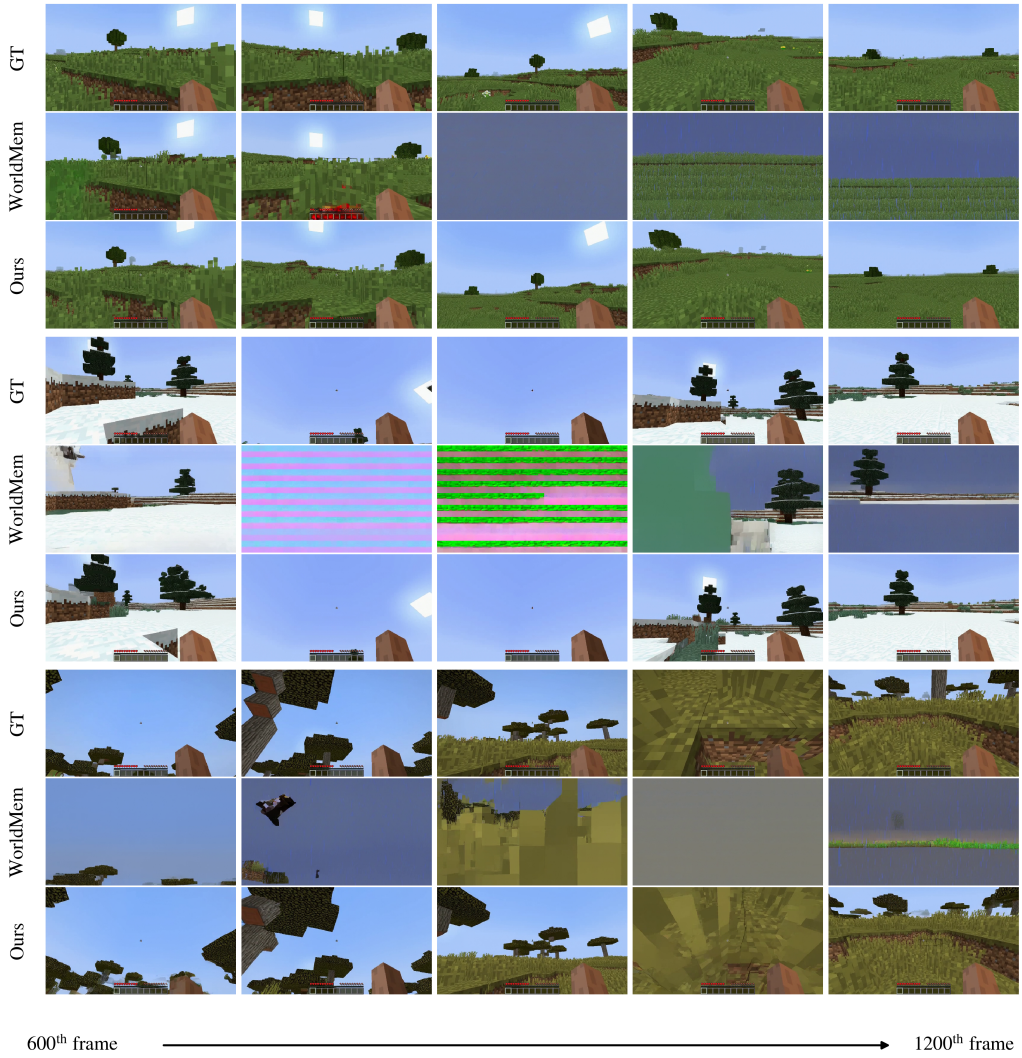


Figure 7: **Additional Qualitative Comparisons on Minecraft.** Three trajectories from frame 600 to 1200. When the camera turns toward the sky or ground and back, WorldMem collapses into blank frames or striped artifacts, while W$^4$Mem restores the original trees and terrain. Compare the last two columns with GT.

