Memory forcing: Spatio-temporal Memory for Long-Horizon Interactive Video Generation

Junchao Huang[1,2] **Xinting Hu**[3] **Boyao Han**[1]
Shaoshuai Shi[4] **Zhuotao Tian**[2] **Tianyu He**[5] **Li Jiang**[1,2*]

[1]The Chinese University of Hong Kong, Shenzhen
[2]Shenzhen Loop Area Institute [3]The University of Hong Kong
[4]Voyager Research, Didi Chuxing [5]Microsoft Research

Abstract

Autoregressive video diffusion models have demonstrated effectiveness in interactive game generation, with Minecraft gameplay serving as a representative application. To faithfully simulate gameplay, a model must generate natural content when exploring new scenes while preserving spatial consistency when revisiting explored areas. Under limited computation budgets, it must compress and exploit historical cues within a finite context window, which exposes a trade-off: models relying solely on temporal context offer flexible exploration but suffer from poor revisit consistency, whereas adding spatial memory strengthens consistency but may degrade new scene generation quality when the model over-relies on sparse or unreliable spatial context. We present Memory Forcing, a learning framework that pairs training protocols with a geometry-indexed spatial memory. Specifically, our Hybrid Training exposes the model to both exploration and revisitation regimes, guiding the model to rely on temporal memory in new scenes while effectively incorporating spatial memory upon revisits. Chained Forward Training creates larger pose variations and encourages reliance on spatial memory for maintaining consistency. For spatial memory, we integrate Point-to-Frame Retrieval with an incremental 3D cache generated by VGGT, enabling constant-time retrieval of relevant historical context regardless of sequence length. Extensive experiments demonstrate that Memory Forcing achieves superior performance in both long-term spatial consistency and visual quality in new scenes with real-time interaction.

1 Introduction

Autoregressive video models [3, 31] based on diffusion [7, 19, 29] have recently emerged as powerful tools for world modeling, showing strong capabilities in interactive scene generation [9, 28], particularly in open-world environments like Minecraft, where multi-dimensional controls enable rich user interactions. These models [5, 6, 10] learn to predict future frames conditioned on past observations and user actions, enabling autoregressive (AR) rollouts that react to player inputs in real time. Within the AR paradigm, the model must condition on a context window of past frames, but latency and memory limits bound the window size. Therefore, it is critical to compress and prioritize historical information (*i.e.*, memory) within this limited window.

In prior works, memory allocation strategies exhibit two characteristic failure modes, as shown in Fig. 1. Models that over-rely on spatial memory preserve consistency on revisits but degrade in novel scene exploration where spatial context is insufficient (Fig. 1(a)). Conversely, temporal-only models enable effective exploration but fail to maintain spatial consistency (Fig. 1(b)). Moreover, teacher-

*Corresponding author.

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

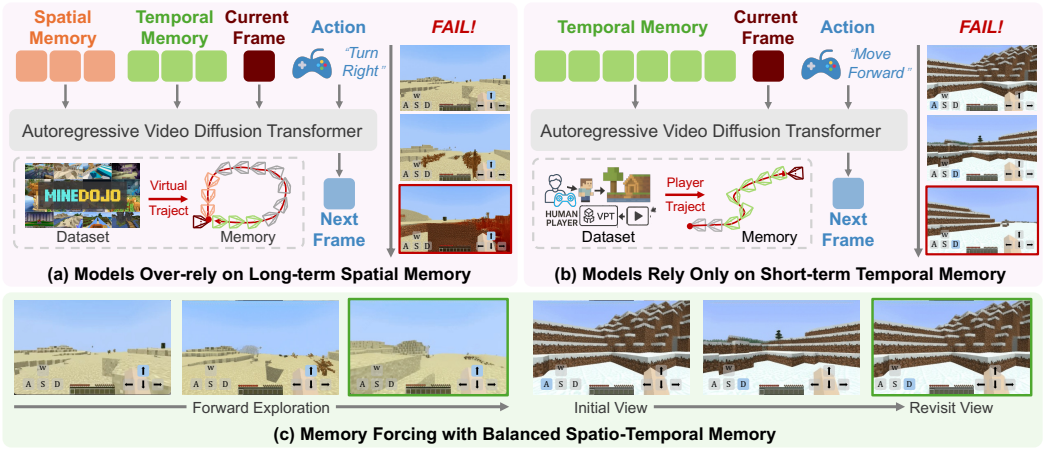

Figure 1: *Memory Forcing* balances temporal context and geometry-indexed spatial memory through Hybrid Training and Chained Forward Training. (a) Previous methods over-relying on long-term spatial memory maintain revisit consistency but fail in new scene exploration. (b) Methods relying only on temporal memory enable effective exploration but lack spatial consistency. (c) Our method excels in both scenarios, maintaining high-quality exploration while ensuring spatial consistency.

forced training [22] underestimates inference-time drift, encouraging over-reliance on short-horizon temporal cues and underuse of spatial memory at test time. These observations motivate a training framework that enables the model to modulate its reliance on temporal and spatial memory across exploration and revisit regimes, thereby balancing exploration flexibility and revisit consistency.

To address these limitations, we introduce Memory Forcing, a training framework that enables the model to flexibly and effectively use temporal and spatial memory under a fixed window. Specifically, Hybrid Training uses distinct data distributions to emulate complementary gameplay regimes, so the model learns to rely on temporal memory for novel-scene exploration and to incorporate spatial memory on revisits for consistency. In practice, we adopt temporal-only conditioning on VPT [1] (human play, exploration-oriented) and spatial&temporal conditioning on MineDojo [8] (simulated trajectories with frequent revisits and adjacent viewpoints), achieving a balanced optimum across the two tasks. Besides, we introduce Chained Forward Training to augment autoregressive learning with model rollouts: it progressively replaces ground-truth temporal context with the model's own predictions, amplifies pose/viewpoint drift across windows, and thus encourages reliance on spatial memory to maintain revisit consistency.

Beyond the training protocol, we equip the model with Geometry-indexed Spatial Memory. Prior frame-level retrieval [4, 42] is appearance-based, sensitive to viewpoint and illumination changes, and prone to accumulating near-duplicate views under neighboring poses. As sequences grow, redundancy and lookup latency grow with the size of the memory bank. State-space methods [30] compress history into latent states and alleviate this efficiency issue, but they lack explicit spatial indexing, making it difficult to specify which spatial evidence to retain and which redundancy to discard. Instead, we maintain a coarse scene representation via streaming 3D reconstruction and retrieve history with point-to-frame mapping: currently visible 3D points are back-traced to their source frames to select a compact, pose-relevant set of views. This geometry-anchored access is robust to viewpoint changes, bounds the candidate set (top-k), and scales with visible spatial coverage rather than sequence length.

We conduct comprehensive experiments on Minecraft benchmark [8] across three critical dimensions: long-term memory with spatial revisitations, generalization on unseen terrains, and generation performance in new environments. Our method achieves superior performance across all three settings compared to both temporal-only and spatial memory baselines, while our Geometry-indexed Spatial Memory demonstrates 7.3× faster retrieval speed with 98.2% less memory storage. These results demonstrate that Memory Forcing effectively resolves the trade-off between spatial consistency and generative quality while maintaining computational efficiency.

In summary, our contributions are threefold:

- We introduce Memory Forcing, a training framework that enables video diffusion models to balance exploration and revisit consistency for autoregressive video generation in Minecraft under a fixed context budget.
- We develop the Hybrid Training and Chained Forward Training strategy that teaches models to use temporal memory for exploration and incorporate spatial memory for revisits, and a Geometry-indexed Spatial Memory built via streaming 3D reconstruction with Point-to-Frame Retrieval for efficient memory retrieval.
- Extensive experiments demonstrate that Memory Forcing achieves superior performance in both spatial consistency and generative quality in new environments, while maintaining computational efficiency for extended sequences.

2 Related Works

Autoregressive Video Models. Autoregressive video generation [16, 18, 25, 32, 40, 43, 46] enables video synthesis by conditioning on preceding frames. Early token-based approaches [23, 40] achieved temporal consistency but compromised visual fidelity. Recent diffusion-based methods [3, 20, 31, 34] achieve superior quality through masked conditioning and per-frame noise control.

Interactive World Model. World models predict future states from current states and actions [12–15]; recent video generation advances extend them to interactive gaming environments [2, 9, 17, 27, 28, 33, 48, 49], including Minecraft game world models [4–6, 10, 30, 42]. MineWorld [10], NFD [5], and LIVE [21] provide strong interaction but lack long-term memory; state-space approaches [30] remain bounded by training context; and WorldMem [42] uses pose-based retrieval for long-term memory but suffers from limited novel scene generation and lacks real-time interactivity.

3D Reconstruction and Memory Retrieval. Learning-based 3D reconstruction has progressed from DUSt3R [37] to multi-view extensions [35, 38, 45], streaming methods [36, 41], and SLAM-style systems [26] for sequential and long-sequence processing. For memory retrieval in video generation, existing approaches range from pose-based methods [42, 47] using field-of-view overlap to 3D geometry-based approaches [24] with surfel-indexed view selection.

3 Memory Forcing

We introduce Memory Forcing, a learning framework that pairs training protocols with geometry-indexed spatial memory to enable long-term spatial consistency. Our approach addresses the fundamental trade-off between temporal memory for generation and spatial memory for revisits through Hybrid Training and Chained Forward Training (CFT). We next describe the preliminaries, memory-augmented architecture, training protocol, and geometry-indexed memory system.

