
WORLD MEM: Long-term Consistent World Simulation with Memory

Zeqi Xiao¹ Yushi Lan¹ Yifan Zhou¹ Wenqi Ouyang¹
Shuai Yang² Yanhong Zeng³ Xingang Pan¹

¹S-Lab, Nanyang Technological University,

²Wangxuan Institute of Computer Technology, Peking University

³Shanghai AI Laboratory

{zeqi001, yushi001, yifan006, wenqi.ouyang, xingang.pan}@ntu.edu.sg
williamyang@pku.edu.cn, zengyh1900@gmail.com

Abstract

World simulation has gained increasing popularity due to its ability to model virtual environments and predict the consequences of actions. However, the limited temporal context window often leads to failures in maintaining long-term consistency, particularly in preserving 3D spatial consistency. In this work, we present WORLD-MEM, a framework that enhances scene generation with a memory bank consisting of memory units that store memory frames and states (*e.g.*, poses and timestamps). By employing state-aware memory attention that effectively extracts relevant information from these memory frames based on their states, our method is capable of accurately reconstructing previously observed scenes, even under significant viewpoint or temporal gaps. Furthermore, by incorporating timestamps into the states, our framework not only models a static world but also captures its dynamic evolution over time, enabling both perception and interaction within the simulated world. Extensive experiments in both virtual and real scenarios validate the effectiveness of our approach. Project page at <https://xizaoqu.github.io/worldmem>.

1 Introduction

World simulation has gained significant attention for its ability to model environments and predict the outcomes of actions (Bar et al., 2024; Decart et al., 2024; Alonso et al., 2025; Feng et al., 2024; Parker-Holder et al., 2024; Valevski et al., 2024). Recent advances in video diffusion models have further propelled this field, enabling high-fidelity rollouts of potential future scenarios based on user actions, such as navigating through an environment or interacting with objects. These capabilities make world simulators particularly promising for applications in autonomous navigation (Feng et al., 2024; Bar et al., 2024) and as viable alternatives to traditional game engines (Decart et al., 2024; Parker-Holder et al., 2024).

Despite these advances, a fundamental challenge remains: the limited probing horizon. Due to computational and memory constraints, video generative models operate within a fixed context window and are unable to condition on the full sequence of past generations. Consequently, most existing methods simply discard previously generated content, leading to a critical issue of world inconsistency, which is also revealed in Wang et al. (2025). As illustrated in Figure 1(a), when the camera moves away and returns, the regenerated content diverges from the earlier scene, violating the coherence expected in a consistent world.

A natural solution is to maintain an external memory that stores and retrieves relevant historical information outside the generative loop. While intuitive, formulating such a memory mechanism is

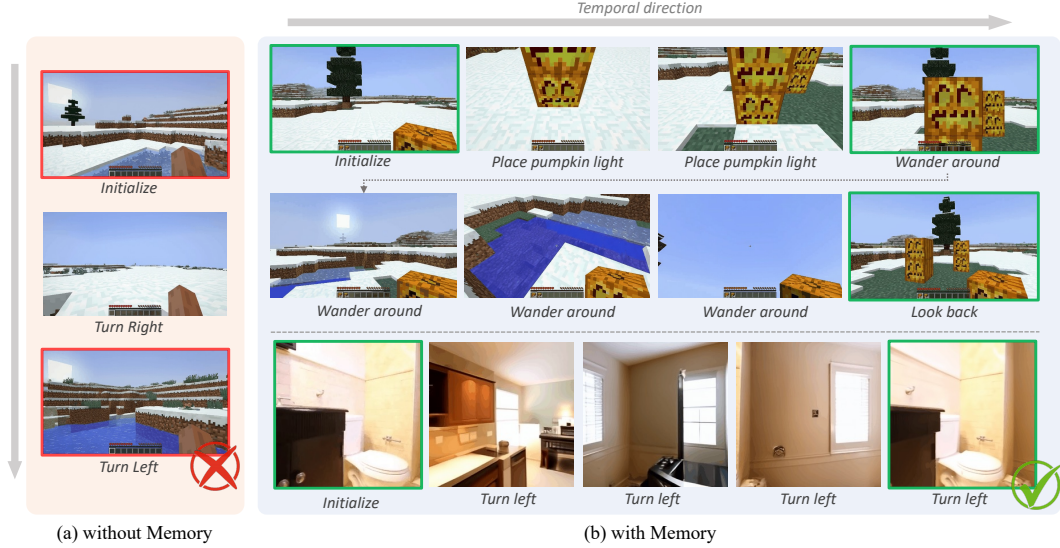


Figure 1: **WORLDMEM enables long-term consistent world generation with an integrated memory mechanism.** (a) Previous world generation methods typically face the problem of inconsistent world due to limited temporal context window size. (b) WORLDMEM empowers the agent to explore diverse and consistent worlds with an expansive action space, *e.g.*, crafting environments by placing objects like pumpkin light or freely roaming around. Most importantly, after exploring for a while and glancing back, we find the objects we placed are still there, with the inspiring sight of the light melting the surrounding snow, testifying to the passage of time. Red and green boxes indicate scenes that should be consistent.

non-trivial. A direct approach might involve explicit 3D scene reconstruction to preserve geometry and detail. However, 3D representations are inflexible in dynamic and evolving environments and are prone to loss of detail, especially for large, unbounded scenes (Wu et al., 2025a).

Instead, we argue that geometry-free representations offer a more flexible solution. These representations, however, pose their own challenges – particularly in balancing detail retention with memory scalability. For example, implicit approaches like storing abstract features via LoRA modules (Hong et al., 2024) offer compactness but lose visual fidelity and spatial specificity. Some recent works represent visual scenes as discrete tokens encoding fine-grained visual information (Sajjadi et al., 2022; Jiang et al., 2025), but they are limited by a fixed token and struggle to capture the complexity of diverse and evolving environments. To address this issue, we observe that for generating the immediate future, only a small subset of historical content is typically relevant. Based on this, we propose a token-level *memory bank* that stores all previously generated latent tokens, and retrieves a targeted subset for each generation step based on relevance.

Conditioning on the retrieved memory requires spatial-temporal reasoning. In contrast to prior work where memory aids local temporal smoothness (Zheng et al., 2024a) or semantic coherence (Wu et al., 2025b; Rahman et al., 2023), long-term world simulation demands reasoning over large spatiotemporal gaps, *e.g.*, memory and query may differ in viewpoint and time, and retain exact scenes with detail. To facilitate this reasoning, we propose augmenting each memory unit with explicit state cues, including spatial location, viewpoint, and timestamp. These cues serve as anchors for reasoning and are embedded as part of the query-key attention mechanism. Through this state-aware attention, our model can effectively reason the current frame with past observations, facilitating accurate and coherent generation. Importantly, such a design leverages standard attention architectures, enabling it to scale naturally with modern hardware and model capacity.

Motivated by this idea, we build our approach, WORLDMEM, on top of the Conditional Diffusion Transformer (CDiT) (Peebles and Xie, 2023) and the Diffusion Forcing (DF) paradigm (Chen et al., 2025), which autoregressively generates first-person viewpoints conditioned on external action signals. As discussed above, at the core of WORLDMEM is a memory mechanism composed of a memory bank and memory attention. To ensure efficient and relevant memory retrieval from the bank, we introduce a confidence-based selection strategy that scores memory units based on field-of-view

(FOV) overlap and temporal proximity. In the memory attention, the latent tokens being generated act as queries, attending to the memory tokens (as keys and values) to incorporate relevant historical context. To ensure robust correspondence across varying viewpoints and time gaps, we enrich both queries and keys with state-aware embeddings. A *relative embedding* design is introduced to ease the learning of spatial and temporal relationships. This pipeline enables precise, scalable reasoning over long-range memory, ensuring consistency in dynamic and evolving world simulations.

