PermaVid: Consistent Video Generation Across Edits via Disentangled Context Memory

Shuai Yang*[1], Bingjie Gao*[1], Ziwei Liu[3], Jiaqi Wang[5],
Dahua Lin[4], Tong Wu[2]
[1] Shanghai Jiao Tong University [2] Stanford University [3] S-Lab, Nanyang Technological University
[4] The Chinese University of Hong Kong [5] Shanghai Innovation Institute

ys-imtech.github.io/projects/PermaVid *Equal contribution.



Figure 1: We propose **PermaVid**, a framework for consistent video generation across edits. For global edits (e.g., style transformation), PermaVid propagates updated semantics consistently across time and viewpoints while maintaining stable geometry. For local edits (e.g., object-level editing), the model reliably recalls the post-edit content during revisiting, preserving both structural integrity and updated local semantics.

Abstract

Consistent video generation under editing operations requires persistence: when edits modify scene appearance or layout, subsequent generations should remain coherent across time and viewpoints. However, existing memory designs struggle to maintain long-term consistency after such modifications, as stored contexts may become outdated or invalid. To address this, we propose PermaVid, a novel framework built upon a multi-modal context memory that disentangles spatial context into semantic appearance and geometric structure, together with an edit-aware memory update and retrieval strategy that keeps memory evolution aligned with subsequent observations. Specifically, we develop two complementary memory banks: an RGB context memory that captures appearance-aware observations while implicitly encoding geometry, and a depth context memory that preserves geometry-only structure disentangled from semantics. Building on this design, we introduce a memory-guided video generation model that performs multi-modal feature fusion under reference conditions drawn from mixed-modality memory contexts. Experiments demonstrate that our method maintains strong long-term semantic and structural consistency after edits, significantly outperforming state-of-the-art methods.

1 Introduction

Recent advances in camera-controlled video generation [1, 2, 3, 4] and video editing methods [5, 6, 7, 8, 9] have significantly expanded the flexibility of visual content creation. Users can now synthesize

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

videos by specifying camera trajectories, or modify existing videos through editing instructions such as style transformation, object insertion, or scene manipulation. These capabilities enable richer interactive experiences and more controllable generation pipelines [10, 11, 1, 3, 12]. However, a fundamental challenge remains: maintaining visual consistency over long temporal horizons. This challenge becomes particularly pronounced when the camera continuously moves and revisits previously observed regions, where the model is required to generate structurally consistent content across varying viewpoints. Furthermore, when editing operations are introduced into the video, such as global style changes or local object modifications, the model must preserve spatial coherence, and ensure that the edited results remain consistent with both past and future content. Achieving such consistency across time, viewpoints, and edits remains an open problem.

To address long-term consistency, existing methods [13, 14, 4, 15, 16, 17, 18] have explored various memory-based strategies. Some recent studies [16, 17, 18] focus on temporal context modeling, where latent states or historical frames are stored to stabilize generation over time. Others [13, 14, 4, 15] leverage pose-conditioned feature retrieval, enabling the model to reuse previously observed visual information under similar viewpoints. These approaches improve temporal coherence and view consistency by incorporating historical information. However, they generally assume that past contexts remain valid throughout the generation process. This assumption breaks down in the presence of editing operations. When global edits (e.g., style transfer) or local edits (e.g., object replacement) occur, historical contexts may become partially or entirely outdated. As a result, models tend to rely on stale information, leading to semantic inconsistency, visual artifacts, or even reversion to pre-edit states.

In this work, we revisit the nature of spatial information in videos and identify a key insight: spatial context can be decomposed into two fundamentally different components—*semantic appearance* and *geometric structure*. These two components exhibit distinct temporal behaviors. Semantic appearance is highly dynamic and can change frequently due to editing operations, lighting variations, or style transformations. In contrast, geometric structure is typically stable over time or changes only locally. Existing methods entangle these two aspects within a unified representation, making the entire memory unreliable when semantics change. Consequently, even if the underlying geometry remains valid, it cannot be effectively reused due to its coupling with outdated appearance information. This observation motivates the need for a disentangled representation of spatial context, where semantic and geometric information can be independently updated and reused. Such a design allows the model to selectively refresh appearance while preserving stable structural knowledge, thereby enabling consistent generation across edits.

Based on this insight, we propose **PermaVid**, a novel framework for consistent video generation under editing operations. Our approach introduces a disentangled multi-modal context memory, where semantic appearance and geometric structure are modeled as separate but complementary memory representations. Specifically, we maintain an RGB-based memory to capture appearance information and a depth-based memory to encode geometry. To support flexible editing scenarios, we further design an edit-aware memory update and retrieval mechanism, which adheres to the division of video edits into global and local categories as presented in Ditto [19]. Global edits trigger semantic updates by invalidating outdated appearance memory while preserving geometric structure, whereas local edits selectively update only the affected regions. During generation, the model retrieves context from these memories in a spatially-aware manner and performs multi-modal feature fusion to guide novel view synthesis. This enables the model to generate videos that remain consistent across time, viewpoints, and editing operations. To enhance model learning capacity, we further leverage Unreal Engine with a built-in navigation agent to construct a long-form video dataset, termed **UE-Mem**. This dataset contains rich cross-scene revisiting trajectories, which enables our model to better learn the disentangled memory mechanism. Extensive experiments demonstrate that our method significantly outperforms existing approaches in maintaining both structural and semantic consistency under complex editing scenarios.

2 Related Works

2.1 Video generation

Diffusion models [20, 21, 22] have become the dominant approach in video generation. Various efforts have been made to enhance the performance of video generation, including improvements in learning

strategies [6, 23, 24], data curation and frame selection [25, 26, 27], and prompt engineering [28, 29]. Video generation has progressed from U-Net–based models [30, 31] to Transformer-based diffusion frameworks [1, 32], enabling realistic and temporally coherent videos. Recently, autoregressive approaches [33, 34] have been explored in video diffusion, reformulating generation from full-sequence denoising to a step-wise process.

2.2 Camera-controlled Video Generation

Camera-controlled video generation has emerged as an important direction toward controllable and interactive video synthesis. These methods introduce camera pose conditions or spatial constraints to guide generation along predefined trajectories, enabling scene exploration, viewpoint traversal, and revisiting [35, 36, 3, 37]. By leveraging pose-conditioned generation or spatially-aware representations, these approaches partially alleviate geometric inconsistency and improve cross-view coherence. Despite these advances, existing methods still struggle to maintain global consistency over long temporal horizons, particularly under complex camera motions, viewpoint revisiting, or iterative editing operations. Camera-controlled approaches often rely on historical frames, latent states, or implicit memories to preserve cross-view coherence, but they are typically designed for static, unedited scenes. When editing introduces global or local edits, such methods may retrieve outdated context, leading to content reversion, inconsistent appearance, or structural misalignment.