3.1 Preliminaries

Autoregressive Video Diffusion Models. Autoregressive Video Diffusion Models generate video sequences by predicting future frames conditioned on past observations. Following Diffusion Forcing [3], we denote a video sequence as $X^{1:T} = x_1, x_2, \ldots, x_T$ where each frame x_t is assigned an independent noise level $k_t \in [0, 1]$. The model learns to predict noise $\epsilon_\theta(\tilde{X}^{1:T}, k^{1:T})$ where $\tilde{X}^{1:T}$ represents the noisy sequence and $k^{1:T} = k_1, k_2, \ldots, k_T$ are the noise levels. The training objective minimizes:

$$\mathcal{L} = \mathbb{E}_{k^{1:T}, X^{1:T}, \epsilon^{1:T}} \left[|\epsilon^{1:T} - \epsilon_\theta(\tilde{X}^{1:T}, k^{1:T})|^2 \right] \tag{1}$$

This enables flexible autoregressive conditioning with arbitrary clean/noisy frame combinations.

Interactive Game World Model. Interactive game environments present unique challenges for video generation models. Players navigate complex 3D environments where actions $\mathcal{A}^{1:T}$ include movement commands, camera rotations, and object interactions that directly influence both immediate visual changes and long-term environment state evolution. For action-conditioned generation, the model predicts noise conditioned on both visual observations and actions: $\epsilon_\theta(\tilde{X}^{1:T}, k^{1:T}, \mathcal{A}^{1:T})$ enabling the model to generate coherent video sequences that respond to player inputs.

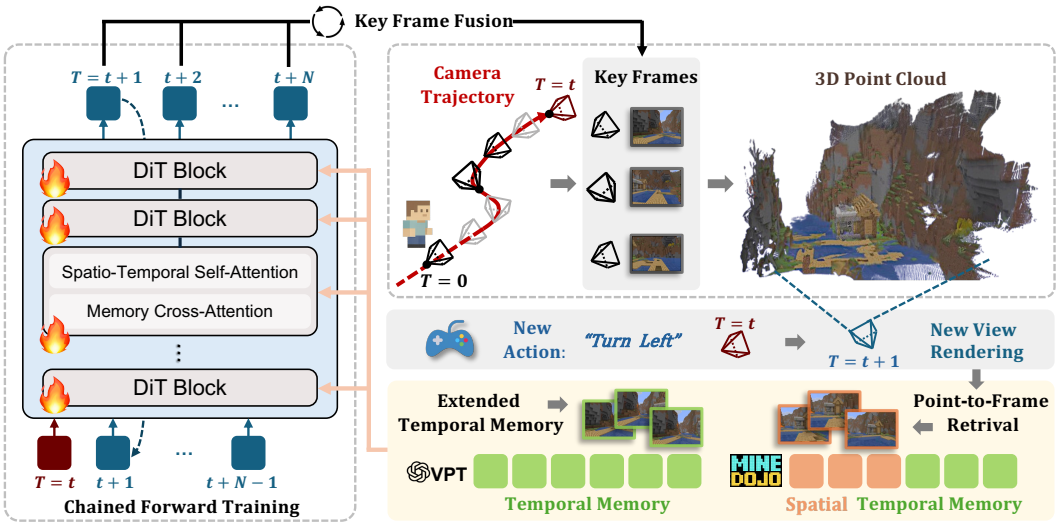

Figure 2: Memory Forcing combines temporal context with geometry-indexed spatial memory for video generation. 3D geometry is maintained through streaming reconstruction along the camera trajectory. During generation, Point-to-Frame Retrieval maps spatial context to historical frames, which are integrated with temporal memory and injected together via memory cross-attention in the DiT backbone. Chained Forward Training (CFT) creates larger pose variations, encouraging the model to effectively utilize spatial memory for maintaining long-term geometric consistency.

3.2 Memory-Augmented Architecture

As illustrated in Figure 2, We follow previous works [5] by adopting Spatio-Temporal Self-Attention for efficient modeling, adaLN-zero conditioning for action integration, and 3D positional embeddings within a Diffusion Transformer (DiT) Backbone. To incorporate long-term spatial memory into the generation process, we introduce several memory-specific components:

Spatial Memory Extraction. We employ the VGGT [35] network with our cross-window scale alignment to enable streaming reconstruction. Historical frames are then efficiently extracted via Point-to-Frame Retrieval (Section 3.4) to provide long-term spatial memory access.

Memory Cross-Attention. Following prior work [42], we integrate Cross-Attention modules within each DiT block to leverage long-term spatial memory during generation. Retrieved historical frames serve as keys and values, while current frame tokens act as queries:

$$\text{Attn}(\tilde{Q}, \tilde{K}_{\text{spatial}}, V_{\text{spatial}}) = \text{Softmax}\left(\frac{\tilde{Q}\tilde{K}_{\text{spatial}}^{T}}{\sqrt{d}}\right) V_{\text{spatial}} \tag{2}$$

where $\tilde{Q}$ and $\tilde{K}_{\text{spatial}}$ are queries and keys augmented with Plücker coordinates to encode relative pose information between current and historical viewpoints.

3.3 Memory Forcing Training

Memory-augmented video generation models face a fundamental capability trade-off. Models relying heavily on long-term spatial memory generate content consistent with previously visited scenes, but degrade when generating new scenes due to insufficient spatial context. We therefore propose Memory Forcing training protocols that teach models to dynamically balance these two capabilities, learning when to rely on temporal context versus spatial memory.

Hybrid Training. Our hybrid training approach operates within a fixed context window of L frames. We strategically allocate half the window ($L/2$ frames) as fixed temporal context frames, while the remaining $L/2$ frames are flexibly assigned based on the scene context. The complete context

window construction can be formalized as:

$$\mathcal{W} = \begin{cases} [\mathcal{T}_{\text{fixed}}, \mathcal{M}_{\text{spatial}}], & \text{if revisiting areas} \\ [\mathcal{T}_{\text{fixed}}, \mathcal{T}_{\text{extended}}], & \text{if exploring new scenes} \end{cases} \tag{3}$$

where $\mathcal{T}_{\text{fixed}}$ represents the fixed $L/2$ recent temporal context frames, $\mathcal{M}_{\text{spatial}}$ represents long-term spatial memory retrieved by our Geometry-indexed Spatial Memory (Section 3.4), and $\mathcal{T}_{\text{extended}}$ represents additional temporal frames from earlier time steps. This dynamic allocation enables the model to leverage the most appropriate memory source for different generation scenario.

Chained Forward Training. We introduce Chained Forward Training (CFT) to enhance our hybrid training strategy. CFT sequentially processes temporal windows where predicted frames from earlier windows are incorporated into subsequent windows, creating cascading dependencies across the sequence. Procedural details are provided in Algorithm 1 in Appendix A.

At each step j, we construct a temporal window $\mathcal{W}_j$ spanning frames $[j, j + W - 1]$. For each frame position k within this range, the window contains either ground-truth frames $\mathbf{x}_k$ (if not previously predicted) or predicted frames $\hat{\mathbf{x}}_k$ (if generated in earlier steps). The conditioning inputs $\mathcal{C}_j = \{\mathcal{A}_j, \mathcal{P}_j, \mathcal{M}_{\text{spatial}}\}$ comprise actions $\mathcal{A}_j$, poses $\mathcal{P}_j$, and spatial memory $\mathcal{M}_{\text{spatial}}$ retrieved for the current window. The training objective is:

$$\mathcal{L}_{\text{chain}} = \frac{1}{T} \sum_{j=0}^{T-1} \mathbb{E}_{t,\epsilon} \left[\| \epsilon - \epsilon_\theta(\mathcal{W}_j(\mathbf{x}, \hat{\mathbf{x}}), \mathcal{C}_j, t) \|^2 \right] \tag{4}$$

This approach extends autoregressive training with model rollouts, where larger pose variations created by chained predictions cause inaccuracies to propagate from earlier windows, encouraging the model to rely on spatial memory for maintaining consistency across revisited areas. Additionally, by replacing ground truth temporal context with the model's own predictions during training, this approach helps reduce accumulation errors [22] that typically arise during autoregressive inference.

3.4 Geometry-indexed Spatial Memory

Our Geometry-indexed Spatial Memory maintains explicit 3D scene geometry to enable efficient retrieval of long-term historical visual information based on spatial relationships. The core idea is to build a global point cloud representation where each 3D point is linked to its source frame, creating traceable connections back to the original visual observations. The system operates through two key components: Point-to-Frame Retrieval for identifying spatially relevant historical frames, and Incremental 3D Reconstruction that continuously updates scene representations using predicted depth maps and pose-derived camera extrinsics.

Point-to-Frame Retrieval. For each current frame, we project the global point cloud onto the current camera pose and analyze the source frame indices of the visible points to determine the most relevant historical frames:

$$\mathcal{H}_t = \text{Top-k}_{f \in \mathcal{F},\ k=L/2} \left| \left\{ p_i \in \mathcal{P}^t_{\text{visible}} : \text{S}(p_i) = f \right\} \right|, \tag{5}$$

where $\mathcal{P}^t_{\text{visible}}$ represents the set of points visible under the current camera pose for frame t, $\text{S}(p_i)$ denotes the source frame index of point p_i, L is the length of the context window, and $\mathcal{H}_t$ contains the top-k most frequently referenced historical frames among the visible points. This retrieval mechanism maintains $O(1)$ complexity regardless of sequence length, enabling scalable processing.