We evaluate WORLDMEM on a customized Minecraft benchmark (Fan et al., 2022) and on RealEstate10K (Zhou et al., 2018). The Minecraft benchmark includes diverse terrains (*e.g.*, plains, savannas, and deserts) and various action modalities (movement, viewpoint control, and event triggers), which is a wonderful environment for idea verification. Extensive experiments show that WORLDMEM significantly improves 3D spatial consistency, enabling robust viewpoint reasoning and high-fidelity scene generation, as shown in Figure 1(b). Furthermore, in dynamic environments, WORLDMEM accurately tracks and follows evolving events and environment changes, demonstrating its ability to both perceive and interact with the generated world. We hope our promising results and scalable designs will inspire future research on memory-based world simulation.

2 Related Work

Video diffusion model. With the rapid advancement of diffusion models (Song et al., 2020; Peebles and Xie, 2023; Chen et al., 2025), video generation has made significant strides (Wang et al., 2023a,b; Chen et al., 2023; Guo et al., 2023; OpenAI, 2024; Jin et al., 2024; Yin et al., 2024). The field has evolved from traditional U-Net-based architectures (Wang et al., 2023a; Chen et al., 2023; Guo et al., 2023) to Transformer-based frameworks (OpenAI, 2024; Ma et al., 2024; Zheng et al., 2024b), enabling video diffusion models to generate highly realistic and temporally coherent videos. Recently, autoregressive video generation (Chen et al., 2025; Kim et al., 2024; Henschel et al., 2024) has emerged as a promising approach to extend video length, theoretically indefinitely. Notably, Diffusion Forcing (Chen et al., 2025) introduces a per-frame noise-level denoising paradigm. Unlike the full-sequence paradigm, which applies a uniform noise level across all frames, per-frame noise-level denoising offers a more flexible approach, enabling autoregressive generation.

Interactive world simulation. World simulation aims to model an environment by predicting the next state given the current state and action. This concept has been extensively explored in the construction of world models (Ha and Schmidhuber, 2018b) for agent learning (Ha and Schmidhuber, 2018a; Hafner et al., 2019, 2020; Hu et al., 2023; Beattie et al., 2016; Yang et al., 2023). With advances in video generation, high-quality world simulation with robust control has become feasible, leading to numerous works focusing on interactive world simulation (Bar et al., 2024; Decart et al., 2024; Alonso et al., 2025; Feng et al., 2024; Parker-Holder et al., 2024; Valevski et al., 2024; Yu et al., 2025c,a,b). These approaches enable agents to navigate generated environments and interact with them based on external commands.

However, due to context window limitations, such methods discard previously generated content, leading to inconsistencies in the simulated world, particularly in maintaining 3D spatial coherence.

Consistent world simulation. Ensuring the consistency of a generated world is crucial for effective world simulation Wang et al. (2025). Existing approaches can be broadly categorized into two types: geometric-based and geometric-free. The geometric-based methods explicitly reconstruct the generated world into a 3D/4D representation (Liu et al., 2024; Gao et al., 2024; Wang and Agapito, 2024; Ren et al., 2025; Yu et al., 2024b,a; Liang et al., 2024). While this strategy can reliably maintain consistency, it imposes strict constraints on flexibility: Once the world is reconstructed, modifying or interacting with it becomes challenging. Geometric-free methods focus on implicit learning. Methods like Alonso et al. (2025); Valevski et al. (2024) ensure consistency by overfitting to predefined scenarios (*e.g.*, specific CS:GO or DOOM maps), limiting scalability. StreamingT2V (Henschel et al., 2024) maintains long-term consistency by continuing on both global and local visual contexts from previous frames, while SlowFastGen (Hong et al., 2024) progressively trains LoRA (Hu et al., 2022) modules for memory recall. However, these methods rely on abstract representations, making accurate scene reconstruction challenging. In contrast, our approach retrieves information from previously generated frames and their states, ensuring world consistency without overfitting to specific scenarios.

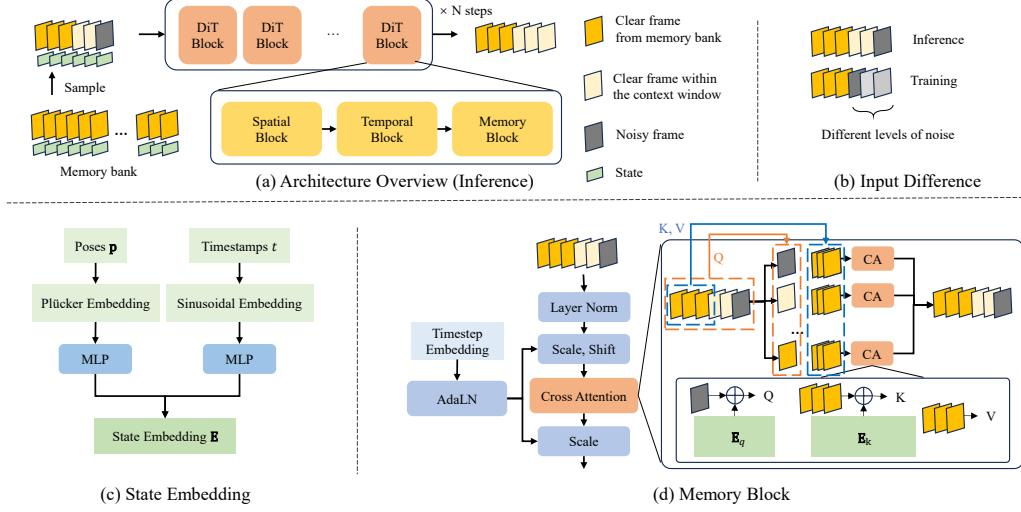


Figure 2: **Comprehensive overview of WORLDMEM.** The framework comprises a conditional diffusion transformer integrated with memory blocks, with a dedicated memory bank storing memory units from previously generated content. By retrieving these memory units from the memory bank and incorporating the information by memory blocks to guide generation, our approach ensures long-term consistency in world simulation.

3 WORLDMEM

This section details the methodology of WORLDMEM. Sec. 3.1 introduces the relevant preliminaries, while Sec. 3.2 describes the interactive world simulator serving as our baseline. Sec. 3.3 and 3.4 present the core of our proposed memory mechanism.

3.1 Preliminary

Video diffusion models. Video diffusion models generate video sequences by iteratively denoising Gaussian noise through a learned reverse process:

$$p_{\theta}(\mathbf{x}_t^{k-1}|\mathbf{x}_t^k) = \mathcal{N}(\mathbf{x}_t^{k-1}; \mu_{\theta}(\mathbf{x}_t^k, k), \sigma_k^2 \mathbf{I}), \quad (1)$$

where all frames $(\mathbf{x}_t^k)_{1 \leq t \leq T}$ share the same noise level k , and T is the context window length. This *full-sequence* approach enables global guidance but lacks flexibility in sequence length and autoregressive generation.

Autoregressive video generation. Autoregressive video generation aims to extend videos over the long term by predicting frames sequentially (Kondratyuk et al., 2024; Wu et al., 2023). While various methods exist for autoregressive generation, Diffusion Forcing (DF) (Chen et al., 2025) provides a neat and effective approach to achieve this. Specifically, DF introduces *per-frame noise levels* k_t :

$$p_{\theta}(\mathbf{x}_t^{k_t-1}|\mathbf{x}_t^{k_t}) = \mathcal{N}(\mathbf{x}_t^{k_t-1}; \mu_{\theta}(\mathbf{x}_t^{k_t}, k_t), \sigma_{k_t}^2 \mathbf{I}), \quad (2)$$

Unlike full-sequence diffusion, DF generates video flexibly and stably beyond the training horizon. Autoregressive generation is a special case when only the last one or a few frames are noisy. With autoregressive video generation, long-term interactive world simulation becomes feasible.

3.2 Interactive World Simulation

Before introducing the memory mechanism, we first present our interactive world simulator, which models long video sequences using an auto-regressive conditional diffusion transformer. Interaction is achieved by embedding external control signals, primarily actions, into the model through dedicated conditioning modules (Parker-Holder et al., 2024; Decart et al., 2024; Yu et al., 2025c).