2.3 Memory-Augmented Video Models

Memory mechanisms have been widely explored to enhance temporal coherence and controllability in video generation. Existing approaches introduce temporal memory, feature caching, or spatial retrieval mechanisms to retain historical information, such as compressed keyframes [17] or spatiotemporal context representations [38, 16]. In memory-based video generation, recent methods further employ pose-conditioned feature retrieval or external memory structures [3, 13, 14, 4, 11, 15] to preserve long-term consistency across time and viewpoints. However, most existing memory designs store past observations in a unified representation, where semantic appearance and geometric structure are implicitly entangled. Such coupled memories are sufficient when the scene state remains unchanged, but they become less flexible under editing operations. For instance, an appearance edit may invalidate the stored semantic content while leaving the underlying geometry still reusable, whereas a local object edit may only affect a spatially bounded region. Without separating these factors, existing methods lack a principled way to determine which memory entries should be preserved, updated, or discarded. This limitation motivates our proposed *disentangled context memory*, which explicitly separates appearance-oriented RGB memory from geometry-oriented depth memory. This design enables edit-aware memory invalidation, selective regional updates, and reliable context reuse under editing operations.

3 Method

Our goal is to achieve consistent video generation across edits, where edited content should persist in subsequent frames and remain coherent when changing viewpoints. This requires preserving the latest edited semantic appearance while maintaining reusable scene geometry over long temporal horizons. To this end, we revisit how spatial context is stored, updated, and retrieved in memory-based generation. Our key insight is that spatial context can be decomposed into **semantic appearance**, which may change frequently due to editing operations, and **geometric structure**, which is usually stable or changes only locally. Based on this observation, PermaVid introduces a disentangled multi-modal context memory (section 3.1) that separates appearance from geometry, enabling selective memory updating, invalidation, and reuse across edits. During generation, PermaVid applies an edit-aware memory update and retrieval strategy (section 3.2) to select spatially relevant and valid references, and fuses the retrieved multi-modal context in a memory-guided video generation model (section 3.3). We further construct a synthetic training dataset with long revisiting trajectories and accurate camera poses to support learning this memory behavior (section 3.4). The pseudocode of the full pipeline is provided in the supplementary material.

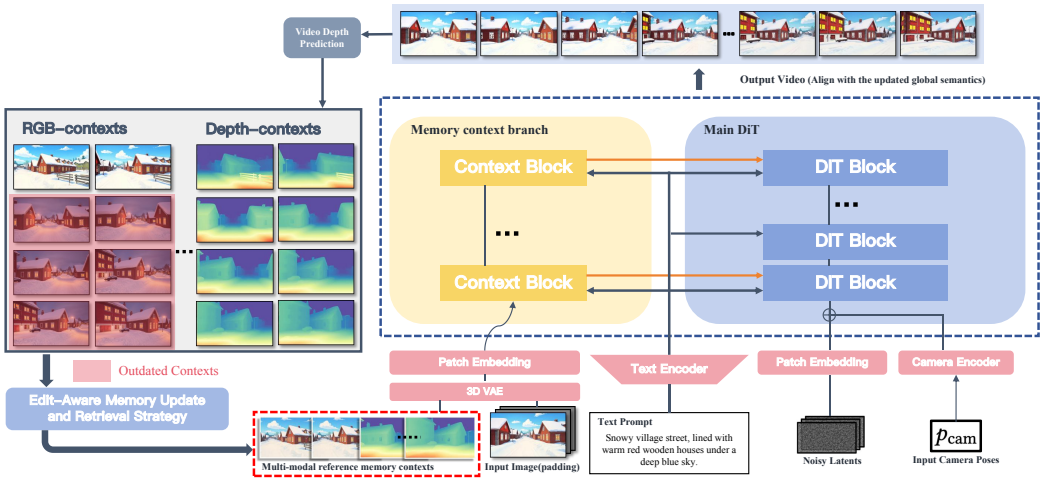

Figure 2: **Overview of PermaVid.** PermaVid maintains a disentangled multi-modal context memory with an RGB bank for semantic appearance and a depth bank for geometric structure. Given target camera poses and editing operations, it updates and retrieves memory in an edit-aware manner, then fuses mixed-modality references to guide consistent video generation across time, viewpoints, and edits.

3.1 Disentangled Multi-modal Context Memory

We decompose spatial context into two complementary factors: semantic appearance A_t and geometric structure G_t. Semantic appearance includes visual attributes such as object identity, texture, color, illumination, and style, which may change after editing operations. Geometric structure describes scene layout, object shape, and spatial relationships, which are usually stable under global appearance edits and change only locally under object-level edits. This distinction motivates two separate memory banks:

$$\mathcal{M}^{\text{rgb}} = \{(I_i^{\text{rgb}}, \mathbf{p}_i, t_i, g_i)\}_{i=1}^{N}, \qquad \mathcal{M}^{\text{dep}} = \{(I_j^{\text{dep}}, \mathbf{p}_j, t_j)\}_{j=1}^{M}. \tag{1}$$

Here I_i^{rgb} and I_j^{dep} denote RGB and depth observations, $\mathbf{p}$ is the camera pose, t is the timestamp, and g_i records the global semantic version when an RGB memory unit is inserted.

The RGB memory provides high-fidelity appearance references for viewpoint-consistent synthesis. However, RGB observations naturally entangle appearance and geometry: the same image contains both semantic content and spatial layout. As a result, old RGB memory can become semantically outdated after an edit even when its underlying geometry is still useful. In contrast, the depth memory stores geometry-oriented context that is largely invariant to changes in texture, lighting, or style. By separating these modalities, PermaVid can refresh outdated appearance while preserving reusable structure, instead of discarding the entire historical context after every edit.

3.2 Edit-aware Memory Update and Retrieval

PermaVid updates and retrieves memory according to the type and spatial extent of the editing operation. Following the video editing taxonomy in Ditto [19], we consider two edit types: *global edits* and *local edits*. Global edits, such as style transfer, seasonal change, or lighting transformation, modify the semantic appearance of the whole scene while usually preserving its geometry. Local edits, such as object insertion, removal, or replacement, affect only a bounded region and may or may not change local geometry.

Memory update. For a global edit, all previously stored RGB contexts become semantically outdated, so PermaVid invalidates the RGB memory and advances the global semantic version g^*. The depth memory is retained because the scene geometry remains valid:

$$\mathcal{M}^{\text{rgb}} \leftarrow \emptyset, \qquad g^* \leftarrow g^* + 1. \tag{2}$$

For a local edit with affected region Ω_e, PermaVid invalidates only memory units whose view footprint overlaps the edited region. Let $\Pi(\mathbf{p})$ denote the spatial footprint of a memory unit observed from pose $\mathbf{p}$. The RGB update is

$$\mathcal{M}^{\text{rgb}} \leftarrow \mathcal{M}^{\text{rgb}} \setminus \{m_i^{\text{rgb}} \mid \Pi(\mathbf{p}_i) \cap \Omega_e \neq \emptyset\}. \tag{3}$$

The depth memory follows the same local invalidation rule only when the edit changes geometry; otherwise, it is preserved. This selective update prevents stale edited content from being reused while keeping valid context in unaffected regions.