Incremental 3D Reconstruction. We adopt a selective reconstruction approach that dynamically determines keyframes based on spatial information content. A frame qualifies as a keyframe when it either reveals previously unobserved areas or when insufficient historical context exists:

$$\text{IsKeyframe}(t) = \text{NovelCov}(I_t, \mathcal{G}_{\text{global}}) \ \textbf{or} \ (|\mathcal{H}_t| < \tau_{\text{hist}}) \tag{6}$$

where $\text{NovelCov}(I_t, \mathcal{G}_{\text{global}})$ evaluates whether the current frame I_t contributes sufficient new spatial coverage relative to the existing global geometry $\mathcal{G}_{\text{global}}$ by rendering the global point cloud from the current pose, and $\tau_{\text{hist}} = L/2$ serves as the minimum historical frame count threshold.

Upon reaching window capacity, we jointly process keyframes, historical frames selected via Point-to-Frame Retrieval for improved geometric consistency, and overlapping frames from the previous window that provide depth scale reference for aligning the new window. VGGT generates relative

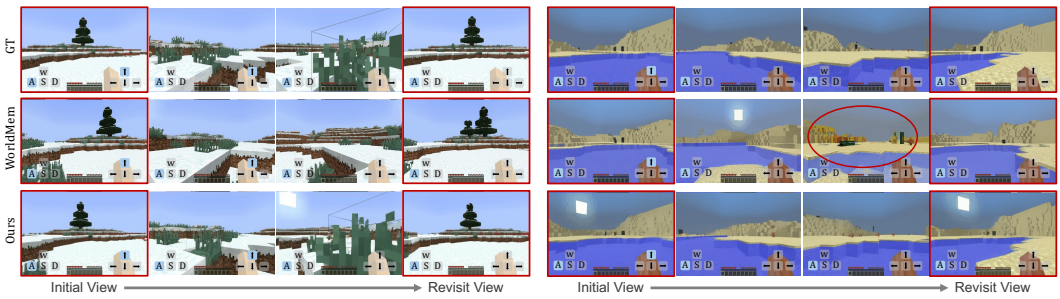

Figure 3: Memory capability comparison after 360° revisitation. Our method exhibits precise memory retention, while WorldMem generates inaccurate scene details (left: one tree becomes two) and visual artifacts (right: circled area). Additional baseline comparisons are in the appendix.

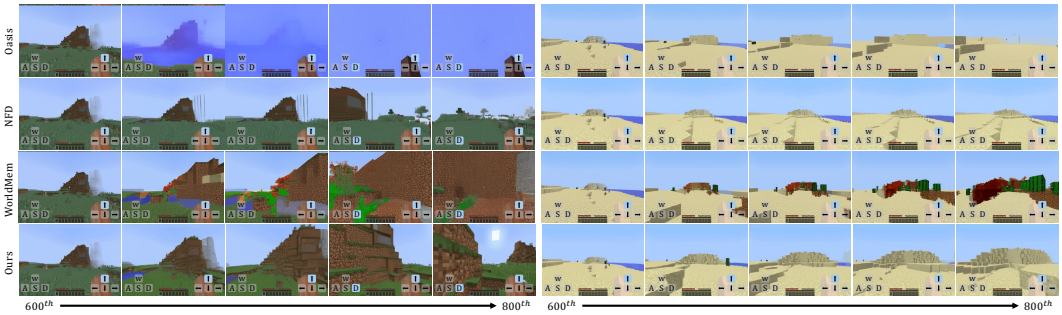

Figure 4: Scene exploration and movement dynamics. Our method progressively refines distant scenes as the agent approaches, while baselines show minimal depth variation and severe artifacts.

depth maps and confidence scores for each frame in this window, followed by our cross-window scale alignment module (detailed in Suppl.) that establishes consistent depth scale across windows through correspondence analysis in overlapping regions. 3D geometry is reconstructed through depth map back-projection using extrinsics derived from quaternion-composed poses:

$$\mathbf{E} = \begin{bmatrix} \mathbf{R}(pitch, yaw) & -\mathbf{RC} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{7}$$

where $\mathbf{R}(pitch, yaw)$ encodes the viewing orientation through quaternion-based rotation composition, and $\mathbf{C} = [x, y, z]^T$ specifies the camera's spatial position. The reconstructed geometry is subsequently integrated into our global representation through spatially-aware voxel sampling.

This design achieves efficient scene representation and retrieval through two key mechanisms. First, selective keyframe reconstruction processes and stores only frames that contribute new spatial coverage, preventing redundant computation and storage when revisiting areas. Second, voxel downsampling maintains an upper bound on point density for any pose region, ensuring constant retrieval complexity regardless of sequence length or scene scope. These mechanisms ensure that memory consumption scales with spatial coverage rather than temporal duration, enabling efficient processing of extended sequences.

4 Experiments

4.1 Experimental Setup

Implementation Details. We train for about 400k steps on 24 H20/H100 GPUs in PyTorch with batch size 4 using Adam at learning rate 4×10^{-5}. Following NFD [5], our model can be distilled to single-step denoising, achieving video generation speeds up to 44.9 FPS on H100 while using 384 × 224 video frames.

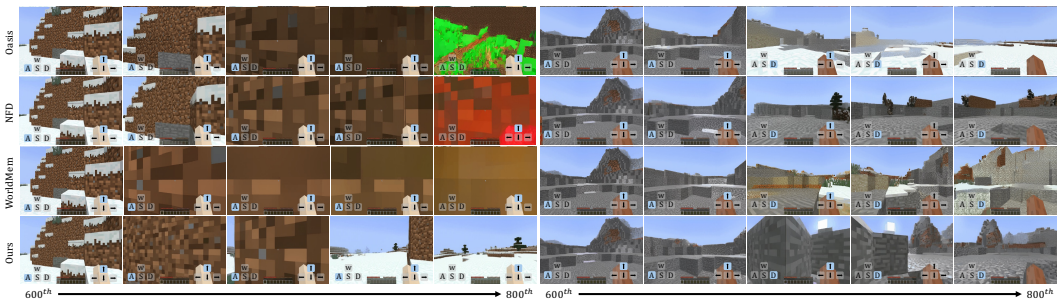

Figure 5: Robust performance across challenging scenarios. When blocked by a wall (left), our method recovers the scene after navigating around the obstacle, while baselines remain stuck with degraded quality. On unseen terrain (right), our method remains stable while baselines degrade.

Table 1: Quantitative comparison on interactive game scenarios (Minecraft [8]) across long-term memory consistency, unseen-scenario generalization, and generation quality.

Method	Long-term Memory				Generalization Performance				Generation Performance			
	FVD ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FVD ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FVD ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Oasis [6]	196.8	16.83	0.5654	0.3791	477.3	14.74	0.5175	0.5122	285.7	14.51	0.5063	0.4704
NFD [5]	220.8	16.35	0.5819	0.3891	442.6	15.49	0.5564	0.4638	349.6	14.64	0.5417	0.4343
WorldMem [42]	122.2	19.32	0.5983	0.2769	328.3	16.23	0.5178	0.4336	290.8	14.71	0.4906	0.4531
Ours	**84.9**	**21.41**	**0.6692**	**0.2156**	**253.7**	**19.86**	**0.6341**	**0.2896**	**185.9**	**17.99**	**0.6155**	**0.3031**

Datasets. For training, we use the VPT [1] dataset, which pairs 25-dimensional action vectors with corresponding video sequences. Additionally, we utilize a synthetic dataset generated from MineDoJo [8] for long-term memory training, following the configuration of WorldMem [42], which consists of 11k videos containing 1,500-frame action sequences with frequent pose transitions to previously visited spatial locations. For evaluation, we use MineDojo-based datasets covering long-term memory, unseen-terrain generalization, and new-environment generation.

Baselines and Metrics. We compare with Oasis [6], NFD [5], and WorldMem [42]; all use a 16-frame context window and matched synthetic training settings for 500–600k steps. We report FVD and LPIPS for perceptual quality, and PSNR/SSIM for pixel-level accuracy.

4.2 Model Capabilities Assessment

For all quantitative and qualitative evaluations, we generate frames from 600 to 800 (200 frames in total) to assess long-sequence generation capabilities using the datasets described in our experimental setup 4.1.

Long-term Memory. On the long-term memory set, Table 1 shows that our method achieves the best metrics and visual fidelity after revisits. Figure 3 further demonstrates that our model exhibits more precise memory retention when the camera rotates back to previously visited viewpoints. While WorldMem [42] shows some memory capabilities, it generates inaccurate scene details (e.g., the left example shows one tree becoming two trees) and exhibits unstable generation with visual artifacts (e.g., the circled area in the right example). Additional comparisons with other baselines without spatial memory mechanisms, along with complete video sequences, are provided in the supplementary materials. These baseline models exhibit spatial inconsistencies where camera viewpoint changes inappropriately alter scene geometry and terrain features due to the absence of long-term memory.

Generalization Performance. On nine unseen terrains, our method outperforms baselines across all metrics and produces stable qualitative results, while WorldMem [42] and NFD [5] show artifacts and Oasis [6] has inconsistent scene-type transitions.

Generation Performance. Our comprehensive evaluation using the generation performance dataset demonstrates that our method outperforms all baselines across metrics in Table 1, highlighting the effectiveness of balancing long-term and temporal memory. Figure 4 shows responsive exploration: distant scenes become clearer as the agent approaches, while WorldMem [42] degrades, NFD [5] lacks

Table 2: Retrieval efficiency comparison between WorldMem [42]'s pose-based retrieval and our Geometry-indexed Spatial Memory across sequence lengths on a single H20 GPU. "Mem." denotes memory-bank frames.