Following prior work (Decart et al., 2024), we adopt a conditional Diffusion Transformer (DiT) (Peebles and Xie, 2023) architecture for video generation, and Diffusion Forecasting (DF) (Chen et al.,

2025) for autoregressive prediction. As shown in Figure 2(a), our model consists of multiple DiT blocks with spatial and temporal modules for spatiotemporal reasoning. The temporal module applies causal attention to ensure that each frame only attends to preceding frames.

The actions are injected by first projected into the embedding space using a multi-layer perceptron (MLP). The resulting action embeddings are added to the denoising timestep embeddings and injected into the temporal blocks using Adaptive Layer Normalization (AdaLN) (Xu et al., 2019), following the paradigm of Bar et al. (2024); Decart et al. (2024). In our Minecraft experiments, the action space contains 25 dimensions, including movements, view adjustments, and event triggers. We also apply timestep embeddings to the spatial blocks in the same manner, although this is omitted from the figure for clarity. Standard architectural components such as residual connections, multi-head attention, and feedforward networks are also not shown.

The combination of conditional DiT and DF provides a strong baseline for long-term interactive video generation. However, due to the computational cost of video synthesis, the temporal context window remains limited. As a result, content outside this window is forgotten, which leads to inconsistencies during long-term generation (Decart et al., 2024).

3.3 Memory Representation and Retrieval

To address the limited context window of video generative models, we introduce a *memory mechanism* that enables the model to retain and retrieve information beyond the current generation window. This mechanism maintains a *memory bank* composed of historical frames and their associated state information: $\{(\mathbf{x}_i^m, \mathbf{p}_i, t_i)\}_{i=1}^N$, where \mathbf{x}_i^m denotes a memory frame, $\mathbf{p}_i \in \mathbb{R}^5$ (x, y, z, pitch, yaw) is its pose, and t_i is the timestamp. Each tuple is referred to as a *memory unit*. We save \mathbf{m}_i in token-level, which is compressed by the visual encoder but retains enough details for reconstruction. The corresponding states $\{(\mathbf{p}, t)\}$ play a critical role not only in memory retrieval but also in enabling state-aware memory conditioning.

Algorithm 1: Memory Retrieval Algorithm

Input: Memory bank of N historical states

$\{(\mathbf{x}_i^m, \mathbf{p}_i, t_i)\}_{i=1}^N$;

Current state $(\mathbf{x}_c, \mathbf{p}_c, t_c)$; memory condition length L_M ;

Similarity threshold tr ; weights w_o, w_t .

Output: A list of selected state indices S

Compute Confidence Score:

Compute FOV overlap ratio \mathbf{o} via Monte Carlo sampling.

Compute time difference $\mathbf{d} = \text{Concat}(\{|t_i - t_c|\}_{i=1}^n)$.

Compute confidence $\alpha = \mathbf{o} \cdot w_o - \mathbf{d} \cdot w_t$.

Selection with Similarity Filtering:

Initialize $S = \emptyset$

for $m = 1$ **to** L_M **do**

 Select i^* with highest α_{i^*}

 Append i^* to S

 Remove all j where $\text{similarity}(i^*, j) > tr$

return S

Memory Retrieval. Since the number of memory frames available for conditioning is limited, an efficient strategy is required to sample memory units from the memory bank. We adopt a greedy matching algorithm based on frame-pair similarity, where similarity is defined using the field-of-view (FOV) overlap ratio and timestamp differences as confidence measures. Algorithm 1 presents our approach to memory retrieval. Although simple, this strategy proves effective in retrieving relevant information for conditioning. Moreover, the model’s reasoning over memory helps maintain performance even when the retrieved content is imperfect.

3.4 State-aware Memory Condition

After retrieving necessary memory units, unlike prior methods that use memory mainly for temporal smoothness (Zheng et al., 2024a) or semantic guidance (Wu et al., 2025b; Rahman et al., 2023), our goal is to explicitly reconstruct previously seen visual content – even under significant viewpoint or scene changes. This requires the model to perform spatiotemporal reasoning to extract relevant information from memory, which we model using cross-attention (Vaswani et al., 2017). Since relying solely on visual tokens can be ambiguous, we incorporate the corresponding states as cues to enable state-aware attention.

State Embedding. State embedding provides essential spatial and temporal context for memory retrieval. To encode spatial information, we adopt Plücker embedding (Sitzmann et al., 2021) to convert 5D poses $\mathbf{p} \in \mathbb{R}^5$ into dense positional features $\text{PE}(\mathbf{p}) \in \mathbb{R}^{h \times w \times 6}$, following (He et al., 2024; Gao et al., 2024). Temporal context is captured via a lightweight MLP over sinusoidal embedded

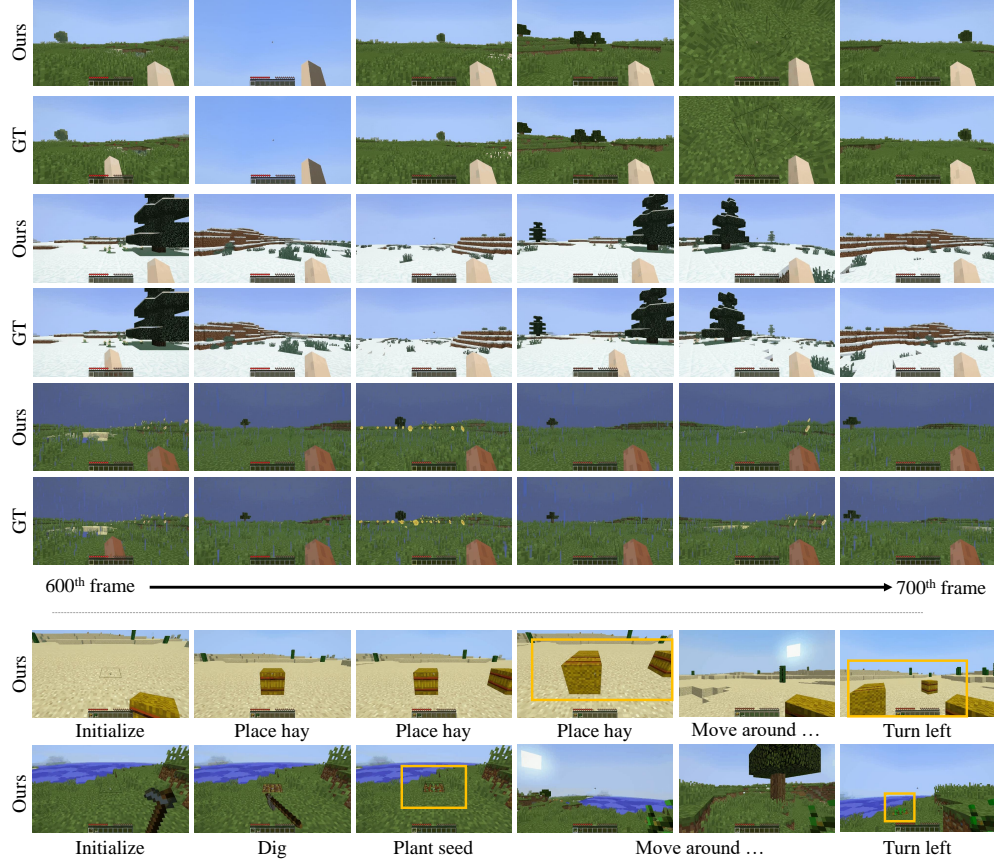


Figure 3: **Qualitative results.** We showcase WORLDMEM’s capabilities through two sets of examples. **Top:** A comparison with Ground Truth (GT). WORLDMEM accurately models diverse dynamics (e.g., rain) by conditioning on 600 past frames, ensuring temporal consistency. **Bottom:** Interaction with the world. Objects like hay in the desert or wheat in the plains persist over time, with wheat visibly growing. For the best experience, see the supplementary videos.

(SE) timestamps. The final embedding is (Figure 2 (c)):

$$\mathbf{E} = G_p(\text{PE}(\mathbf{p})) + G_t(\text{SE}(t)), \quad (3)$$

where G_p and G_t are MLPs mapping pose and time into a shared space.