Memory retrieval. At generation time, the model retrieves references that are both spatially relevant to the target camera trajectory and valid under the current edited state. For each memory unit, we compute a trajectory-level overlap score

$$s_i = \max_{\mathbf{p}_q \in \mathbf{P}} \mathcal{L}(\mathbf{p}_i, \mathbf{p}_q), \tag{4}$$

where $\mathbf{P}$ is the target camera trajectory and $\mathcal{L}$ measures normalized view-frustum overlap. RGB memory is retrieved only if it is spatially relevant and belongs to the current global semantic version:

$$R_c^{\mathrm{rgb}} = \{m_i^{\mathrm{rgb}} \in \mathcal{M}^{\mathrm{rgb}} \mid s_i > \tau,\ g_i = g^*\}. \tag{5}$$

Depth memory is retrieved based only on spatial relevance, since it is not invalidated by purely semantic edits:

$$R_c^{\mathrm{dep}} = \{m_j^{\mathrm{dep}} \in \mathcal{M}^{\mathrm{dep}} \mid s_j > \tau\}. \tag{6}$$

The final candidate set is $R_c^{\mathrm{rgb}} \cup R_c^{\mathrm{dep}}$. To keep the reference set compact, we greedily select at most B memory units that cover the target trajectory while filtering out redundant views with high mutual overlap. This produces a spatially diverse mixed-modality reference set $\mathcal{R}^{\mathrm{mem}}$ for generation.

This update-and-retrieval design is the key mechanism that aligns memory with editing operations. Global edits refresh outdated appearance globally while preserving geometry; local edits only refresh affected regions; retrieval then avoids stale appearance memory while still reusing valid geometric context.

3.3 Memory-guided Video Generation

Given the selected memory references $\mathcal{R}^{\mathrm{mem}}$, PermaVid uses a memory-guided video generation model built upon a diffusion Transformer (DiT). Target camera poses $p_{\mathrm{cam}} = [R, T] \in \mathbb{R}^{f \times (3\times 4)}$ are encoded by a camera encoder and injected into the main DiT tokens, together with the text condition. To incorporate memory references, we add a dedicated memory context branch composed of distributed and cascaded Context Blocks duplicated from selected DiT layers.

The memory branch encodes RGB and depth references as mixed-modality conditions. Since retrieved memory frames are sparse observations sampled from different timestamps, each RGB or depth frame is independently encoded using a shared 3D VAE. The resulting memory tokens are concatenated with relative positional encoding within the memory set, avoiding reliance on absolute temporal indices and improving robustness to long temporal gaps. To better support heterogeneous conditions, the memory branch follows the VACE [39] context-branch design and is initialized from pretrained weights. The generation process can be summarized as

$$x_{t-1} = \epsilon_\theta(x_t,\, c_{\mathrm{text}},\, p_{\mathrm{cam}},\, \mathcal{R}^{\mathrm{mem}}), \tag{7}$$

where $\mathcal{R}^{\mathrm{mem}}$ provides appearance and geometry references for consistent novel-view synthesis under edits.

3.4 Dataset Construction

To support learning long-term memory behavior, we require long videos with revisiting trajectories and accurate camera poses. Existing public datasets either lack pose metadata or, like [40, 41], mainly contain unidirectional camera motion without sufficient revisits. We therefore build an automatic data synthesis pipeline in Unreal Engine 5 [42, 43]. Within each scene, a navigation agent combines two policies: a goal-driven navigator that explores distant regions and a pose tracker that induces local loops by following targets. Their combination naturally produces trajectories that revisit previously observed regions while maintaining broad scene coverage.

The agent uses a continuous 4-DoF control vector $\mathbf{u} = [v_x, v_y, \omega, g]$, where v_x and v_y denote lateral and forward velocity, ω is yaw rate, and g is pitch angle. During recording, the agent is hidden, and a first-person camera rigidly attached to it captures long RGB and depth sequences with accurate 6-DoF camera poses. The resulting UE-Mem dataset contains 4k high-quality videos, each with 1000 frames, across 100 Unreal Engine scenes. We annotate each video caption using Qwen3-VL-7B [44].

Table 1: Quantitative comparison across edits. We achieve the best overall performance under both **global** and **local** edits, demonstrating strong long-term consistency.

Methods	Global Edits					Local Edits				
	Structure Consistency			Semantics consistency	Visual Quality	View Recall Consistency			Visual Quality	
	PSNR↑	SSIM↑	LPIPS↓	CLIP-Vid↑	VBench-Avg↑	PSNR↑	SSIM↑	LPIPS↓	VBench-Avg↑	
HY-Worldplay	<u>22.34</u>	<u>0.8674</u>	<u>0.2339</u>	19.96	0.8488	<u>21.922</u>	<u>0.8523</u>	<u>0.2470</u>	0.8320	
HY-Gamecraft	18.83	0.8194	0.2951	<u>26.17</u>	**0.8691**	15.492	0.4388	0.4351	<u>0.8377</u>	
Matrix-Game-2.0	16.40	0.7065	0.4054	21.86	0.8417	14.408	0.3394	0.3465	0.8086	
VMem	14.92	0.6621	0.4386	18.74	0.8264	16.207	0.7018	0.3216	0.7893	
Ours	**22.84**	**0.8703**	**0.2102**	**27.87**	<u>0.8565</u>	**22.332**	**0.8622**	**0.2369**	**0.8544**	

4 Experiments

4.1 Implementation Details

We implement our memory-guided video generation model based on the Wan2.1-14B [5] architecture. During training, we employ variable-length video clips, with the number of frames randomly sampled between 25 and 81, with a fixed resolution of 480×832. To preserve the strong multi-modal feature perception capabilities in the memory context branch pretrained from VACE [39], we keep this branch frozen and train only the parameters of the main DiT and camera encoder. The training process is conducted in two stages. In the first stage, we train the model on the public short-video dataset SpatialVid [41] with camera pose annotations to activate its camera-guided video generation capability. Training is performed on 32 NVIDIA H200 GPUs for 10k steps. Specifically, in this stage, the input image is padded to the target video length and fed into the memory context branch without any additional reference images. In the second stage, we train the model on our constructed long-video dataset UE-Mem, which features revisiting trajectories, to further learn memory-guided generation from mixed multi-modal reference contexts. This stage is conducted for 8k training steps. Among the training samples, 40% use pure RGB reference contexts, 40% use pure depth reference contexts, and the remaining 20% employ mixed-modality contexts. At inference time, we set the memory context size to 10, indicating that references are retrieved from 10 distinct views, with each view providing information in either the RGB or depth modality. At each autoregressive iteration, the last frame of the previously generated video chunk is used as the input image. The generated video is then fed into a depth predictor [45] to obtain the corresponding depth contexts, which are subsequently incorporated into our multi-modal context memory, as illustrated in Figure 2.