Frame Range	0–999		1000–1999		2000–2999		3000–3999		Total (0–3999)	
Method	Speed (FPS ↑)	Mem. (Count ↓)	Speed (FPS ↑)	Mem. (Count ↓)	Speed (FPS ↑)	Mem. (Count ↓)	Speed (FPS ↑)	Mem. (Count ↓)	Speed (FPS ↑)	Mem. (Count ↓)
WorldMem [42]	10.11	+1000	3.43	+1000	2.06	+1000	1.47	+1000	4.27	4000
Ours	**18.57**	**+25.45**	**27.08**	**+19.70**	**41.36**	**+14.55**	**37.84**	**+12.95**	**31.21**	**72.65**

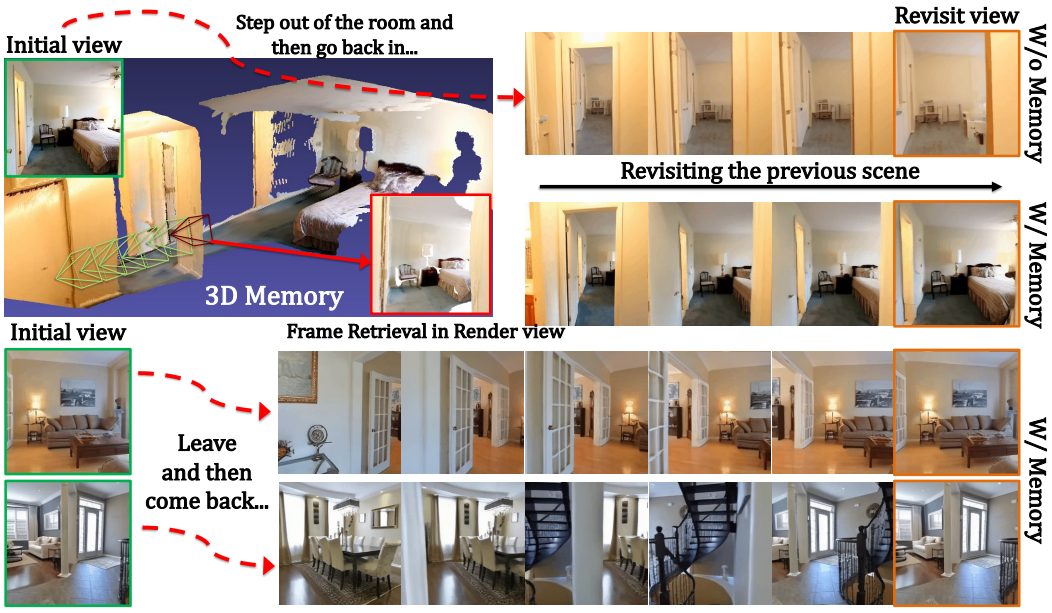

Figure 6: Qualitative results on RealEstate10K. Upon revisiting a previously seen scene, our 3D memory renders and retrieves relevant context to reconstruct the original view (W/ Memory), while the model without memory loses scene fidelity (W/o Memory).

depth variation, and Oasis [6] produces oversimplified artifacts. Figure 5 shows obstacle recovery: after wall occlusion and leftward navigation, our method recovers the scene, while baselines remain stuck with persistent degradation.

4.3 Spatial Memory Efficiency

Table 2 compares retrieval and storage on 20 4000-frame MineDojo videos.

Retrieval Efficiency. Our 3D geometry-indexed retrieval achieves 7.3× higher overall speed (31 FPS vs. 4 FPS) and 25.7× speedup at 3000–3999 frames, because voxel downsampling bounds local point density while WorldMem's pose-based retrieval slows linearly; reported speeds include our full 3D memory pipeline (reconstruction and retrieval) and WorldMem's retrieval over stored frames.

Storage Efficiency. WorldMem stores all frames and grows linearly, whereas our selective keyframes reduce storage by 98.2% (4000 to 72.65 frames) by keeping only frames with new scenes.

4.4 Evaluation on Real-World Indoor Scenes

We further evaluate our method on RealEstate10K [50], a real-world indoor scene dataset, comparing against DFoT [31] and Geometry Forcing [39]. Beyond standard image quality metrics (PSNR, SSIM, LPIPS), we introduce three memory-specific metrics: Chamfer Distance (C_{dist}), Rotation Distance (R_{dist}), and Translation Distance (T_{dist}), to assess geometric and camera pose consistency. These are evaluated via a forward-backward protocol: the model generates frames as the camera leaves a region,

Table 3: Quantitative comparison on RealEstate10K, where C_{dist}, R_{dist}, and T_{dist} are reconstruction-based memory metrics.

Method	PSNR↑	LPIPS↓	SSIM↑	$C_{dist}\downarrow$	$R_{dist}\downarrow$	$T_{dist}\downarrow$
DFoT [31]	15.09	0.3291	0.4736	0.1364	2.6730	0.1199
Geometry Forcing [39]	15.93	0.2708	0.5310	0.1185	2.1701	0.1187
Ours w/o memory	16.47	0.2661	0.5585	0.1246	2.7652	0.0921
Ours w/ memory	**25.69**	**0.0824**	**0.8120**	**0.0389**	**0.7699**	**0.0309**

Table 4: Ablation of training strategies and retrieval mechanisms. FT: full-parameter fine-tuning; HT-w/o-CFT: hybrid training without CFT; MF: Memory Forcing with HT and CFT.

Training Strategies			**Retrieval Strategies**		**Metrics**			
FT	HT-w/o-CFT	MF	Pose-based	3D-based	FVD ↓	PSNR ↑	SSIM ↑	LPIPS ↓
✓				✓	366.1	15.09	0.5649	0.4122
	✓			✓	230.4	16.24	0.5789	0.3598
		✓	✓		225.9	16.24	0.5945	0.3722
		✓		✓	**165.9**	**18.17**	**0.6222**	**0.2876**

then returns along the reversed trajectory, and VGGT [35] reconstructs scene geometry from both passes to measure consistency. As shown in Table 3, our method with memory achieves substantially superior performance across all metrics, with particularly pronounced gains on memory-specific metrics, confirming the effectiveness of our 3D spatial memory in real-world settings. Figure 6 further demonstrates that our model faithfully reconstructs previously observed scenes upon revisitation, while the model without memory fails to maintain spatial consistency.

4.5 Ablation Studies

Table 4 shows ablation results on 300 videos from the Long-term Memory and Generation Performance datasets in Section 4.1, analyzing the contributions of various training strategies and retrieval mechanisms.

Training Strategy Analysis. We compare three training approaches: full-parameter Fine-Tuning (FT) after VPT pre-training, Hybrid Training (HT) without Chained Forward Training (CFT), and our complete Memory Forcing training strategy (MF). Direct fine-tuning achieves limited performance as the model struggles to balance temporal memory and spatial memory, typically over-relying on one modality at the expense of the other. Hybrid Training (HT) demonstrates improvement by integrating real and synthetic data, but inadequately trains the model's dependence on spatial memory during spatial revisitation scenarios. Our Memory Forcing training approach achieves optimal performance by enabling the model to adaptively utilize temporal context when exploring new scenes while leveraging spatial memory when revisiting previously observed areas, effectively resolving the fundamental capability trade-off between generation quality and long-term memory retention in long-horizon generation.

Retrieval Mechanism Comparison. Our 3D-based approach substantially outperforms pose-based retrieval by leveraging explicit geometric representations for more precise identification of spatially relevant historical frames. This design not only improves accuracy but also achieves superior computational efficiency and significantly reduced storage overhead, as detailed in Table 2. The advantages become increasingly pronounced with longer sequences as pose-based approaches degrade linearly with sequence length as memory banks grow.

5 Conclusions

We introduced Memory Forcing, a novel framework that addresses the fundamental trade-off between long-term spatial memory and new scene generation in autoregressive video models. Our approach consists of two key innovations: a hybrid training strategy featuring Chained Forward Training that teaches models to dynamically balance temporal and spatial memory utilization, and an efficient Geometry-indexed Spatial Memory that achieves constant-time retrieval complexity through streaming 3D reconstruction and point-to-frame retrieval. Our framework encourages adaptive contextual selection, relying on temporal memory for new scene generation while leveraging spatial memory for consistency in previously encountered areas. Extensive experiments on both game environments and real-world scenes demonstrate that Memory Forcing achieves superior performance in both spatial consistency and generative quality while maintaining computational efficiency for extended sequences, effectively resolving the capability trade-off that has limited prior memory-augmented video models in long-horizon generation.

Acknowledgments and Disclosure of Funding

This work was supported by the Shenzhen Science and Technology Program under Grant No. ZDCY20250901113000001 and the Shenzhen Loop Area Institute under Grant No. 2026NSFP006.

References

[1] Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 2022.

[2] Haoxuan Che, Xuanhua He, Quande Liu, Cheng Jin, and Hao Chen. Gamegen-x: Interactive open-world game video generation. In *The Thirteenth International Conference on Learning Representations*, 2024.

[3] 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*, 2024.

[4] Taiye Chen, Xun Hu, Zihan Ding, and Chi Jin. Learning world models for interactive video generation. *arXiv preprint arXiv:2505.21996*, 2025.

[5] Xinle Cheng, Tianyu He, Jiayi Xu, Junliang Guo, Di He, and Jiang Bian. Playing with transformer at 30+ fps via next-frame diffusion. *arXiv preprint arXiv:2506.01380*, 2025.

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

[7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 2021.

[8] 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. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL `https://openreview.net/forum?id=rc8o_j8I8PX`.

[9] 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.