State-aware Memory Attention. To support reconstruction under viewpoint and temporal shifts, we introduce a state-aware attention mechanism that incorporates spatial-temporal cues into memory retrieval. By conditioning attention on both visual features and state information, the model achieves more accurate reasoning between input and memory.

Let $\mathbf{X}_q \in \mathbb{R}^{l_q \times d}$ denote the flattened feature map of input frames (queries), and $\mathbf{X}_k \in \mathbb{R}^{l_k \times d}$ the concatenated memory features (keys and values). We first enrich both with their corresponding state embeddings \mathbf{E}_q and \mathbf{E}_k :

$$\tilde{\mathbf{X}}_q = \mathbf{X}_q + \mathbf{E}_q, \quad \tilde{\mathbf{X}}_k = \mathbf{X}_k + \mathbf{E}_k. \quad (4)$$

Cross-attention is then applied to retrieve relevant memory content and output updated \mathbf{X}' :

$$\mathbf{X}' = \text{CrossAttn}(Q = p_q(\tilde{\mathbf{X}}_q), K = p_k(\tilde{\mathbf{X}}_k), V = p_v(\mathbf{X}_k)), \quad (5)$$

where p_q , p_k , and p_v are learnable projections.

To simplify the reasoning space, we adopt a *relative state* formulation. For each query frame, the state is set to a zero reference (e.g., the pose is reset to the identity and the timestamp to zero), while the states of key frames are normalized to relative values. This design, illustrated in Figure 2(d), improves alignment under viewpoint changes and simplifies the learning objective.

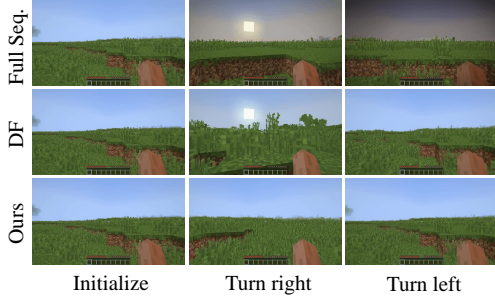


Figure 4: **Within context window evaluation.** The motion sequence involves turning right and returning to the original position, showing self-contained consistency.

Table 1: Evaluation on Minecraft

Within context window			
Methods	PSNR \uparrow	LPIPS \downarrow	rFID \downarrow
Full Seq.	20.35	0.0691	13.87
DF	26.56	0.0094	13.88
Ours	27.01	0.0072	13.73

Beyond context window			
Methods	PSNR \uparrow	LPIPS \downarrow	rFID \downarrow
Full Seq.	/	/	/
DF	18.04	0.4376	51.28
Ours	25.32	0.1429	15.37

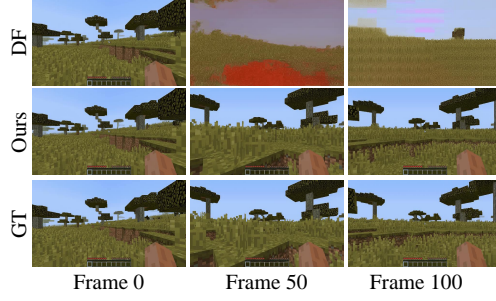


Figure 5: **Beyond context window evaluation.** Diffusion-Forcing suffers inconsistency over time, while ours maintains quality and recovers past scenes.

Table 2: Ablation on embedding designs

Pose type	Embed. type	PSNR \uparrow	LPIPS \downarrow	rFID \downarrow
Sparse	Absolute	20.67	0.2887	39.23
Dense	Absolute	23.63	0.1830	29.34
Dense	Relative	25.32	0.1429	15.37

Table 3: Ablation on memory retrieve strategy

Strategy	PSNR \uparrow	LPIPS \downarrow	rFID \downarrow
Random	18.32	0.3224	47.35
+ Confidence Filter	23.12	0.1863	24.33
+ Similarity Filter	25.32	0.1429	15.37

Incorporating memory into pipeline. We incorporate memory frames into the pipeline by treating them as *clean* inputs during both training and inference. As shown in Figure 2 (a-b), during training, memory frames are assigned the lowest noise level k_{\min} , while context window frames receive independently sampled noise levels from the range $[k_{\min}, k_{\max}]$. During inference, both memory and context frames are assigned k_{\min} , while the current generating frames are assigned k_{\max} .

To restrict memory influence only to memory blocks, we apply a temporal attention mask:

$$A_{\text{mask}}(i, j) = \begin{cases} 1, & i \leq L_M \text{ and } j = i \\ 1, & i > L_M \text{ and } j \leq i \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where L_M is the number of memory frames that are appended before frames within the context window. This guarantees causal attention while preventing memory units from affecting each other.

4 Experiments

Datasets. We use MineDojo (Fan et al., 2022) to create diverse training and evaluation datasets in Minecraft, configuring diverse environments (*e.g.*, plains, savannas, ice plains, and deserts), agent actions, and interactions. For real-world scenes, we utilize RealEstate10K (Zhou et al., 2018) with camera pose annotations to evaluate long-term world consistency.

Metrics. For quantitative evaluation, we employ reconstruction metrics, where the method of obtaining ground truth (GT) varies by specific settings. We then assess the consistency and quality of the generated videos using PSNR, LPIPS (Zhang et al., 2018), and reconstruction FID (rFID) (Heusel et al., 2017), which collectively measure pixel-level fidelity, perceptual similarity, and overall realism.

Experimental details. For our experiments on Minecraft (Fan et al., 2022), we utilize the Oasis (Decart et al., 2024) as the base model. Our model is trained using the Adam optimizer with a fixed



Figure 6: **Results on RealEstate (Zhou et al., 2018).** We visualize loop closure consistency over a full camera rotation. The visual similarity between the first and last frames serves as a qualitative indicator of 3D spatial consistency.

Table 4: Evaluation on RealEstate10K

Methods	PSNR \uparrow	LPIPS \downarrow	rFID \downarrow
CameraCtrl (He et al., 2024)	13.19	0.3328	133.81
TrajAttn (Xiao et al., 2024)	14.22	0.3698	128.36
Viewcrafter (Yu et al., 2024c)	21.72	0.1729	58.43
DFoT (Song et al., 2025)	16.42	0.2933	110.34
Ours	23.34	0.1672	43.14

learning rate of 2×10^{-5} . Training is conducted at a resolution of 640×360 , where frames are first encoded into a latent space via a VAE at a resolution of 32×18 , then further patchified to 16×9 . Our training dataset comprises approximately 12K long videos, each containing 1500 frames, generated from Fan et al. (2022). During training, we employ an 8-frame temporal context window alongside an 8-frame memory window. The model is trained for approximately 500K steps using 4 GPUs, with a batch size of 4 per GPU. For the hyperparameters specified in Algorithm 1 of the main paper, we set the similarity threshold tr to 0.9, w_o to 1, and w_t to $0.2/t_c$. For the noise levels in Eq. (5) and Eq. (6), we set k_{\min} to 15 and k_{\max} to 1000.

For our experiments on RealEstate10K (Zhou et al., 2018), we adopt DFoT (Song et al., 2025) as the base model. The RealEstate10K dataset provides a training set of approximately 65K short video clips. Training is conducted at a resolution of 256×256 , with frames patchified to 128×128 . The model is trained for approximately 50K steps using 4 GPUs, with a batch size of 8 per GPU.

4.1 Results on Generation Benchmark

Comparisons on Minecraft Benchmark. We compare our approach with a standard full-sequence (Full Seq.) training method (He et al., 2024; Wang et al., 2024) and Diffusion Forcing (DF) (Chen et al., 2025). The key differences are as follows: the full-sequence conditional diffusion transformer (Peebles and Xie, 2023) maintains the same noise level during training and inference, DF introduces different noise levels for training and inference, and our method incorporates a memory mechanism. To assess both short-term and long-term world consistency, we conduct evaluations within and beyond the context window. We evaluate both settings on 300 test videos. In the following experiments, the agent’s poses are generated by the game simulator as ground truth. However, in real-world scenarios, only the action input is available, and the pose is not directly observable. In such cases, the next-frame pose can be predicted based on the previous scenes, past states, and the upcoming action. We explore this design choice in the supplementary material.