4.2 Metrics and Baselines.

To evaluate consistent video generation across edits, we compare our approach against existing state-of-the-art methods, including HY-Worldplay [3], HY-Gamecraft [35], Matrix-Game-2.0 [46], and VMem [15]. We construct a benchmark test set of 200 images sourced from free websites and AI-generated images, covering both realistic and stylized scenes as well as indoor and outdoor environments. The evaluation set includes random and complex camera trajectories with viewpoint revisiting, making it suitable for testing long-term cross-view consistency after edits. For each case, all methods are evaluated with the same predefined action scripts, including the input camera trajectory and the prompt and timing of the editing operation. To ensure a controlled comparison, we use Qwen-Image [47] to apply the edit during streaming generation for all methods. We evaluate two settings following the global/local edit taxonomy. 1) For the **long-term consistency after local edits**, we measure view recall consistency and visual quality. View recall consistency is measured using PSNR, SSIM, and LPIPS on paired RGB frames captured after the edit at the same camera location, within video sequences generated along forward and reversed camera trajectories. Visual quality is measured using VBench [48]. 2) For the **long-term consistency after global edits**, we decompose view recall consistency into structural consistency, measured by PSNR, SSIM, and LPIPS on paired depth frames, and semantic consistency, evaluated by CLIP-Vid similarity [49], since global edits change semantic appearance while preserving geometry.

4.3 Qualitative Comparison

4.3.1 Long-term consistency after global edits

We conduct qualitative comparisons with other state-of-the-art methods, focusing on two key criteria: structural consistency and semantic consistency. This evaluation is motivated by the fact that

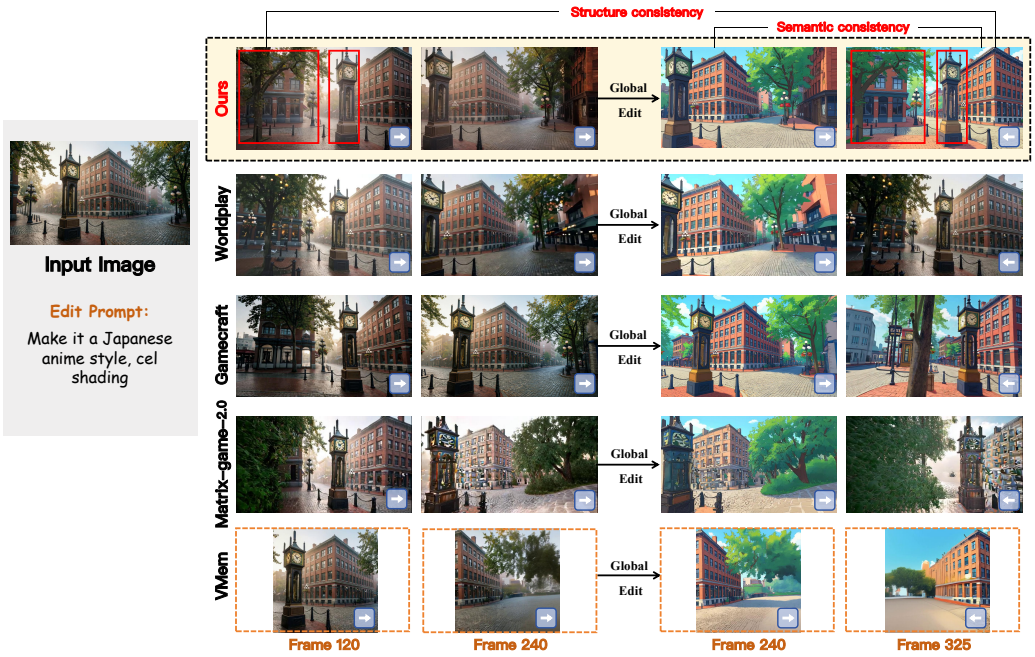

Figure 3: **Qualitative comparison under global edits.** Under a global edit (e.g., style transformation), our method maintains stable geometric structure while consistently propagating the edited semantic appearance across time and viewpoints.

global edits alter the overall semantic appearance, while the underlying geometric structure should remain stable. As shown in Figure 3, after applying a global edit at frame 240, the camera revisits previously observed viewpoints along the trajectory. Our method preserves structural consistency under revisiting views, as evidenced by the stable spatial layout and intact geometric structures (e.g., building placements and clock tower geometry). At the semantic level, our model updates the scene to reflect the latest edited appearance, enabling coherent style propagation across different viewpoints and over time. In contrast, other methods exhibit clear memory degradation. HY-WorldPlay preserves geometric consistency under reversed viewpoints but fails to update semantics, resulting in outdated styles after the global edit. HY-Gamecraft and Matrix-Game-2.0 fail to maintain consistency in both structural and semantic aspects. VMem performs better than purely short-context baselines in some consistency-related metrics, but still lags behind our method due to its limited ability to preserve sharp structure while updating edited semantics. This suggests that simply reusing historical memory is insufficient for stable long-term propagation without explicit disentanglement between geometry and appearance.

4.3.2 Long-term consistency after local edits

We evaluate view recall consistency under local edits by examining whether the edited content is correctly recalled when the camera revisits the same viewpoints. As shown in Figure 4, after a local edit is applied to a specific region, our method consistently recalls the post-edit content during revisiting, preserving both structural integrity and updated local semantics. In contrast, HY-WorldPlay fails to properly perceive and retain local edits: although it maintains view consistency during revisits, the recalled content corresponds to the pre-edit state, indicating insufficient edit awareness. HY-Gamecraft and Matrix-Game-2.0 exhibit severe inconsistency, as their limited short-term context (e.g., chunk-level KV cache) is insufficient to support reliable view recall over time. VMem can recall the edited object under revisiting, but the recalled content is blurry and spatially unstable. This indicates that memory retrieval alone does not guarantee accurate local edit preservation when appearance and geometry are not explicitly coordinated.

4.4 Quantitative Comparison

4.4.1 Long-term consistency after global edits

We quantitatively evaluate long-term consistency after global edits by decomposing view recall consistency into structural and semantic consistency. As shown in Table 1, our method achieves