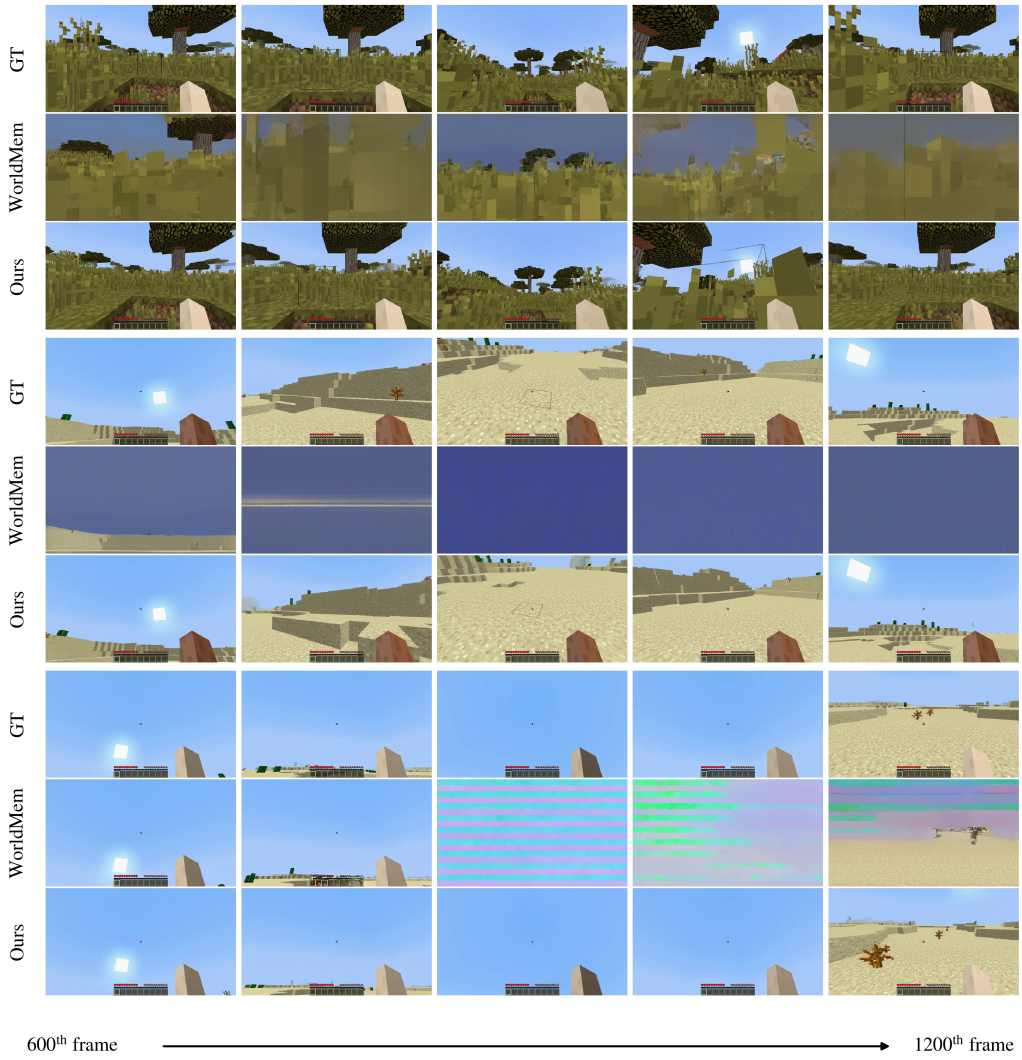


Figure 8: **Additional Qualitative Comparisons on Minecraft.** Three more trajectories were generated from frame 600 to 1200. WorldMem loses the acacia tree, the sandstone terraces, and eventually the whole scene. In the last example, the camera faces the sky for most of the segment, yet W$^4$Mem still recovers the desert scene, which is consistent with the ground truth when it looks down.




Figure 9: **Qualitative Results of 1,000-frame long rollouts on Minecraft.** GT and W$^4$Mem generation results from frame 400 to 1400, twice longer than the main evaluation horizon. Even in such a longer-horizon generation test, W$^4$Mem can still keep the terrain type, horizon, and major landmarks generated mostly aligned with GT to the end.