Within context window. For this experiment, all methods use a context window of 16, while our approach additionally maintains a memory window of 8. We test on customized motion scenarios (e.g., turn left, then turn right or move forward, then backward) to assess self-contained consistency, where the ground truth consists of previously generated frames at the same positions. As shown in Table 1 and Figure 4, the full-sequence baseline suffers from inconsistencies even within its own context window. DF improves consistency by enabling greater information exchange among generated frames. Our memory-based approach achieves the best performance, demonstrating the effectiveness of integrating a dedicated memory mechanism.

Table 5: Ablation on sampling strategy for training

Sampling strategy	PSNR \uparrow	LPIPS \downarrow	rFID \downarrow
Small-range	19.23	0.3786	46.55
Large-range	21.11	0.3855	42.96
Progressive	25.32	0.1429	15.37

Beyond context window. In this setting, all methods use a context window of 8 and generate 100 future frames; our method further employs a memory window of 8 while initializing a 600-frame memory bank. We compute the reconstruction error using the subsequent 100 ground truth frames after 600 frames. Full-sequence methods can not roll out that long so we exclude it. DF exhibits poor PSNR and LPIPS scores, indicating severe inconsistency with the ground truth beyond the context window. Additionally, its low rFID suggests notable quality degradation. In contrast, our memory-augmented approach consistently outperforms others across all metrics, demonstrating superior long-term consistency and quality preservation. Figure 5 further substantiates these findings.

Figure 3 showcases WORLDMEM’s capabilities. The top section demonstrates its ability to operate in a free action space across diverse environments. Given a 600-frame memory bank, our model generates 100 future frames while preserving the ground truth’s actions and poses, ensuring strong world consistency. The bottom section highlights dynamic environment interaction. By using timestamps as embeddings, the model remembers environmental changes and captures natural event evolution, such as plant growth over time.

Comparisons on Real Scenarios. We compare our method with prior works (He et al., 2024; Xiao et al., 2024; Yu et al., 2024c; Song et al., 2025) on the RealEstate10K dataset (Zhou et al., 2018). We design 5 evaluation trajectories, each starting and ending at the same pose, across 100 scenes. The trajectory lengths range from 37 to 60 frames – exceeding the training lengths of all baselines (maximum 25 frames).

CameraCtrl (He et al., 2024), TrajAttn (Xiao et al., 2024), and DFoT (Song et al., 2025) discard past frames and suffer from inconsistency. Viewcrafter (Yu et al., 2024c) incorporates explicit 3D reconstruction, yielding better results, but is constrained by errors in post-processing such as reconstruction and rendering. As shown in Table 4 and Figure 6, our approach achieves superior performance across all metrics. However, the RealEstate dataset inherently limits the full potential of our method, as it consists of short, non-interactive clips with limited temporal complexity. We leave evaluation under more challenging and interactive real-world scenarios for future work.

4.2 Ablation

Embedding designs. The design of embeddings within the memory block is crucial for cross-frame relationship modeling. We evaluate three strategies (Table 2): (1) sparse pose embedding with absolute encoding, (2) dense pose embedding with absolute encoding, and (3) dense pose embedding with relative encoding. Results show that dense pose embeddings (Plücker embedding) significantly enhance all metrics, emphasizing the benefits of richer pose representations. Switching from absolute to relative encoding further improves performance, particularly in LPIPS and rFID, by facilitating relationship reasoning and information retrieval. As illustrated in Figure 7, absolute embeddings accumulate errors over time, while relative embeddings maintain stability even beyond 300 frames.

Sampling strategy for training. We compare different sampling strategies during training in the Minecraft benchmark. Small-range sampling restricts memory conditioning to frames within 2m in the Minecraft world, while large-range sampling extends this range to 8m. Progressive sampling, on the other hand, begins with small-range samples for initial training steps and then gradually expands to large-range samples.

As shown in Table 5, both small-range and large-range sampling struggle with consistency and quality, whereas progressive sampling significantly improves all metrics. This suggests that gradually increasing difficulty during training helps the model learn to reason and effectively query information from memory blocks.

Time condition. We ablate the effectiveness of timestamp embedding in Table 6. We curate 100 video samples featuring placing events and evaluate whether future generations align with event progression. As shown in the table, incorporating time embeddings significantly improves PSNR

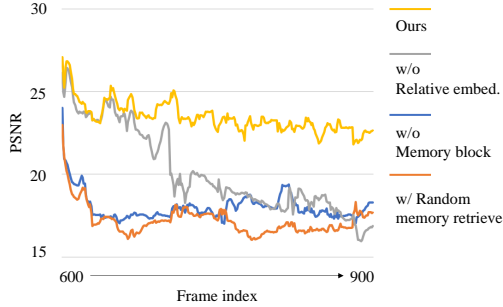


Figure 7: **Long-term Generation Comparison.** This figure presents the PSNR of different ablation methods compared to the ground truth over a 300-frame sequence. The results show that our method without memory blocks or using random memory retrieval exhibits immediate inconsistencies with the ground truth. Additionally, the model lacking relative embeddings begins to degrade significantly beyond 100 frames. In contrast, our full method maintains strong consistency even beyond 300 frames.



Figure 8: **Results w/o and w/ time condition.** Without timestamps, the model fails to differentiate memory units from the same location at different times, causing errors. With time conditioning, it aligns with the updated world state, ensuring consistency.

Table 6: Ablation on time condition

Time embedding	PSNR \uparrow	LPIPS \downarrow	rFID \downarrow
w/o	23.17	0.1989	23.89
w/	25.12	0.1613	16.53

and LPIPS, indicating that adding temporal information helps the model faithfully reproduce event changes in world simulation. Since events like plant growth are inherently unpredictable, we do not conduct quantitative evaluations on such cases but instead provide qualitative illustrations in Figure 8.

Memory retrieve strategy. We analyze memory retrieval strategies in Table 3. Random sampling from the memory bank leads to poor performance and severe quality degradation, as evidenced by a sharp drop in rFID and rapid divergence from the ground truth (Figure 7). The confidence-based filtering significantly enhances consistency and generation quality. Additionally, we refine retrieval by filtering out redundant memory units based on similarity, further improving all evaluation metrics and demonstrating the effectiveness of our approach.

5 Limitations and Future works

Despite the effectiveness of our approach, certain issues warrant further exploration. First, we cannot guarantee that we can always retrieve all necessary information from the memory bank. In some corner cases (*e.g.*, when views are blocked by obstacles), relying solely on view overlap may be insufficient. Second, our current interaction with the environment lacks diversity and realism. In future work, we plan to extend our models to real-world scenarios with more realistic and varied interactions. Lastly, our memory design still entails linearly increasing memory usage, which may impose limitations when handling extremely long sequences.

6 Conclusion

In conclusion, WORLDMEM tackles the longstanding challenge of maintaining long-term consistency in world simulation by employing a memory bank of past frames and associated states. Its memory attention mechanism enables accurate reconstruction of previously observed scenes, even under large viewpoints or temporal gaps, and effectively models dynamic changes over time. Extensive experiments in both virtual and real settings confirm WORLDMEM’s capacity for robust, immersive world simulation. We hope our work will encourage further research on the design and applications of memory-based world simulators.

Acknowledgements. This research is supported by the National Research Foundation, Singapore, under its NRF Fellowship Award <NRF-NRFF16-2024-0003>. This research is also supported by NTU SUG-NAP, as well as cash and in-kind funding from NTU S-Lab and industry partner(s).