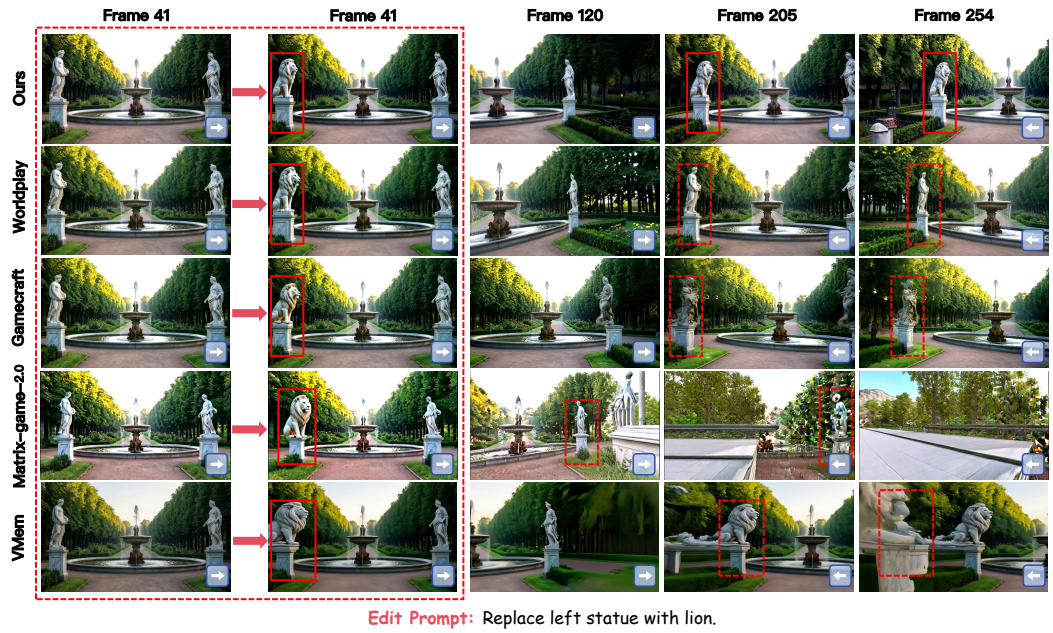

Figure 4: **Qualitative comparison under local edits.** Under a local edit, our method consistently recalls the edited region during revisiting while preserving the surrounding geometric structure.

the best performance in PSNR, SSIM, and LPIPS, indicating strong preservation of geometric structure under global semantic edits. It also significantly outperforms all baselines in semantic consistency (CLIP-Vid), reflecting effective propagation of the edited appearance. In contrast, existing methods either preserve structure but fail to update semantics (e.g., HY-WorldPlay) or degrade in both aspects due to the absence of persistent edit-aware memory (e.g., VMem, HY-GameCraft and Matrix-Game-2.0).

4.4.2 Long-term consistency after local edits

We quantitatively evaluate long-term consistency after local edits in terms of view recall consistency and visual quality. As shown in Table 1, our method achieves the best results in PSNR, SSIM, and LPIPS, indicating accurate recall of edited local content when revisiting the same viewpoints. Moreover, our approach also delivers the best visual quality measured by VBench-Avg. In contrast, baseline methods show obvious degradation in view recall consistency, reflecting their inability to reliably retain and recall local edits over time.

4.5 Ablation Study

We ablate the proposed disentangled context memory by comparing two settings: *w/ Disentangled Contexts* and *w/o Disentangled Contexts*. w/ Disentangled Contexts uses the proposed multi-modal memory design, where retrieved reference contexts can be RGB-only, depth-only, or mixed-modality depending on the edit-aware retrieval strategy. In this setting, semantic appearance and geometric structure are handled separately, allowing the model to reuse geometry while selectively updating outdated appearance information after edits. By contrast, w/o Disentangled Contexts stores all historical contexts only in RGB form and directly reuses them through the same camera-based retrieval index, without separating appearance from geometry or filtering out semantically outdated contexts after edits. As shown in Figure 6, after a global day-to-night edit at Frame 143, *w/ Disentangled Contexts* consistently propagates the updated semantics across subsequent frames and revisited views while preserving stable geometric structure. In contrast, *w/o Disentangled Contexts* repeatedly retrieves outdated RGB contexts, which prevents it from reflecting the latest global edit and leads to progressive semantic inconsistency over time.

4.6 Memory Overhead Analysis

We further analyze the computational overhead introduced by the proposed memory mechanism during long-horizon generation. To quantify this overhead, we profile both the relative runtime ratio

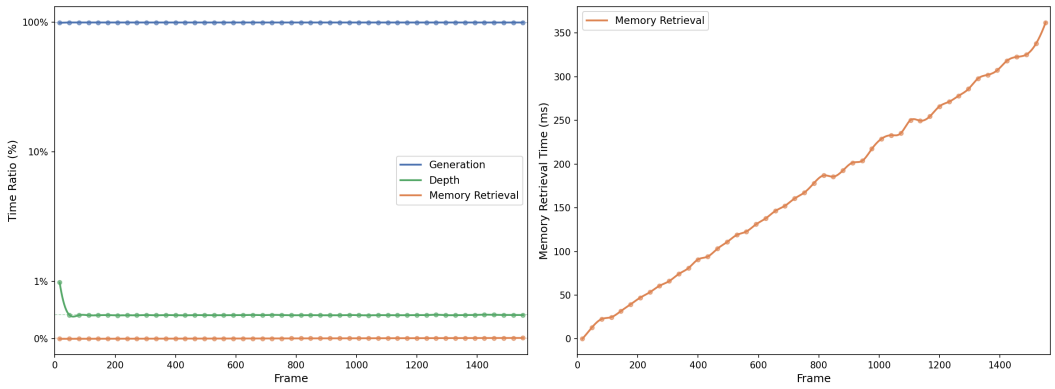

Figure 5: Memory overhead profiling during long-duration generation. Left: component time ratios show that inference is dominated by the video generation backbone, while depth prediction and memory retrieval contribute negligibly. Right: memory retrieval time gradually increases as historical contexts accumulate, but remains at the millisecond level, indicating only marginal runtime overhead.

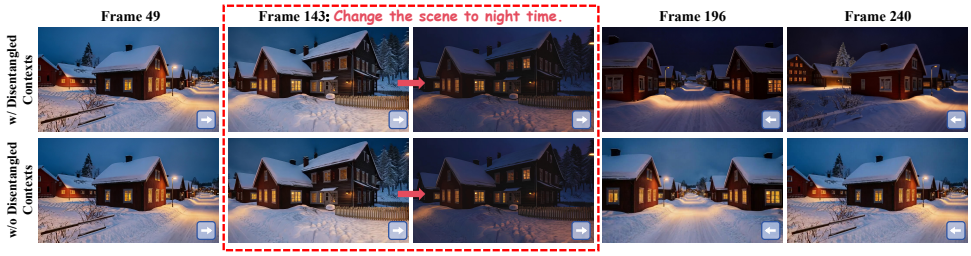

Figure 6: **Ablation study on disentangled context memory.** With disentangled context memory, the model consistently propagates updated global semantics after the edit while preserving stable geometry, whereas entangled RGB contexts reuse outdated semantics, leading to degraded global semantic consistency over time.

of each component and the absolute retrieval time throughout a long generation sequence with a large-loop camera trajectory. Figure 5 shows that the overall runtime is dominated by the video generation backbone. As illustrated in the left plot, the generation module consistently accounts for nearly all inference time, while depth estimation and memory retrieval remain negligible across the entire sequence. This result indicates that the additional operations required for memory maintenance and retrieval contribute only a very small fraction of the total computational cost. The right plot reports the absolute retrieval time as the generation proceeds. As expected, retrieval becomes gradually slower as more historical contexts accumulate in memory. Nevertheless, the growth remains mild, and the retrieval cost stays at the millisecond level even near the end of the sequence, reaching only a few hundred milliseconds. This overhead is negligible relative to the diffusion-based video generation process and does not constitute a practical bottleneck. Overall, these results show that PermaVid scales efficiently to long-video generation by maintaining and reusing disentangled multi-modal contexts with only marginal runtime overhead.