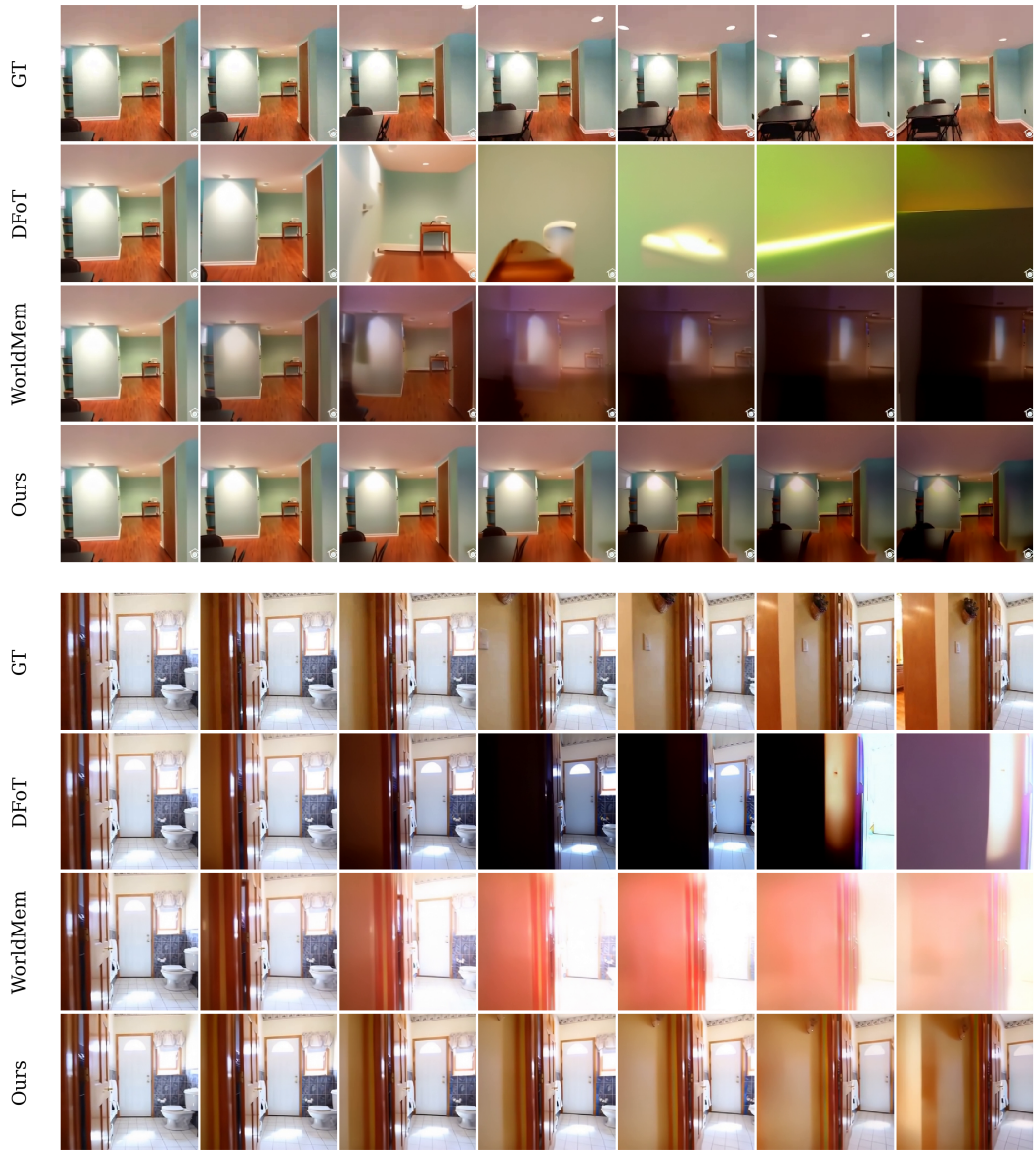


Figure 10: **Additional Qualitative Comparison Results on RealEstate10K.** DFoT drifts into a different room at the beginning, then goes dark, and WorldMem fades into haze gradually to the end. W$^4$Mem keeps the wall colors, lighting, and door frames, and places the dining table (top) close to its GT position.