References

- Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos J Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2025.
- Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models, 2024.
- Charles Beattie, Joel Z Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, et al. Deepmind lab. *arXiv preprint arXiv:1612.03801*, 2016.
- 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, 2025.
- Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023.
- Decart, Julian Quevedo, Quinn McIntyre, Spruce Campbell, Xinlei Chen, and Robert Wachen. Oasis: A universe in a transformer. 2024. Project website.
- 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.
- Ruili Feng, Han Zhang, Zhantao Yang, Jie Xiao, Zhilei Shu, Zhiheng Liu, Andy Zheng, Yukun Huang, Yu Liu, and Hongyang Zhang. The matrix: Infinite-horizon world generation with real-time moving control. *arXiv preprint arXiv:2412.03568*, 2024.
- Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024.
- Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023.
- David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31, 2018a.
- David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018b.
- Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.
- Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.
- Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024.
- Roberto Henschel, Levon Khachatryan, Daniil Hayrapetyan, Hayk Poghosyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Streamingt2v: Consistent, dynamic, and extendable long video generation from text. *arXiv preprint arXiv:2403.14773*, 2024.
- 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.
- Yining Hong, Beide Liu, Maxine Wu, Yuanhao Zhai, Kai-Wei Chang, Linjie Li, Kevin Lin, Chung-Ching Lin, Jianfeng Wang, Zhengyuan Yang, Ying Nian Wu, and Lijuan Wang. Slowfast-vgen: Slow-fast learning for action-driven long video generation. *arXiv preprint arXiv:2410.23277*, 2024.
- Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.
- Hanwen Jiang, Hao Tan, Peng Wang, Haian Jin, Yue Zhao, Sai Bi, Kai Zhang, Fujun Luan, Kalyan Sunkavalli, Qixing Huang, et al. Rayzer: A self-supervised large view synthesis model. *arXiv preprint arXiv:2505.00702*, 2025.
- Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024.
- Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. FIFO-diffusion: Generating infinite videos from text without training. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, Krishna Somandepalli, Hassan Akbari, Yair Alon, Yong Cheng, Josh Dillon, Agrim Gupta, Meera Hahn, Anja Hauth, David Hendon, Alonso Martinez, David Minnen, Mikhail Sirotenko, Kihyuk Sohn, Xuan Yang, Hartwig Adam, Ming-Hsuan Yang, Irfan Essa, Huisheng Wang, David A. Ross, Bryan Seybold, and Lu Jiang. Videopoet: A large language model for zero-shot video generation, 2024.
- Hanwen Liang, Junli Cao, Vidit Goel, Guocheng Qian, Sergei Korolev, Demetri Terzopoulos, Konstantinos N Plataniotis, Sergey Tulyakov, and Jian Ren. Wonderland: Navigating 3d scenes from a single image. *arXiv preprint arXiv:2412.12091*, 2024.
- Fangfu Liu, Wenqiang Sun, Hanyang Wang, Yikai Wang, Haowen Sun, Junliang Ye, Jun Zhang, and Yueqi Duan. Reconx: Reconstruct any scene from sparse views with video diffusion model. *arXiv preprint arXiv:2408.16767*, 2024.
- Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024.
- OpenAI. Video generation models as world simulators. <https://openai.com/research/video-generation-models-as-world-simulators>, 2024.
- Jack Parker-Holder, Philip Ball, Jake Bruce, Vibhavari Dasagi, Kristian Holsheimer, Christos Kaplanis, Alexandre Moufarek, Guy Scully, Jeremy Shar, Jimmy Shi, Stephen Spencer, Jessica Yung, Michael Dennis, Sultan Kenjeyev, Shangbang Long, Vlad Mnih, Harris Chan, Maxime Gazeau, Bonnie Li, Fabio Pardo, Luyu Wang, Lei Zhang, Frederic Besse, Tim Harley, Anna Mitenkova, Jane Wang, Jeff Clune, Demis Hassabis, Raia Hadsell, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 2: A large-scale foundation world model. 2024.
- William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.
- Tanzila Rahman, Hsin-Ying Lee, Jian Ren, Sergey Tulyakov, Shweta Mahajan, and Leonid Sigal. Make-a-story: Visual memory conditioned consistent story generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2493–2502, 2023.
- Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.
- Mehdi SM Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lučić, Daniel Duckworth, Alexey Dosovitskiy, et al. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6229–6238, 2022.
- Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. *Advances in Neural Information Processing Systems*, 34:19313–19325, 2021.
- Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. *arXiv preprint arXiv:2502.06764*, 2025.

- Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.
- Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837*, 2024.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.
- Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. *arXiv preprint arXiv:2408.16061*, 2024.
- Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023a.
- Jing Wang, Fengzhuo Zhang, Xiaoli Li, Vincent YF Tan, Tianyu Pang, Chao Du, Aixin Sun, and Zhuoran Yang. Error analyses of auto-regressive video diffusion models: A unified framework. *arXiv preprint arXiv:2503.10704*, 2025.
- Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *arXiv preprint arXiv:2309.15103*, 2023b.
- Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.
- Sibo Wu, Congrong Xu, Binbin Huang, Andreas Geiger, and Anpei Chen. Genfusion: Closing the loop between reconstruction and generation via videos. *arXiv preprint arXiv:2503.21219*, 2025a.
- Tong Wu, Zhihao Fan, Xiao Liu, Yeyun Gong, Yelong Shen, Jian Jiao, Hai-Tao Zheng, Juntao Li, Zhongyu Wei, Jian Guo, Nan Duan, and Weizhu Chen. Ar-diffusion: Auto-regressive diffusion model for text generation, 2023.
- Xindi Wu, Uriel Singer, Zhaojiang Lin, Andrea Madotto, Xide Xia, Yifan Xu, Paul Crook, Xin Luna Dong, and Seungwhan Moon. Corgi: Cached memory guided video generation. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4585–4594. IEEE, 2025b.
- Zeqi Xiao, Wenqi Ouyang, Yifan Zhou, Shuai Yang, Lei Yang, Jianlou Si, and Xingang Pan. Trajectory attention for fine-grained video motion control. *arXiv preprint arXiv:2411.19324*, 2024.
- Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. Understanding and improving layer normalization. *Advances in neural information processing systems*, 32, 2019.
- Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 1(2):6, 2023.
- Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast causal video generators. *arXiv preprint arXiv:2412.07772*, 2024.
- Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T Freeman, and Jiajun Wu. Wonderworld: Interactive 3d scene generation from a single image. *arXiv preprint arXiv:2406.09394*, 2024a.
- Hong-Xing Yu, Haoyi Duan, Junhwa Hur, Kyle Sargent, Michael Rubinstein, William T Freeman, Forrester Cole, Deqing Sun, Noah Snively, Jiajun Wu, et al. Wonderjourney: Going from anywhere to everywhere. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6658–6667, 2024b.
- Jiwen Yu, Yiran Qin, Haoxuan Che, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, Kun Gai, Hao Chen, and Xihui Liu. A survey of interactive generative video. *arXiv preprint arXiv:2504.21853*, 2025a.
- Jiwen Yu, Yiran Qin, Haoxuan Che, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Position: Interactive generative video as next-generation game engine. *arXiv preprint arXiv:2503.17359*, 2025b.
- Jiwen Yu, Yiran Qin, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Gamefactory: Creating new games with generative interactive videos. *arXiv preprint arXiv:2501.08325*, 2025c.

- Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024c.
- Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.
- Longtao Zheng, Yifan Zhang, Hanzhong Guo, Jiachun Pan, Zhenxiong Tan, Jiahao Lu, Chuanxin Tang, Bo An, and Shuicheng Yan. Memo: Memory-guided diffusion for expressive talking video generation. *arXiv preprint arXiv:2412.04448*, 2024a.
- Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all, 2024b.
- Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *SIGGRAPH*, 2018.

7 Supplementary Materials

7.1 Details and Experiments

Embedding designs. We present the detailed designs of embeddings for timesteps, actions, poses, and timestamps in Figure 10, where F, C, H, W, A denote the frame number, channel count, height, width, and action count, respectively.

The input pose is parameterized by position (x, z, y) and orientation (pitch θ and yaw ϕ). The extrinsic matrix $\mathbf{T} \in \mathbb{R}^{4 \times 4}$ is formed as:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R}_c & \mathbf{c} \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad (7)$$

where $\mathbf{c} = (x, z, y)^T$ and $\mathbf{R}_c = \mathbf{R}_y(\phi)\mathbf{R}_x(\theta)$.