5 Conclusion

In this work, we study how to preserve long-term consistency in video generation across editing operations. To address the limitations of existing memory mechanisms, we propose a multi-modal context memory that disentangles spatial context into semantic appearance and geometric structure, together with an edit-aware memory update and retrieval strategy. By selectively updating and reusing memory across modalities, our approach keeps retrieved context aligned with the latest edited state while preserving reusable geometry. We further build a memory-guided video generation model that performs multi-modal feature fusion over modality-asymmetric reference contexts, enabling consistent generation across time, viewpoints, and edits. Extensive experiments under both global and local edits demonstrate the effectiveness of our approach, showing strong semantic and structural consistency and clear advantages over state-of-the-art methods.

References

[1] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.

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

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

[4] Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory. *arXiv preprint arXiv:2506.05284*, 2025.

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

[6] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

[7] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[8] Jing Tan, Shuai Yang, Tong Wu, Jingwen He, Yuwei Guo, Ziwei Liu, and Dahua Lin. Imagine360: Immersive 360 video generation from perspective anchor. *arXiv preprint arXiv:2412.03552*, 2024.

[9] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

[10] 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.

[11] 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.

[12] Shuai Yang, Jing Tan, Mengchen Zhang, Tong Wu, Gordon Wetzstein, Ziwei Liu, and Dahua Lin. Layerpano3d: Layered 3d panorama for hyper-immersive scene generation. In *Proceedings of the special interest group on computer graphics and interactive techniques conference conference papers*, pages 1–10, 2025.

[13] 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.

[14] 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.

[15] 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.

[16] Lvmin Zhang, Shengqu Cai, Muyang Li, Chong Zeng, Beijia Lu, Anyi Rao, Song Han, Gordon Wetzstein, and Maneesh Agrawala. Pretraining frame preservation in autoregressive video memory compression. *arXiv preprint arXiv:2512.23851*, 2025.

[17] Lvmin Zhang, Shengqu Cai, Muyang Li, Gordon Wetzstein, and Maneesh Agrawala. Frame context packing and drift prevention in next-frame-prediction video diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[18] 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, 2025.

[19] Qingyan Bai, Qiuyu Wang, Hao Ouyang, Yue Yu, Hanlin Wang, Wen Wang, Ka Leong Cheng, Shuailei Ma, Yanhong Zeng, Zichen Liu, et al. Scaling instruction-based video editing with a high-quality synthetic dataset. *arXiv preprint arXiv:2510.15742*, 2025.

[20] 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.

[21] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[23] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. In *International Conference on Learning Representations (ICLR)*, 2023.

[24] Xiaoxue Wu, Bingjie Gao, Yu Qiao, Yaohui Wang, and Xinyuan Chen. Cinetrans: Learning to generate videos with cinematic transitions via masked diffusion models. In *International Conference on Learning Representations*, volume 2026, pages 60198–60230, 2026.

[25] Haonan Qiu, Menghan Xia, Yong Zhang, Yingqing He, Xintao Wang, Ying Shan, and Ziwei Liu. Freenoise: Tuning-free longer video diffusion via noise rescheduling. *arXiv preprint arXiv:2310.15169*, 2023.

[26] Xingyao Li, Fengzhuo Zhang, Jiachun Pan, Yunlong Hou, Vincent YF Tan, and Zhuoran Yang. Enhancing multi-text long video generation consistency without tuning: Time-frequency analysis, prompt alignment, and theory. *arXiv preprint arXiv:2412.17254*, 2024.

[27] Shuochen Chang, Bingjie Gao, Qingyang Liu, Qianli Ma, Yibo Miao, Haonan Zhao, Zhaohe Liao, Xiaofeng Zhang, Jiangtong Li, and Li Niu. From label priors to task evidence: Efficient long-video frame selection via bayesian grpo. *arXiv preprint arXiv:2610.25418*, 2026.

[28] Bingjie Gao, Xinyu Gao, Xiaoxue Wu, Yujie Zhou, Yu Qiao, Li Niu, Xinyuan Chen, and Yaohui Wang. The devil is in the prompts: Retrieval-augmented prompt optimization for text-to-video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3173–3183, 2025.

[29] Do Xuan Long, Xingchen Wan, Hootan Nakhost, Chen-Yu Lee, Tomas Pfister, and Sercan Ö Arık. Vista: A test-time self-improving video generation agent. *arXiv preprint arXiv:2510.15831*, 2025.

[30] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023.

[31] 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.

[32] Xin Ma, Yaohui Wang, Xinyuan Chen, Gengyun Jia, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024.

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

[34] 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*, pages 2568–2577, 2025.

[35] Jiaqi Li, Junshu Tang, Zhiyong Xu, Longhuang Wu, Yuan Zhou, Shuai Shao, Tianbao Yu, Zhiguo Cao, and Qinglin Lu. Hunyuan-gamecraft: High-dynamic interactive game video generation with hybrid history condition. *arXiv preprint arXiv:2506.17201*, 2025.

[36] Xiaofeng Mao, Shaoheng Lin, Zhen Li, Chuanhao Li, Wenshuo Peng, Tong He, Jiangmiao Pang, Mingmin Chi, Yu Qiao, and Kaipeng Zhang. Yume: An interactive world generation model. *arXiv preprint arXiv:2507.17744*, 2025.

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

[38] Shusheng Yang, Jihan Yang, Pinzhi Huang, Ellis Brown, Zihao Yang, Yue Yu, Shengbang Tong, Zihan Zheng, Yifan Xu, Muhan Wang, et al. Cambrian-s: Towards spatial supersensing in video. *arXiv preprint arXiv:2511.04670*, 2025.

[39] Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. *arXiv preprint arXiv:2503.07598*, 2025.

[40] Zhen Li, Chuanhao Li, Xiaofeng Mao, Shaoheng Lin, Ming Li, Shitian Zhao, Zhaopan Xu, Xinyue Li, Yukang Feng, Jianwen Sun, et al. Sekai: A video dataset towards world exploration. *arXiv preprint arXiv:2506.15675*, 2025.

[41] Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Yi Zhang, Chang Zeng, Yanxi Zhou, et al. Spatialvid: A large-scale video dataset with spatial annotations. *arXiv preprint arXiv:2509.09676*, 2025.

[42] Nigel Spivey. *Epic Games*, page 250–263. Dec 2017.

[43] Fangwei Zhong, Kui Wu, Churan Wang, Hao Chen, Hai Ci, Zhoujun Li, and Yizhou Wang. Unrealzoo: Enriching photo-realistic virtual worlds for embodied ai. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5769–5779, 2025.

[44] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025.

[45] Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22831–22840, 2025.

[46] 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.