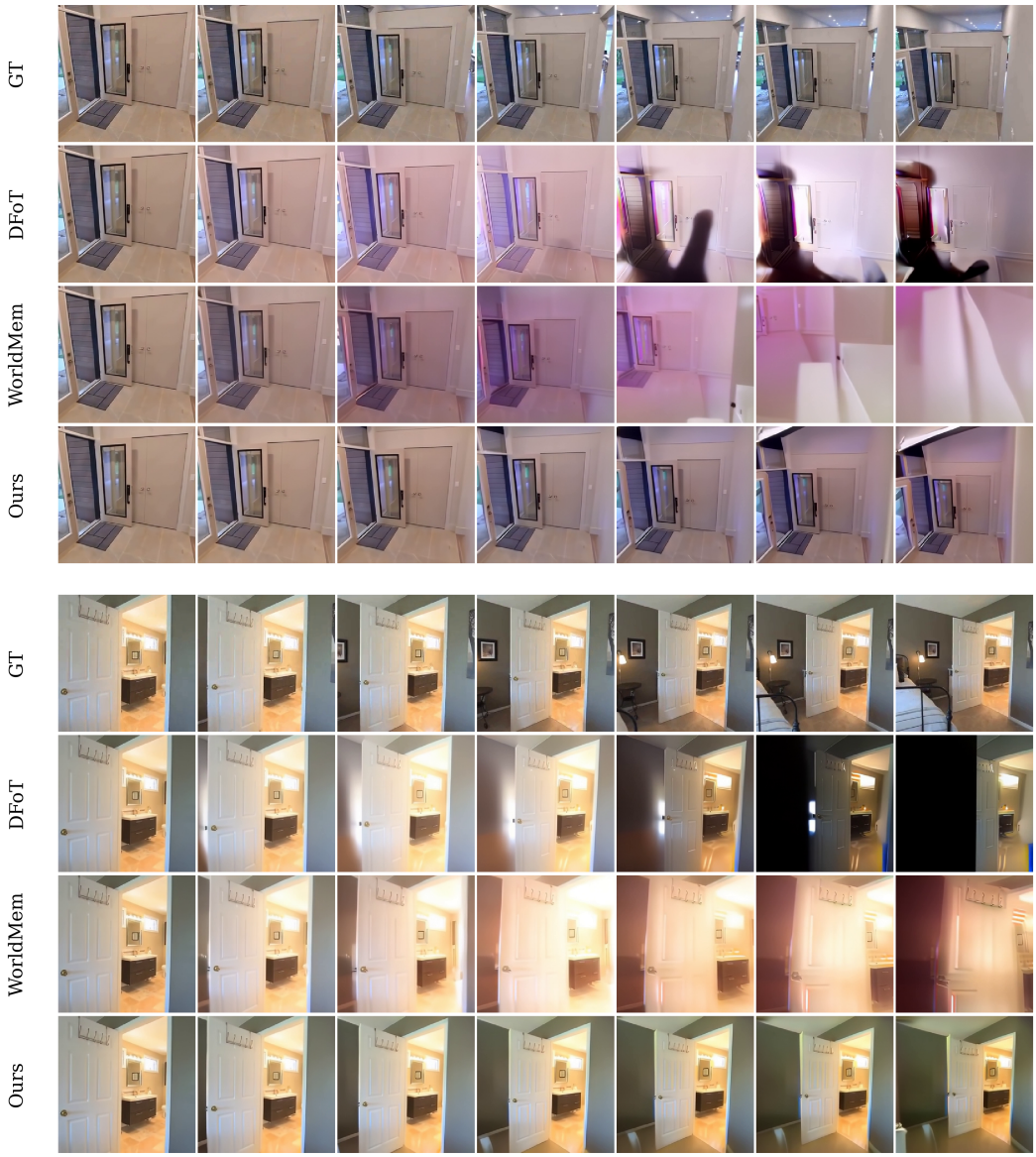


Figure 11: **Additional Qualitative Comparison Results on RealEstate10K.** DFoT introduces dark artifacts and WorldMem washes both scenes out, while W$^4$Mem preserves the front door and doormat (top) and the bathroom vanity (bottom) throughout the sequence.