To encode camera pose, we adopt the Plücker embedding. Given a pixel (u, v) with normalized camera coordinates:

$$\boldsymbol{\pi}_{uv} = \mathbf{K}^{-1}[u, v, 1]^T, \quad (8)$$

its world direction is:

$$\mathbf{d}_{uv} = \mathbf{R}_c \boldsymbol{\pi}_{uv} + \mathbf{c}. \quad (9)$$

The Plücker embedding is:

$$\mathbf{l}_{uv} = (\mathbf{c} \times \mathbf{d}_{uv}, \mathbf{d}_{uv}) \in \mathbb{R}^6. \quad (10)$$

For a frame of size $H \times W$, the full embedding is:

$$\mathbf{L}_i \in \mathbb{R}^{H \times W \times 6}. \quad (11)$$

Memory context length. We evaluate how different memory context lengths affect performance in the Minecraft benchmark. Table 7 shows that increasing the context length from 1 to 8 steadily boosts PSNR, lowers LPIPS, and reduces rFID. However, extending the length to 16 deteriorates results, indicating that excessive memory frames may introduce noise or reduce retrieval precision. A context length of 8 provides the best trade-off, yielding the highest PSNR and the lowest LPIPS and rFID.

Pose prediction. For interactive play, ground truth poses are not accessible. To address this, we designed a lightweight pose prediction module that estimates the pose of the next frame. As illustrated in Figure 9, the predictor takes the previous image, the previous pose, and the upcoming action as inputs and outputs the predicted next pose. This module enables the system to operate using actions alone, eliminating the need for ground truth poses during inference. In Table 8, we compare the performance of using predicted poses versus ground truth poses. While using ground truth poses yields better results across all metrics, the performance drop with predicted poses is acceptable. This is because our method does not rely heavily on precise pose predictions – new frames are generated based on these predictions – and the ground truth poses generated by the Minecraft simulator also contain a certain degree of randomness.

Table 7: Ablation on length of memory context length

Length	PSNR \uparrow	LPIPS \downarrow	rFID \downarrow
1	22.18	0.1899	20.47
4	24.68	0.1568	16.54
8	25.32	0.1429	15.37
16	23.14	0.1687	18.33

Table 8: Comparison between using predicted poses and ground truth poses

Pose Type	PSNR \uparrow	LPIPS \downarrow	rFID \downarrow
Ground truth	25.32	0.1429	15.37
Predicted	23.13	0.1786	20.36

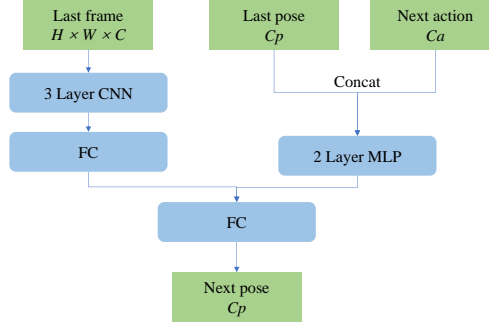


Figure 9: Structure of pose predictor.

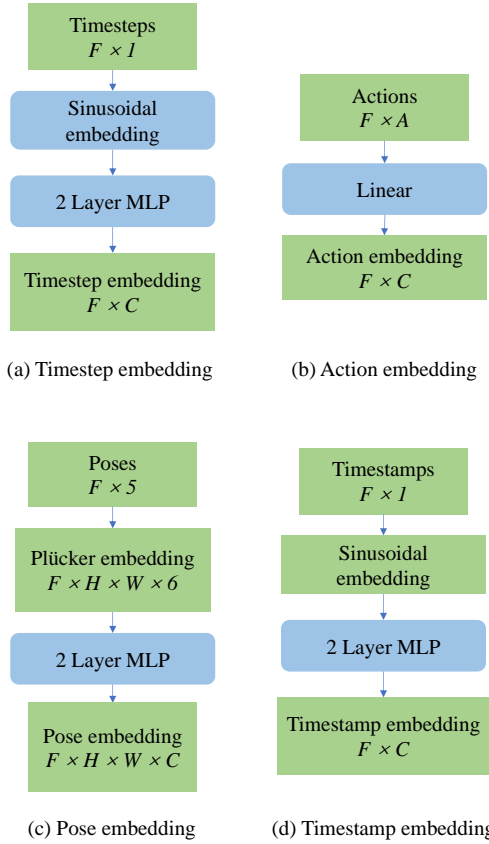


Figure 10: Illustration of different embeddings.

7.2 Memory Usage and Scalability Analysis

To assess the scalability and practical feasibility of our method, we provide detailed quantitative analysis covering memory usage, generation duration, training cost, and inference efficiency.

Memory Usage of the Memory Bank. The memory bank is lightweight. Storing 600 visual memory tokens with shape $[600, 16, 18, 32]$ in `float32` takes approximately **21MB**.

Retrieval Latency. Below we report the average retrieval time (for 8 memory frames) as a function of memory bank size:

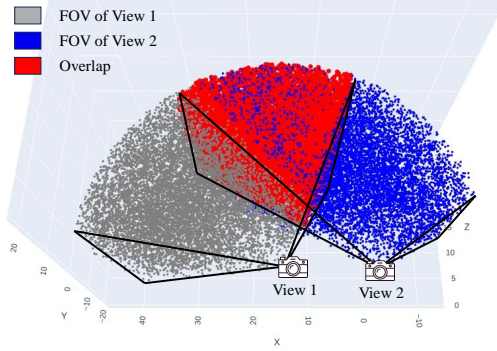


Figure 11: Two-view FOV overlapping visualization.

Number of Memory Candidates	Retrieval Time (s)
10	0.04
100	0.06
600	0.10
1000	0.16

The generation cost (20 denoising steps) is ~ 0.9 s per frame. Retrieval time accounts for only 10–20% of total inference time even with 1000 candidates.

Comparison with Baseline. We compare our method with a baseline model (without memory), under consistent settings: 8 context frames, 8 memory frames, 20 denoising steps, and no acceleration techniques, on single H200.

Method	Training		Inference	
	Mem. Usage	Speed (it/s)	Mem. Usage	Speed (it/s)
w/o Memory	33 GB	3.19	9 GB	1.03
with Memory	51 GB	1.76	11 GB	0.89

Adding memory introduces moderate training overhead. During inference, the impact is minimal: only a small increase in memory usage and a slight decrease in speed.

Inference Optimization. With modern acceleration techniques (e.g., timestep distillation, early exit, sparse attention), inference speed can reach ~ 10 FPS, making our method practical for deployment.

FOV Overlapping Computation. We present the details of Monte Carlo-based FOV overlapping computation in Alg. 11, as well as the two-view overlapping sampling in Figure 11.

7.3 Visualizations

In this section, we provide more visualization of different aspects to facilitate understanding.

Minecraft Training Examples. We present a diverse set of training environments that include various terrain types, action spaces, and weather conditions, as shown in Figure 12. These variations help enhance the model’s adaptability and robustness in different scenarios.

Trajectory Examples in Minecraft. Figure 13 illustrates trajectory examples in the x-z space over 100 frames. The agent’s movement exhibits a random action pattern, ensuring diverse learning objectives and a broad range of sampled experiences.

Pose Distribution. We collect and visualize 800 samples within a sampling range of 8, as shown in Figure 14. The random pattern observed in Figure 14 ensures a diverse distribution of sampled poses in space, which is beneficial for learning the reasoning process within the memory blocks.

Algorithm 2: Monte Carlo-based FOV Overlap Computation (Notationally Disjoint)

Input:

- $Q_{\text{ref}} \in \mathbb{R}^{F \times 5}$: reference poses from memory bank (x,y,z,pitch,yaw), F is the number of stored poses.
- $Q_{\text{tgt}} \in \mathbb{R}^5$: pose of the current (target) frame.
- M : number of 3D sample points (default 10,000).
- R : radius of the sampling sphere (default 30 m).
- ϕ_h, ϕ_v : horizontal/vertical field-of-view angles (in degrees).