[47] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

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

[49] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

A Technical appendices and supplementary material

We provide additional materials to the main paper. In Section A.1, we provide the full procedural pipeline of PermaVid. In Section A.2, we describe the UE-Mem dataset in more detail, highlighting its diversity in environment types, visual styles, and long-horizon revisiting trajectories. In Section A.3, we analyze representative failure cases under challenging visual inputs. Section A.4 presents extensive results under both local and global editing operations. Section A.5 reports additional experimental details about the evaluation test set and quantitative results. In Section A.6, we discuss broader limitations of our approach and outline potential directions for future research.

A.1 Pipeline Algorithm

Algorithm 1 summarizes the full PermaVid pipeline. Given an initial observation, target camera trajectories, and a stream of editing operations, PermaVid updates the RGB and depth memory banks according to the edit type, retrieves spatially relevant and edit-valid references, generates the next video chunk, and inserts the generated observations back into memory.

A.2 More Dataset Details

To illustrate the diversity of UE-Mem, the dataset is organized along two axes, environment type (indoor or outdoor) and visual style (realistic or unrealistic), resulting in four complementary categories, as shown in Figure 7. Outdoor scenes range from realistic urban streets and natural landscapes to stylized open environments, supporting large-scale navigation and appearance variation, while indoor scenes include both realistic multi-room interiors and non-photorealistic layouts that emphasize viewpoint revisiting and structural consistency. All scenes are synthesized in Unreal Engine and rendered as long video sequences with accurate camera poses, following navigation-agent-driven trajectories that revisit previously observed locations from diverse viewpoints. Together, these data provide rich supervision for learning memory-guided video generation with strong generalization across structure and appearance.

A.3 Failure Cases

Despite the overall robustness of our approach, we observe occasional failures when the input images exhibit uncommon visual styles or suffer from low image quality as shown in Figure 8. In such cases, the generated videos may show mild temporal jitter or instability, which we attribute to a large domain gap between these inputs and our training data. Nevertheless, for both realistic scenes and common stylized settings, our model maintains strong spatiotemporal consistency, benefiting from the high diversity of scenes covered in our dataset and the proposed two-stage training strategy. Empirically, these failure cases occur infrequently and account for only a small portion of the evaluated samples.

A.4 More Results

Figure 9 and Figure 10 provide a detailed view of how our model adapts its memory behavior under different editing operations. As shown in Figure 9, for local edits such as adding new objects (e.g., furniture or decorations), removing existing instances, or replacing objects with different categories, the model updates semantic memory only in the edited regions while preserving the underlying geometric memory of the scene. This allows newly introduced objects to remain spatially anchored and consistently reappear when the camera revisits the same locations, while unaffected regions

ALGORITHM 1: PermaVid: Edit-aware Multi-modal Memory and Video Generation

Input: Initial RGB observation I_0^{rgb}; initial depth observation I_0^{dep}; initial camera pose $\mathbf{p}_0$; text condition c_{text}; camera trajectory chunks $\{\mathbf{P}^{(k)}\}_{k=1}^{K}$; edit stream $\mathcal{E} = \{e_k\}_{k=1}^{K}$; overlap threshold τ; memory budget B.

Output: Generated video chunks $\{V^{(k)}\}_{k=1}^{K}$ and updated memories $\mathcal{M}^{\mathrm{rgb}}, \mathcal{M}^{\mathrm{dep}}$.

Initialize $\mathcal{M}^{\mathrm{rgb}} \leftarrow \{(I_0^{\mathrm{rgb}}, \mathbf{p}_0, 0, g^*)\}$, $\mathcal{M}^{\mathrm{dep}} \leftarrow \{(I_0^{\mathrm{dep}}, \mathbf{p}_0, 0)\}$, and $g^* \leftarrow 0$;

for $k \leftarrow 1$ **to** K **do**

  Obtain target poses $\mathbf{P}^{(k)}$ and edit $e_k = (T(e_k), \Omega_{e_k}, t_{e_k})$;

  if e_k *is not empty* **then**

    if $T(e_k) = \mathrm{global}$ **then**

      $g^* \leftarrow g^* + 1$;

      $\mathcal{M}^{\mathrm{rgb}} \leftarrow \emptyset$; `// Remove outdated appearance`

      Keep $\mathcal{M}^{\mathrm{dep}}$ unchanged; `// Reuse stable geometry`

    else if $T(e_k) = \mathrm{local}$ **then**

      $\mathcal{M}^{\mathrm{rgb}} \leftarrow \mathcal{M}^{\mathrm{rgb}} \setminus \{m_i^{\mathrm{rgb}} \mid \Pi(\mathbf{p}_i) \cap \Omega_{e_k} \neq \emptyset\}$; `// Invalidate affected RGB contexts`

      if e_k *changes geometry* **then**

        $\mathcal{M}^{\mathrm{dep}} \leftarrow \mathcal{M}^{\mathrm{dep}} \setminus \{m_j^{\mathrm{dep}} \mid \Pi(\mathbf{p}_j) \cap \Omega_{e_k} \neq \emptyset\}$; `// Invalidate affected depth contexts`

      end

    end

  end

  Retrieve RGB candidates: $R_c^{\mathrm{rgb}} \leftarrow \{m_i^{\mathrm{rgb}} \in \mathcal{M}^{\mathrm{rgb}} \mid \mathcal{L}(\mathbf{p}_i, \mathbf{P}^{(k)}) > \tau,\ g_i = g^*\}$;

  Retrieve depth candidates: $R_c^{\mathrm{dep}} \leftarrow \{m_j^{\mathrm{dep}} \in \mathcal{M}^{\mathrm{dep}} \mid \mathcal{L}(\mathbf{p}_j, \mathbf{P}^{(k)}) > \tau\}$;

  $\mathcal{R}^{\mathrm{mem}} \leftarrow \mathrm{GreedySelect}(R_c^{\mathrm{rgb}} \cup R_c^{\mathrm{dep}}, B)$; `// Maximize coverage and reduce redundancy`

  $V^{(k)} \leftarrow \epsilon_\theta\left(c_{\mathrm{text}}, \mathbf{P}^{(k)}, \mathcal{R}^{\mathrm{mem}}\right)$; `// Generate next video chunk`

  $D^{(k)} \leftarrow \mathrm{DepthPredictor}(V^{(k)})$;

  Insert selected RGB frames from $V^{(k)}$ into $\mathcal{M}^{\mathrm{rgb}}$ with semantic flag g^*;

  Insert corresponding depth maps from $D^{(k)}$ into $\mathcal{M}^{\mathrm{dep}}$;

end

return $\{V^{(k)}\}_{k=1}^{K}$, $\mathcal{M}^{\mathrm{rgb}}$, $\mathcal{M}^{\mathrm{dep}}$;

maintain stable appearance and structure over long temporal gaps. In contrast, Figure 10 illustrates global edits including style transformation, seasonal shifts, lighting variation, and weather changes, where the model performs a global semantic memory refresh while fully retaining the original geometric memory. As a result, global attributes such as color tone, illumination, and environmental style are propagated coherently across all viewpoints and time steps, without altering scene layout or object placement. Together, these results demonstrate that our model adapts the scope of memory updates according to edit types while maximizing the effective reuse of historical memories. By leveraging multi-modal context representations together with an edit-aware retrieval and update strategy, the model selectively retains valid memory cues while filtering out outdated modality-specific contexts. This enables more efficient and robust long-term spatiotemporal consistency under both local and global edits.