[10] Junliang Guo, Yang Ye, Tianyu He, Haoyu Wu, Yushu Jiang, Tim Pearce, and Jiang Bian. Mineworld: a real-time and open-source interactive world model on minecraft. *arXiv preprint arXiv:2504.08388*, 2025.

[11] William H Guss, Brandon Houghton, Nicholay Topin, Phillip Wang, Cayden Codel, Manuela Veloso, and Ruslan Salakhutdinov. Minerl: A large-scale dataset of minecraft demonstrations. *arXiv preprint arXiv:1907.13440*, 2019.

[12] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 2018.

[13] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[14] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

[15] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

[16] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. *Advances in neural information processing systems*, 2022.

[17] 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.

[18] Roberto Henschel, Levon Khachatryan, Hayk Poghosyan, Daniil Hayrapetyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Streamingt2v: Consistent, dynamic, and extendable long video generation from text. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025.

[19] Jonathan Ho, AjayN. Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 2020.

[20] Yining Hong, Beide Liu, Maxine Wu, Yuanhao Zhai, Kai-Wei Chang, Linjie Li, Kevin Lin, Chung-Ching Lin, Jianfeng Wang, Zhengyuan Yang, et al. Slowfast-vgen: Slow-fast learning for action-driven long video generation. *arXiv preprint arXiv:2410.23277*, 2024.

[21] 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.

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

[23] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023.

[24] Runjia Li, Philip Torr, Andrea Vedaldi, and Tomas Jakab. Vmem: Consistent interactive video scene generation with surfel-indexed view memory. *arXiv preprint arXiv:2506.18903*, 2025.

[25] Zongyi Li, Shujie Hu, Shujie Liu, Long Zhou, Jeongsoo Choi, Lingwei Meng, Xun Guo, Jinyu Li, Hefei Ling, and Furu Wei. Arlon: Boosting diffusion transformers with autoregressive models for long video generation. In *ICLR*, 2025.

[26] Dominic Maggio, Hyungtae Lim, and Luca Carlone. Vggt-slam: Dense rgb slam optimized on the sl (4) manifold. *arXiv preprint arXiv:2505.12549*, 2025.

[27] OpenAI. Video generation models as world simulators, 2024.

[28] J Parker-Holder, P Ball, J Bruce, V Dasagi, K Holsheimer, C Kaplanis, A Moufarek, G Scully, J Shar, J Shi, et al. Genie 2: A large-scale foundation world model. *URL: https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model*, 2024.

[29] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2023.

[30] Ryan Po, Yotam Nitzan, Richard Zhang, Berlin Chen, Tri Dao, Eli Shechtman, Gordon Wetzstein, and Xun Huang. Long-context state-space video world models. *arXiv preprint arXiv:2505.20171*, 2025.

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

[32] Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, WQ Zhang, Weifeng Luo, et al. Magi-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211*, 2025.

[33] Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *The Thirteenth International Conference on Learning Representations*, 2024.

[34] Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. *Advances in neural information processing systems*, 2022.

[35] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025.

[36] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025.

[37] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[38] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. pi3: Scalable permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025.

[39] 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. *arXiv preprint arXiv:2507.07982*, 2025.

[40] Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. ivideogpt: Interactive videogpts are scalable world models. *Advances in Neural Information Processing Systems*, 2024.

[41] Yuqi Wu, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Point3r: Streaming 3d reconstruction with explicit spatial pointer memory. *arXiv preprint arXiv:2507.02863*, 2025.

[42] Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. Worldmem: Long-term consistent world simulation with memory. *arXiv preprint arXiv:2504.12369*, 2025.

[43] Desai Xie, Zhan Xu, Yicong Hong, Hao Tan, Difan Liu, Feng Liu, Arie Kaufman, and Yang Zhou. Progressive autoregressive video diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025.

[44] Wilson Yan, Danijar Hafner, Stephen James, and Pieter Abbeel. Temporally consistent transformers for video generation. In *International Conference on Machine Learning*. PMLR, 2023.

[45] Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025.

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

[47] Jiwen Yu, Jianhong Bai, Yiran Qin, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Context as memory: Scene-consistent interactive long video generation with memory retrieval. *arXiv preprint arXiv:2506.03141*, 2025.

[48] 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*, 2025.

[49] Yifan Zhang, Chunli Peng, Boyang Wang, Puyi Wang, Qingcheng Zhu, Fei Kang, Biao Jiang, Zedong Gao, Eric Li, Yang Liu, et al. Matrix-game: Interactive world foundation model. *arXiv preprint arXiv:2506.18701*, 2025.

[50] 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.

Supplementary Material

A Chained Forward Training Algorithm

Algorithm 1 Chained Forward Training (CFT)

Require: Video $\mathbf{x}$, conditioning inputs $\mathcal{C}$
Require: Forward steps T, window size W
Require: Diffusion model ϵ_θ
Initialize $\mathcal{F}_{\text{pred}} \leftarrow \emptyset$, $\mathcal{L}_{\text{total}} \leftarrow 0$
for $j = 0$ **to** $T - 1$ **do**
  Construct window $\mathcal{W}_j$:
  for $k \in [j, j + W - 1]$ **do**
    if $k \in \mathcal{F}_{\text{pred}}$ **then**
      $\mathcal{W}_j[k - j] \leftarrow \mathcal{F}_{\text{pred}}[k]$ {Use predicted frame}
    else
      $\mathcal{W}_j[k - j] \leftarrow \mathbf{x}_k$ {Use ground truth frame}
    end if
  end for
  Apply noise to window: $\mathcal{W}_j \rightarrow \tilde{\mathcal{W}}_j$
  Enable gradient computation
  Compute loss: $\mathcal{L}_j \leftarrow \|\epsilon - \epsilon_\theta(\tilde{\mathcal{W}}_j, \mathcal{C}_j, t)\|^2$
  $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{total}} + \mathcal{L}_j$
  if $j < T - 1$ **then**
    Disable gradient computation
    $\hat{\mathbf{x}}_{j+W-1} \leftarrow \text{denoise}(\mathcal{W}_j, \mathcal{C}_j)$ {Fewer steps}
    $\mathcal{F}_{\text{pred}}[j + W - 1] \leftarrow \hat{\mathbf{x}}_{j+W-1}$
  end if
end for
return $\mathcal{L}_{\text{chain}} \leftarrow \mathcal{L}_{\text{total}}/T$

B Cross-Window Scale Alignment Module

The Cross-Window Scale Alignment Module addresses the challenge of maintaining consistent depth scale across processing windows when using relative depth estimation. Since VGGT produces relative depth maps that may have different scales across windows, we need to align them with previously stored geometry to maintain global consistency.

Overlap Detection. For each new window with frame indices $\mathcal{W}_{new}$, we first identify overlapping frames with the global frame store:

$$\mathcal{O} = \{f \mid f \in \mathcal{W}_{new} \cap \mathcal{F}_{global}\}$$

where $\mathcal{F}_{global}$ represents all frames previously stored.

Scale Factor Computation. For each overlapping frame $f \in \mathcal{O}$, we extract the corresponding depth maps and confidence scores from both the stored global representation and the current window: (1), Stored depth: $D_{old}^{(f)} \in \mathbb{R}^{H\times W\times 1}$; (2), New depth: $D_{new}^{(f)} \in \mathbb{R}^{H\times W\times 1}$; (3), Stored confidence: $C_{old}^{(f)} \in \mathbb{R}^{H\times W}$; (4), New confidence: $C_{new}^{(f)} \in \mathbb{R}^{H\times W}$. We concatenate all overlapping frames to form stacked representations and apply a multi-stage filtering process to ensure robust scale estimation:

- Validity Filtering: Remove pixels with invalid depths ($< 10^{-6}$).
- Confidence Thresholding: Keep only pixels where both old and new confidences exceed a minimum threshold $\tau_{min} = 0.1$.

Table 5: Comparison of Video Generation Models on Minecraft

Method	Memory Type (Complexity)	Memory Storage	Memory Scope	Dataset	Dataset Size	Video Length	Actions
Oasis	/	/	/	VPT	4000+ hours	64 frames	25
Mineworld	/	/	/	VPT	4000+ hours	64 frames	25
NFD	/	/	/	VPT	4000+ hours	64 frames	25
LSVM	State Space Model (O(1))	Compressed hidden state	Limited by training length	TECO [44]	800+ hours	150 frames	4
VRAG	Similarity-based RAG (O(1))	Fixed-length buffer	Limited by buffer size	MineRL [11]	30+ hours	1200 frames	5
Worldmem	Pose-based RAG (O(n))	Memory bank (Stores all frames)	Long-term	MineDojo	800+ hours	1500 frames	8
Ours	**3D-based RAG (O(1))**	**Scene-dependent sparse storage**	**Long-term**	**VPT+MineDojo**	**3500+ hours**	**1500 frames**	**25**

- Percentile-based Selection: Among valid pixels, compute confidence percentiles and retain only the top 60% most confident pixels for both old and new estimates.

The scale factor s is computed using least-squares optimization on the filtered correspondences:

$$s^* = \arg\min_s \sum_{i \in \mathcal{V}} (D_{old,i} - s \cdot D_{new,i})^2$$

where $\mathcal{V}$ represents the set of valid pixels after filtering. This yields the closed-form solution:

$$s = \frac{\sum_{i \in \mathcal{V}} D_{old,i} \cdot D_{new,i}}{\sum_{i \in \mathcal{V}} D_{new,i}^2}$$

Once the scale factor is determined, all depth maps in the current window are scaled: $D_{aligned} = s \cdot D_{new}$. This confidence-guided scale alignment enables efficient streaming reconstruction by maintaining consistent depth scales across processing windows, allowing the system to incrementally build coherent 3D representations without global re-optimization of the entire sequence.