Output:

- $\rho \in \mathbb{R}^F$: overlapping ratios between each reference pose and the target pose.

begin**Δ Step 1: Random Sampling in a Sphere**

Generate M points \mathbf{q} uniformly in a 3D sphere of radius R :

$$\mathbf{q} \leftarrow \text{PointSampling}(M, R).$$

Δ Step 2: Translate Points to Q_{tgt} as Center

Let $Q_{\text{tgt}}(x, y, z)$ be the 3D coordinates of the current camera pose. Shift all sampled points:

$$\mathbf{q} \leftarrow \mathbf{q} + Q_{\text{tgt}}(x, y, z).$$

Δ Step 3: FOV Checks

Compute a boolean matrix $\mathbf{v}_{\text{ref}} \in \{0, 1\}^{F \times M}$, where each entry indicates if a point in \mathbf{q} lies in the FOV of a reference pose:

$$\mathbf{v}_{\text{ref}} \leftarrow \text{IsInsideFOV}(\mathbf{q}, Q_{\text{ref}}, \phi_h, \phi_v).$$

Similarly, compute a boolean vector $\mathbf{v}_{\text{tgt}} \in \{0, 1\}^M$ for the target pose:

$$\mathbf{v}_{\text{tgt}} \leftarrow \text{IsInsideFOV}(\mathbf{q}, Q_{\text{tgt}}, \phi_h, \phi_v).$$

Δ Step 4: Overlapping Ratio Computation

Obtain the final overlapping ratio vector $\rho \in \mathbb{R}^F$ by combining \mathbf{v}_{ref} and \mathbf{v}_{tgt} . For instance,

$$\rho[i] = \frac{1}{M} \sum_{j=1}^M (\mathbf{v}_{\text{ref}}[i, j] \cdot \mathbf{v}_{\text{tgt}}[j]),$$

to measure the fraction of sampled points that are visible in both the i -th reference pose and the target pose.

Return ρ .

end

More Qualitative Results. For additional qualitative examples, we recommend consulting the attached web page, which offers enhanced visualizations.

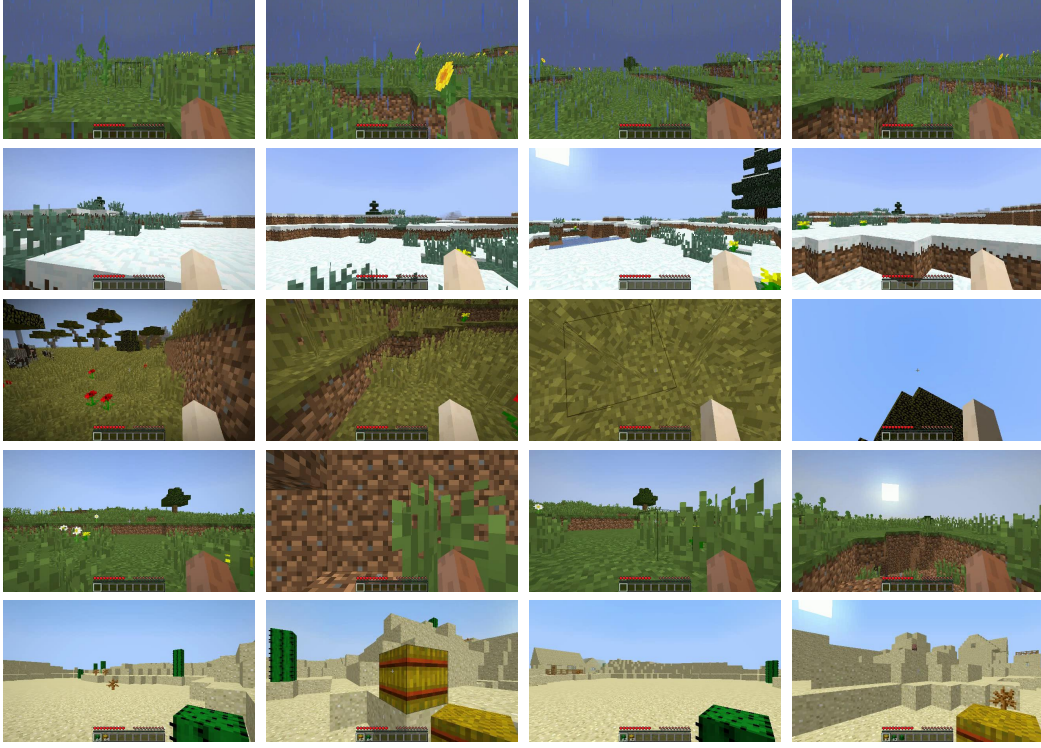


Figure 12: **Training Examples.** Our training environments encompass diverse terrains, action spaces, and weather conditions, providing a comprehensive setting for learning.

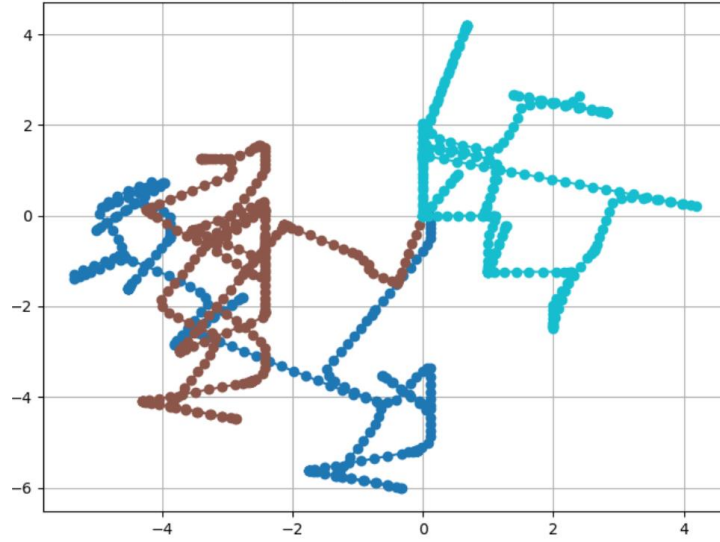


Figure 13: **Visualization of Trajectory Examples in the X-Z Space.** The axis scales represent distances within the Minecraft environment.

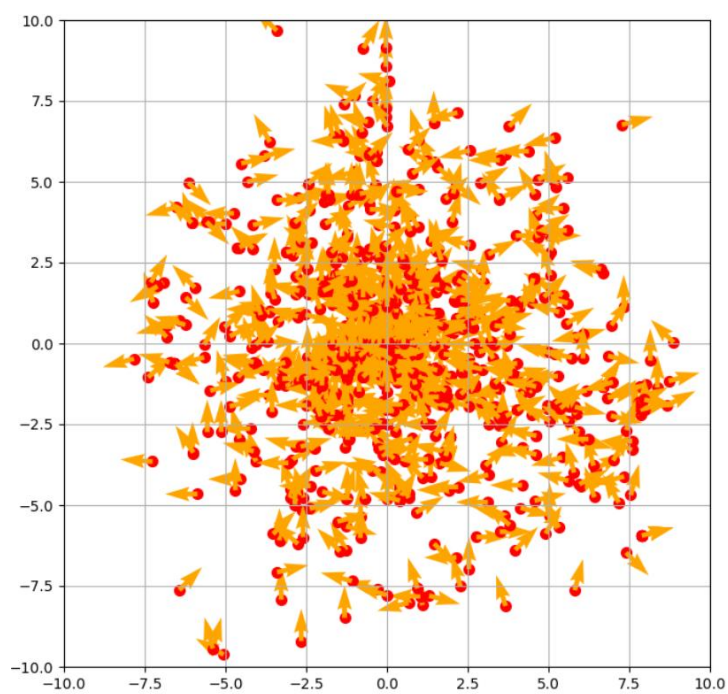


Figure 14: **Visualization of Relative Pose Distribution for Training in X-Z Space.** Red dots indicate positions, while yellow arrows represent directions.