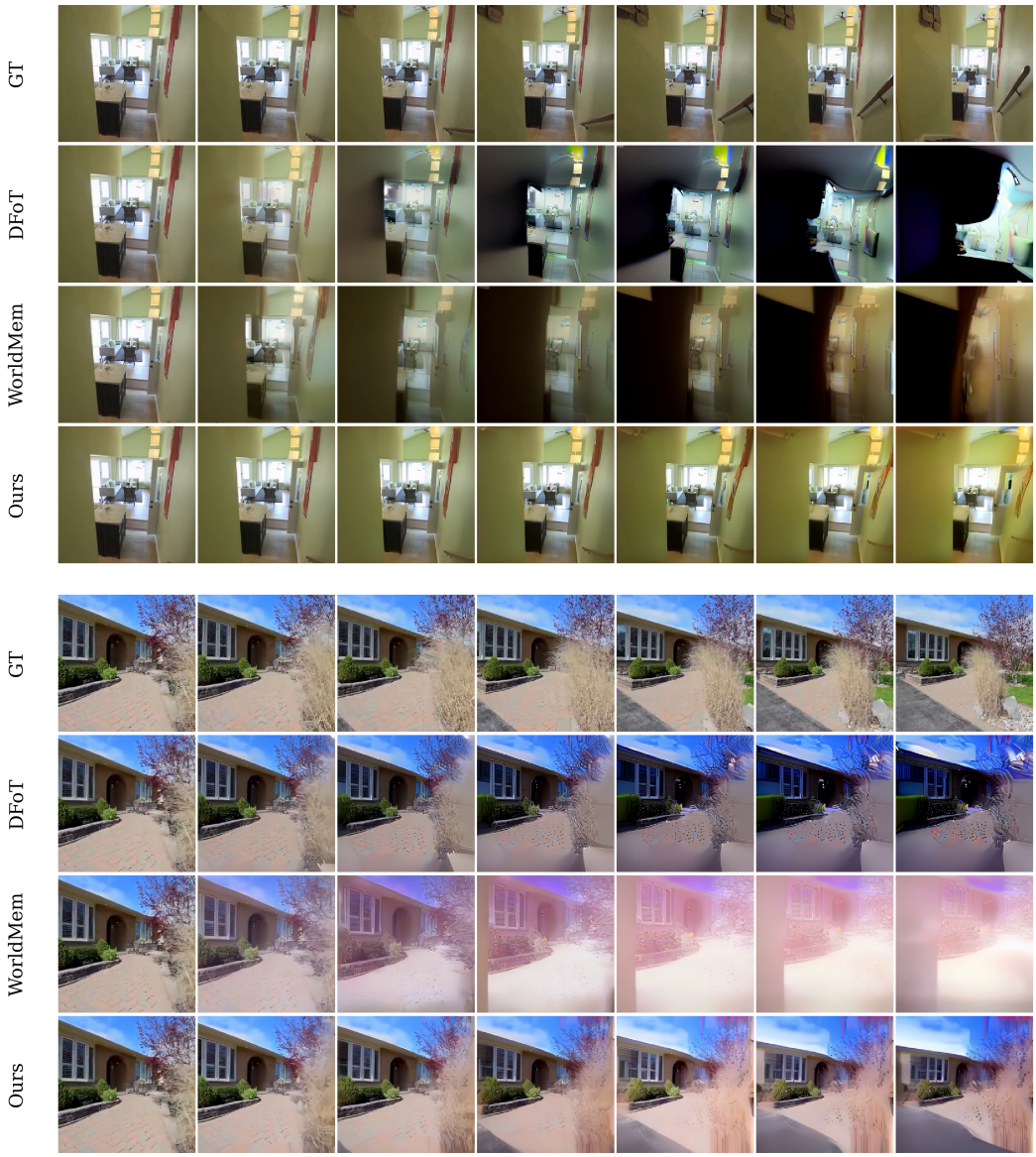


Figure 12: **Additional Qualitative Comparison Results on RealEstate10K.** In the hallway scene(top), DFoT and WorldMem darken until the room is barely visible. In the outdoor scene (bottom), DFoT shifts colors and WorldMem fades to pink, while W$^4$Mem keeps the facade, path, and grass in place.