C Dataset Details

We utilize the WorldMem [42] dataset together with three additional datasets collected using MineDojo [8] to evaluate our model across multiple dimensions: memory capacity, generalization abilities, scene exploration, and efficiency. The Long-term Memory evaluation set isolates 150 long videos of 1,500 frames each from WorldMem, sampled from five terrain types: *ice_plains*, *desert*, *savanna*, *sunflower_plains*, and *plains*. The Generalization Performance dataset comprises 150 sequences of 800 frames each, sampled from nine unseen terrains: *extreme_hills*, *taiga*, *stone_beach*, *swampland*, *river*, *beach*, *mesa*, *frozen_ocean*, and *forest_hills*. The Generation Performance dataset includes 300 sequences of 800 frames each for new-environment generation, while the Efficiency dataset consists of 20 sequences of 4,000 frames. For action configurations, all MineDojo-collected datasets adopt the same setup as WorldMem. For fair baseline comparison, all models follow their respective training configurations and are trained on identical synthetic datasets for approximately 500–600k steps.

D Model Comparison

Table 5 provides a comprehensive comparison of our approach with existing video generation models for Minecraft environments, highlighting key differences in memory mechanisms, storage efficiency, and capabilities across different methodological paradigms.

The first group represents traditional autoregressive video models without explicit memory mechanisms. Models like Oasis [6], Mineworld [10], and NFD [5] rely solely on temporal context windows and demonstrate strong performance in scene generation but suffer from spatial inconsistency when revisiting previously encountered areas due to their limited memory scope.

The second group encompasses recent memory-augmented approaches that attempt to extend model capabilities through various memory mechanisms. LSVM [30] employs state space models to compress historical information into hidden states, achieving constant-time complexity but with memory scope fundamentally limited by training sequence length. VRAG [4] utilizes similarity-based retrieval with fixed-length buffers, providing constant-time access but constraining long-term memory capacity. WorldMem implements pose-based retrieval with frame storage, enabling long-term memory but suffering from linear complexity growth as memory banks accumulate redundant information during extended sequences.

Table 6: Speed comparison when the model is distilled to different denoising steps on a single NVIDIA H100 GPU.

Steps	FPS ↑	SSIM ↑	PSNR ↑	LPIPS ↓
4	11.54	0.7647	19.64	0.1757
2	23.07	**0.7667**	**19.93**	**0.1732**
1	**44.96**	0.7644	19.86	0.1834

Our framework uniquely combines the advantages of both paradigms while addressing their respective limitations. Unlike traditional models, we maintain long-term spatial memory through explicit 3D scene representation. Unlike existing memory-augmented approaches, our 3D-based retrieval system achieves constant-time complexity with scene-dependent sparse storage that adapts to spatial redundancy patterns. This design enables efficient scaling to extended sequences while preserving both Long-term spatial consistency and scene generation capabilities.

E Generation Efficiency

NFD [5] extends consistency distillation to the video domain, enabling efficient inference with few sampling steps while preserving visual quality. We apply the distillation technique from NFD to our model. To evaluate the trade-off between generation efficiency and visual quality, we conduct experiments on the WorldMem [42] test set. Table 6 presents a comprehensive comparison of our model distilled to different denoising steps, with all experiments conducted on a single NVIDIA H100 GPU.

Our framework can be effectively distilled to single-step generation, achieving 44.96 FPS with only marginal degradation in visual quality. This capability makes our approach particularly suitable for real-time interactive applications, where maintaining responsive user experiences in long-horizon scenarios requires both high generation speed and consistent visual fidelity.

F Limitations and Future Work

While Memory Forcing demonstrates strong performance in memory retention and generation quality, our current approach relies on an external 3D reconstruction network (VGGT) for geometry extraction and spatial memory indexing. This dependency introduces additional computational overhead and potential points of failure, as the quality of our spatial memory system is inherently constrained by the reconstruction network's accuracy and robustness. Future research should focus on developing an end-to-end architecture that integrates geometry estimation and spatial memory within a unified framework. By learning to jointly predict visual content and underlying 3D structure, such an approach could eliminate the external reconstruction dependency while potentially achieving better alignment between geometric understanding and generation quality. An end-to-end framework would allow geometry estimation to be optimized directly for the generation task, enabling the model to learn spatial representations specifically tailored for maintaining long-term visual consistency.

G Broader Impacts

Memory Forcing may benefit interactive simulation, game development, virtual environment creation, and embodied-agent research by improving long-horizon visual consistency. At the same time, more consistent interactive video generation could be misused to create deceptive synthetic scenes or misleading virtual experiences if deployed without safeguards. Our experiments use simulated environments and existing real-world scene benchmarks rather than newly collected human-subject or sensitive personal data. Future releases should include clear usage guidelines, provenance disclosure for generated content, and restrictions against deceptive or harmful applications.

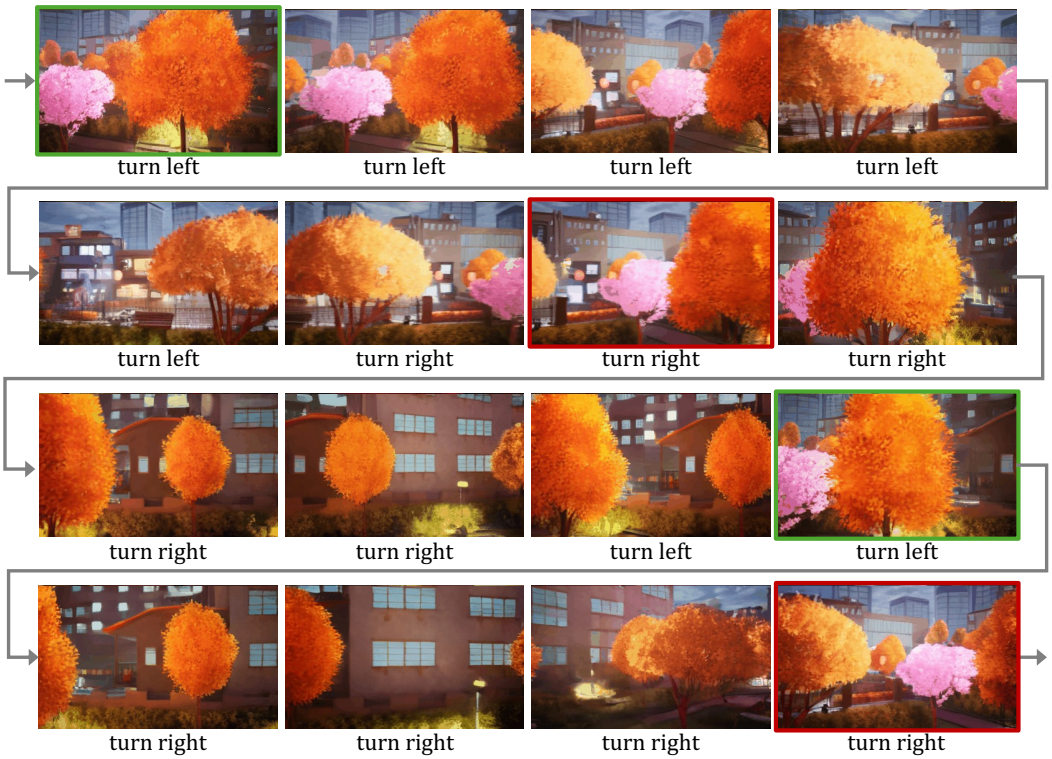

Figure 7: Cross-domain capability. Memory Forcing successfully maintains spatial consistency when revisiting locations in diverse visual styles beyond Minecraft, trained on dataset from [47].

H Asset Documentation

We credit the existing assets used in this work through their original citations, including VPT [1], MineDojo [8], WorldMem [42], RealEstate10K [50], and VGGT [35]. We use these resources for research evaluation and method implementation under their public terms and intended academic or research usage, and any released code, data, or model package will preserve the corresponding attribution, license, and terms-of-use information. The new assets introduced by this work are the constructed MineDojo evaluation splits and derived metadata described in Section C. We document their source, sequence counts, lengths, terrain categories, action configuration, and evaluation purpose in the supplementary material, and will release the associated construction scripts and documentation with the full code and datasets.

I Additional Qualitative Comparisons

We present additional qualitative analyses across Figures 8–12 to demonstrate our method's performance in scene generation. These comparisons cover diverse scenarios including generalization to unseen terrains, spatial and temporal consistency across different environments, long-term memory utilization, and rollout stability. Across all scenarios, our approach consistently outperforms baseline methods in maintaining spatial coherence, temporal continuity, and scene fidelity, while effectively handling challenging cases such as frozen ocean terrain absent from the training set and complex agent movements in unfamiliar environments.

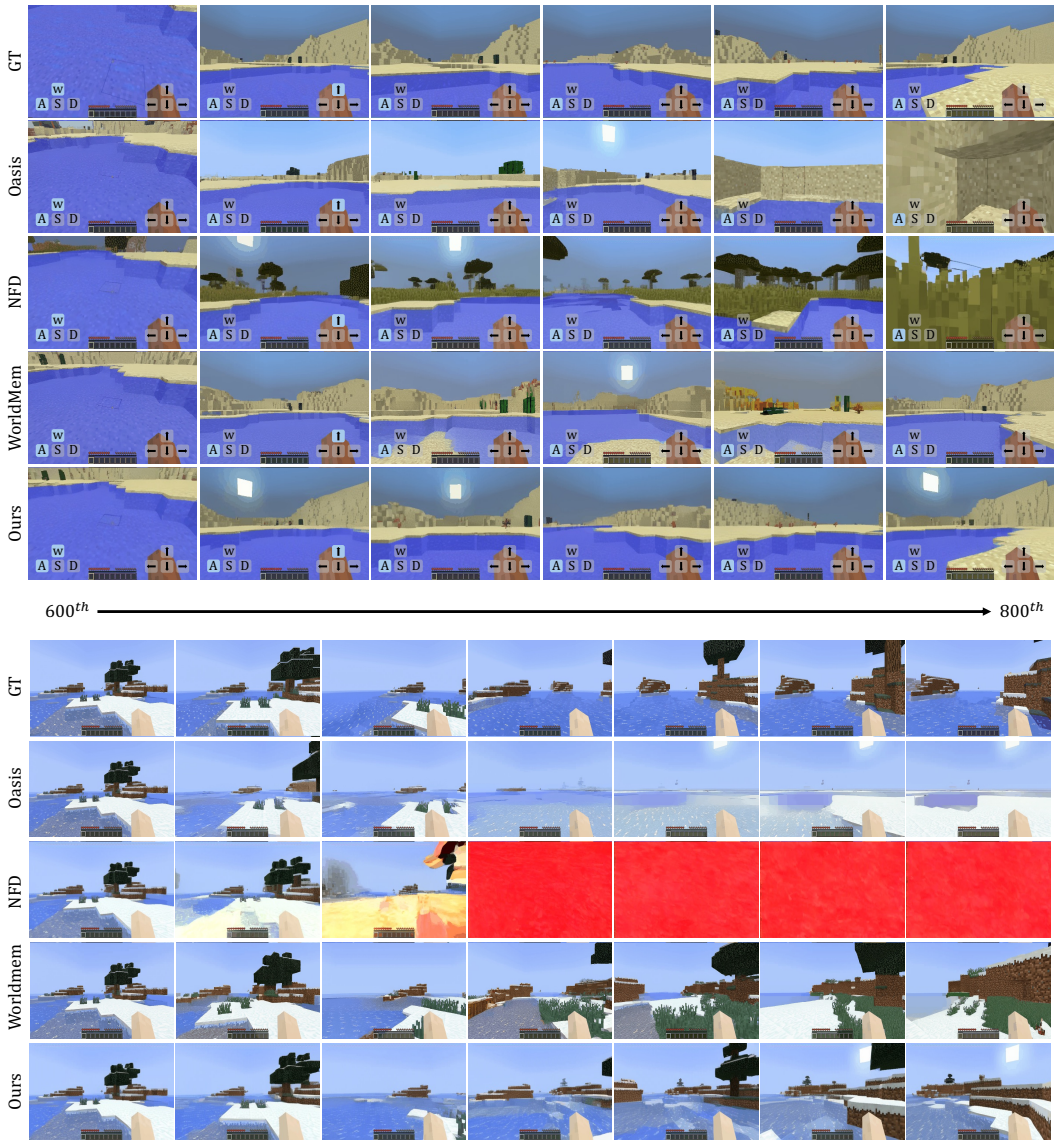

Figure 8: **Top:** Extended rollout sequences complementing the right example in Figure 3. **Bottom:** Comparison of generalization performance on frozen ocean terrain. WorldMem [42] fails to preserve the frozen ocean characteristics and produces scenes resembling plains terrain from the training set. Oasis and NFD exhibit complete visual collapse. In contrast, our model successfully maintains the frozen ocean terrain across novel scene generations.

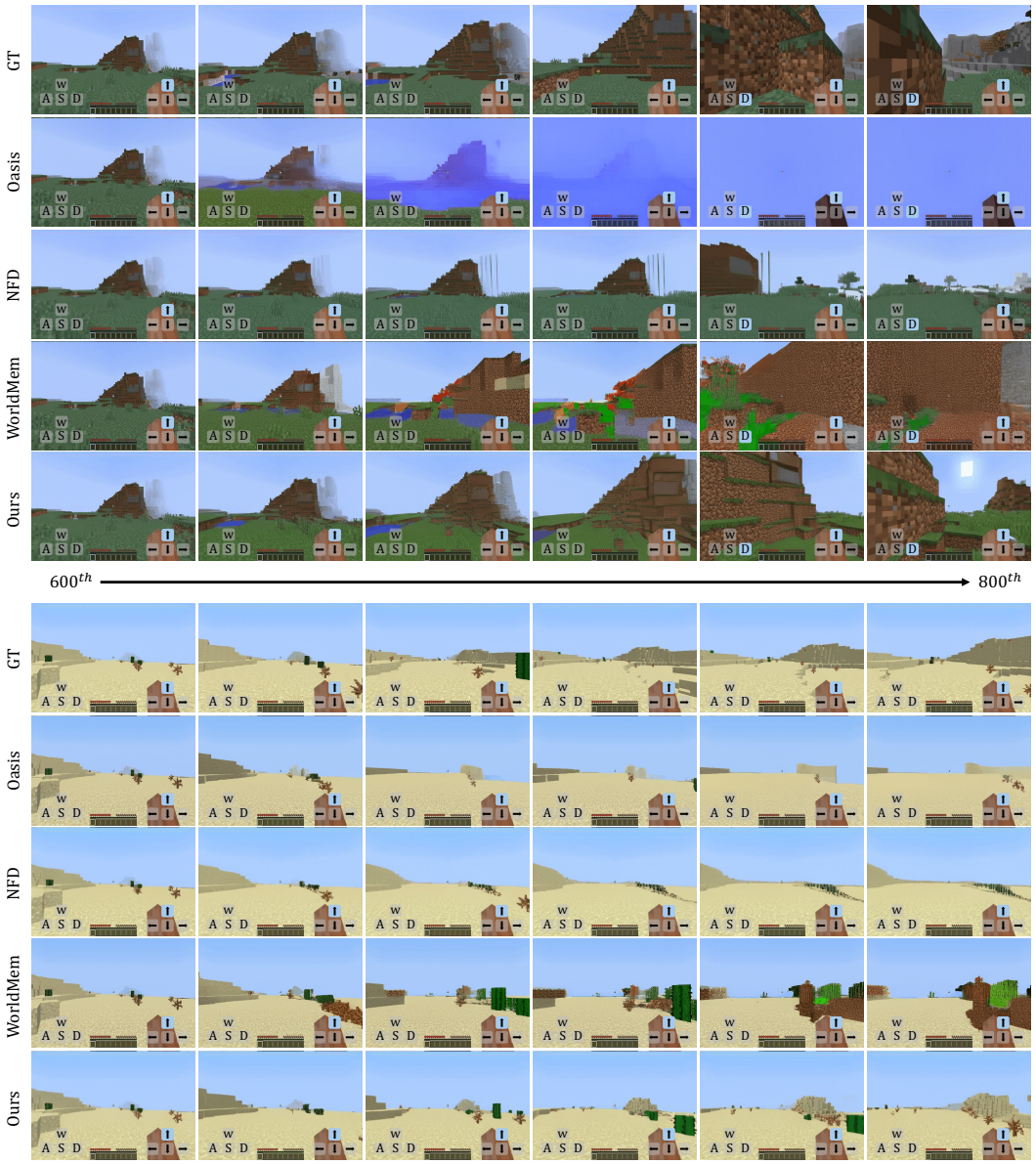

Figure 9: Qualitative results. Comparison of different models' novel scene generation on two terrains: extreme hills (top) and desert (bottom). In extreme hills, our method generates novel views while preserving spatial consistency, whereas Oasis [6] fails, collapsing to blue images. WorldMem [42] and NFD [5] produce unrealistic views that break spatial consistency. In desert, Oasis [6] and NFD [5] fail to reflect the agent's forward motion, and WorldMem [42] lacks temporal and spatial consistency. By contrast, our method maintains spatial coherence and produces rich, realistic views.

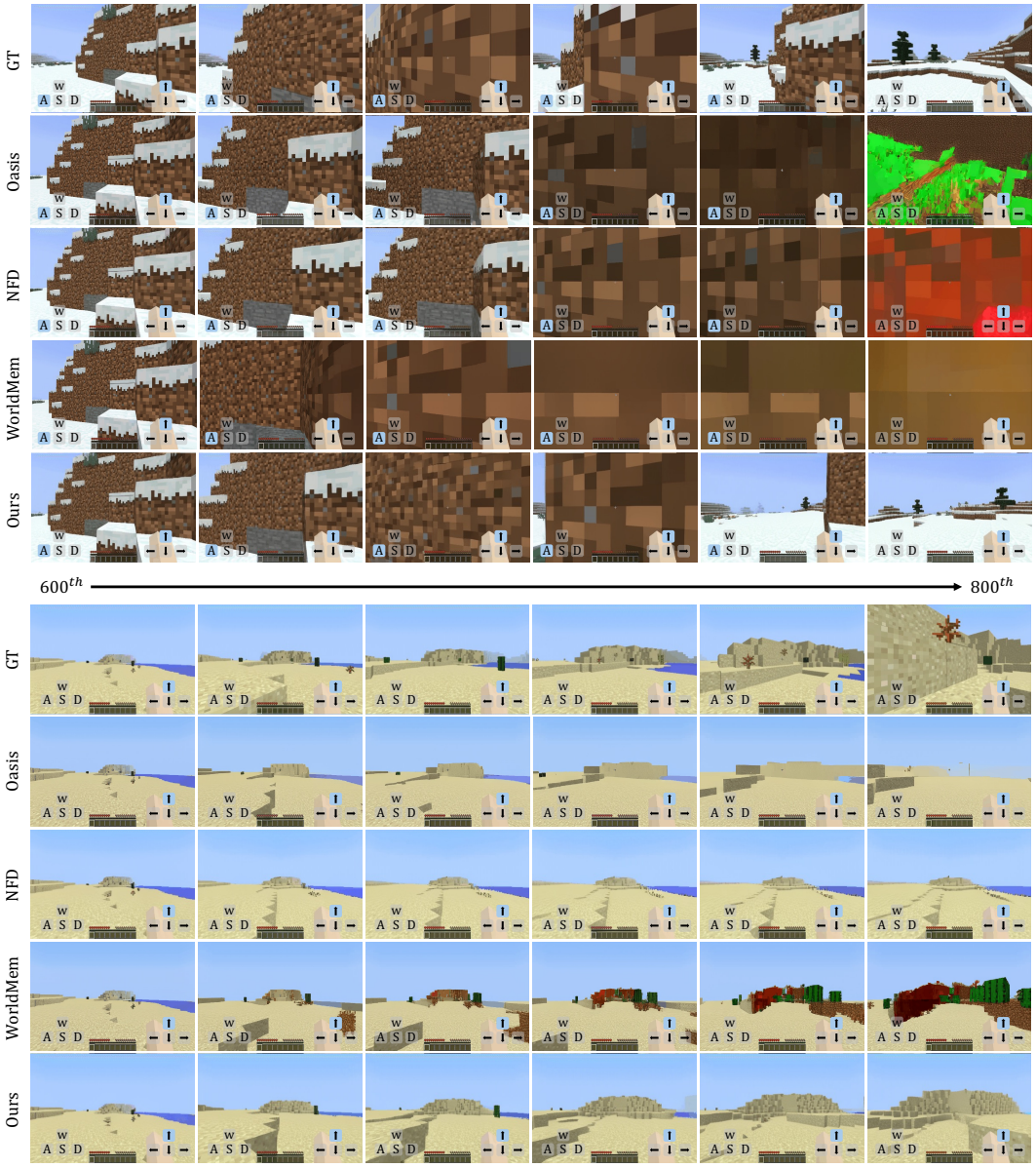

Figure 10: Qualitative results. Comparison of different models' novel scene generation on two terrains: ice plains (top) and desert (bottom). In the ice plains scenario, Oasis [6], NFD [5], and WorldMem [42] fail to produce correct views when the agent turns left and moves forward, remaining trapped. By contrast, our model successfully generates novel views after escaping while preserving the ice plains terrain. In the desert scenario, NFD [5] fails to reflect the agent's forward motion, while WorldMem [42] and Oasis [6] violate temporal and spatial consistency. Our method consistently maintains spatial coherence and generates realistic novel views.



Figure 11: Qualitative results on long-term memory across different models. We compare the generative capabilities of different models under long-term memory settings. Our model achieves the best spatial consistency, temporal continuity, and preserves rich scene details.

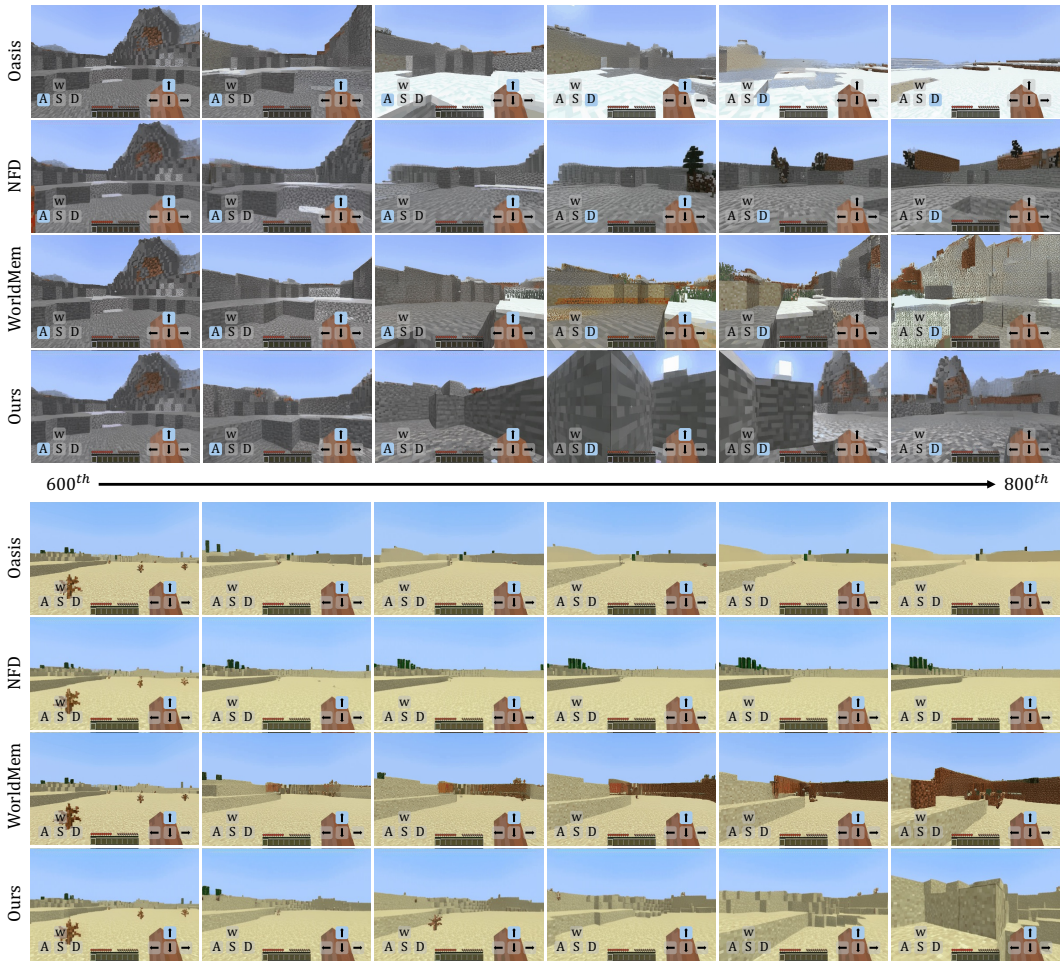

Figure 12: **Top:** Extended sequences from Figure 6, demonstrating generalization performance on unseen terrain. Our method maintains stable visual quality across longer rollouts, while baselines show significant quality degradation when encountering novel terrain. **Bottom:** Additional examples of scene exploration and movement dynamics. Our method shows progressive visual refinement as the agent approaches distant scenes, while baselines exhibit minimal depth variation and severe visual artifacts.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes].

   Justification: The abstract and Section 1 state the paper's motivation, Memory Forcing framework, geometry-indexed memory mechanism, and empirical claims. Section 4 reports evaluations corresponding to these claims.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes].

   Justification: The supplementary Limitations and Future Work section discusses reliance on an external 3D reconstruction network and the resulting overhead and failure modes.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes].

Justification: The paper's mathematical content consists of model objectives and algorithmic retrieval/reconstruction procedures rather than standalone theorem-style results. All variables, assumptions, and procedural details needed for these formulations are defined in the main text and appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes].

Justification: The paper discloses the model architecture, training protocol, datasets, baselines, evaluation frame ranges, context window size, metrics, optimizer, learning rate, batch size, GPU types, resolution, and appendix algorithm/dataset details needed to reproduce the main experimental claims.

Guidelines:

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

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No].

   Justification: This paper does not provide a public code or dataset release URL. We plan to release the full codebase and constructed datasets.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes].

   Justification: Section 4.1 describes datasets, baselines, evaluation frame ranges, context window size, metrics, optimizer, learning rate, batch size, GPU types, and resolution. The appendix adds dataset details and efficiency settings.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No].

   Justification: The current tables report point estimates on fixed evaluation sets and do not include repeated-run error bars, confidence intervals, standard deviations, or statistical significance tests. Full repeated training/evaluation runs are computationally expensive for long-sequence video generation, so the current submission focuses on controlled comparisons under matched settings.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes].

   Justification: The paper reports GPU types, 24-GPU training, training steps, batch size, and single-GPU speed measurements for retrieval and distilled generation, providing the compute information needed to understand and reproduce the reported experiments.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes].

   Justification: The work uses existing datasets and simulated environments, and the current draft does not involve new human-subject data collection or sensitive personal data collection.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes].

    Justification: Appendix G discusses potential benefits for interactive simulation, game development, virtual environments, and embodied-agent research, as well as potential misuse risks from deceptive synthetic scenes or misleading virtual experiences.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A].

    Justification: The current draft does not state that a high-risk model or dataset is being released with the submission. If a model or dataset is released later, the release plan should describe appropriate safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

Answer: [Yes].

Justification: Appendix H credits the existing assets used in the work, including VPT, MineDojo, WorldMem, RealEstate10K, and VGGT, and states that their attribution, licenses, and terms of use will be preserved in any released code, data, or model package.

Guidelines:

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

13. **New assets**

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

Answer: [Yes].

Justification: Appendix H identifies the constructed MineDojo evaluation splits and derived metadata as the new assets, while Appendix C documents their sources, sequence counts, lengths, terrain categories, action configuration, and evaluation purpose. The associated construction scripts and documentation will be released with the full code and datasets.

Guidelines:

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

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

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

Answer: [N/A].

Justification: The paper does not report new crowdsourcing experiments or new research with human subjects.

Guidelines:

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

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

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

    Answer: [N/A].

    Justification: The paper does not report new crowdsourcing experiments or new research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A].

    Justification: The core method does not use LLMs as an important, original, or non-standard component.

    Guidelines:

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