A.5 Additional Experimental Details

A.5.1 Additional details of the Benchmark test set

To comprehensively evaluate the robustness and generalization of our method, we construct an evaluation test set that spans a wide range of scenes. Specifically, the test set covers both indoor and outdoor environments, and includes realistic as well as unrealistic visual domains, such as anime style scenes, Ghibli style and other stylized renderings. Beyond stylistic differences, the test set further incorporates diverse global appearance changes, including variations in season, weather conditions, and lighting. This broad coverage enables us to assess the model's ability to maintain long-term spatiotemporal consistency under substantial variations in scene layout, semantic appearance, and overall visual conditions.

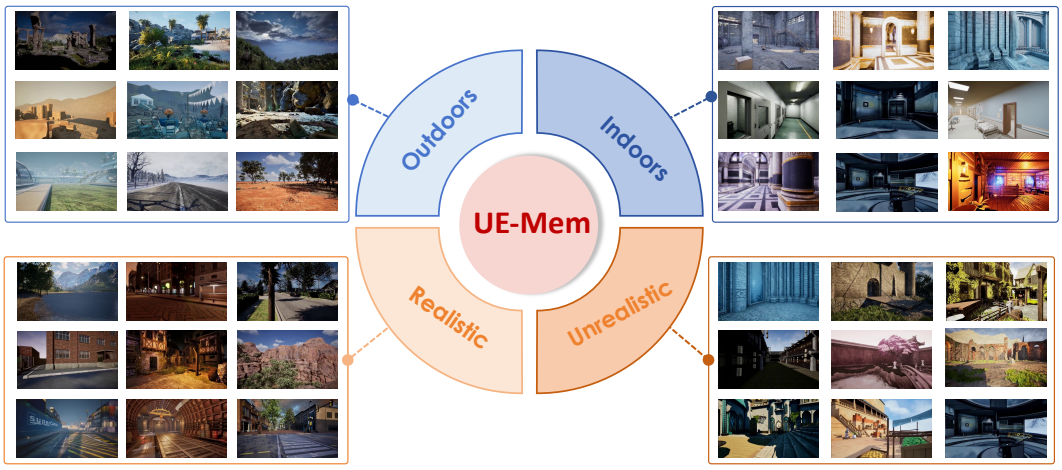

Figure 7: **UE-Mem dataset overview.** UE-Mem is a diverse dataset covering indoor and outdoor scenes with both realistic and stylized appearances, rendered as long videos with revisiting trajectories to support learning under rich structural and visual variations.



Figure 8: **Failure case representation.** Cases showing mild temporal instability under uncommon or low-quality inputs, reflecting limitations from domain gaps, while remaining rare in practice.

For local edit evaluation, we further design a set of structured image editing prompts grouped into three representative categories: add, remove, and replace. These edits involve introducing new objects into the scene, removing existing instances, or replacing objects with different semantic categories, respectively. The resulting test cases provide rich and diverse local editing scenarios, allowing us to systematically evaluate how well the model performs selective semantic updates while preserving geometric structure and unaffected regions. Together, this test set serves as a challenging and comprehensive benchmark for analyzing memory update and reuse behavior under global or local edits.

A.5.2 Additional Quantitative Details

We report detailed quantitative results using VBench to complement our evaluation under global and local edits (Tables 3 and 2). Our method performs strongly on consistency-related metrics, particularly under local edits, where it achieves the best results in subject consistency, background consistency, and motion smoothness, reflecting its ability to preserve object identity, spatial coherence, and temporal stability through selective semantic updates. Under global edits, our approach remains competitive across most metrics, while showing slightly lower imaging and perceptual scores compared to appearance-oriented baselines. This reflects an inherent trade-off, as our model prioritizes long-term spatiotemporal consistency and memory correctness over purely perceptual optimization.

We emphasize that VBench is a general-purpose video generation benchmark focusing on perceptual quality and motion plausibility, which does not fully align with our focus on long-term consistency across viewpoints and edits. Consequently, key advantages of our approach, such as consistent spatial anchoring and robustness under long revisiting trajectories, are not fully captured by VBench metrics alone. Overall, these results complement our qualitative evaluations, confirming that our method excels in consistency-critical aspects while maintaining competitive visual quality.

Table 2: Quantitative comparison under **global edits** on VBench.

Methods	subject consistency	background consistency	motion smoothness	dynamic degree	aesthetic quality	imaging quality
HY-Worldplay [3]	0.8488	0.8489	0.8656	**0.9595**	**1.0000**	<u>0.6712</u>
HY-Gamecraft [35]	**0.8692**	**0.9462**	**0.9438**	<u>0.9593</u>	<u>0.9408</u>	**0.6713**
Matrix-Game-2.0 [46]	0.8421	0.8701	0.8949	0.9531	**1.0000**	0.5893
VMem [15]	0.8236	0.8154	0.8427	0.9418	0.9135	0.5572
Ours	<u>0.8564</u>	<u>0.8956</u>	<u>0.8957</u>	0.9586	**1.0000**	0.6371

Table 3: Quantitative comparison under **local edits** on VBench.

Methods	subject consistency	background consistency	motion smoothness	dynamic degree	aesthetic quality	imaging quality
HY-Worldplay [3]	0.8219	0.8291	<u>0.9558</u>	**1.0000**	0.6505	**0.7349**
HY-Gamecraft [35]	<u>0.8923</u>	<u>0.8824</u>	0.9571	<u>0.9605</u>	**0.6586**	0.6755
Matrix-Game-2.0 [46]	0.8020	0.8190	0.9553	**1.0000**	0.5545	<u>0.7207</u>
VMem [15]	0.7814	0.7936	0.9412	0.9487	0.5298	0.6426
Ours	**0.9394**	**0.8914**	**0.9778**	<u>0.9605</u>	<u>0.6562</u>	0.7014

A.6 Limitations and Future Work

While our method shows strong generalization and robust memory preservation under both local and global edits, it assumes that an edit's spatial impact can be inferred from its semantics. This assumption becomes less reliable in highly challenging scenarios where changes evolve stochastically over time, such as water leakage, fire spread, vegetation growth, or gradual structural degradation, whose spatial influence may expand, shift, or attenuate unpredictably. In these cases, affected regions cannot be accurately bounded in advance, limiting edit-aware memory retrieval and update based on fixed spatial scopes. Addressing such cases will require end-to-end models that jointly learn edit dynamics and memory evolution from long-term observations.

In future work, we plan to explore models that can more accurately handle complex, long-term, and interactive editing scenarios. In parallel, we will further explore memory mechanisms by investigating more expressive geometric representations and adaptive, potentially learnable memory management strategies, so as to better balance memory retrieval, update, and compression when scaling to longer temporal horizons and more complex edits.

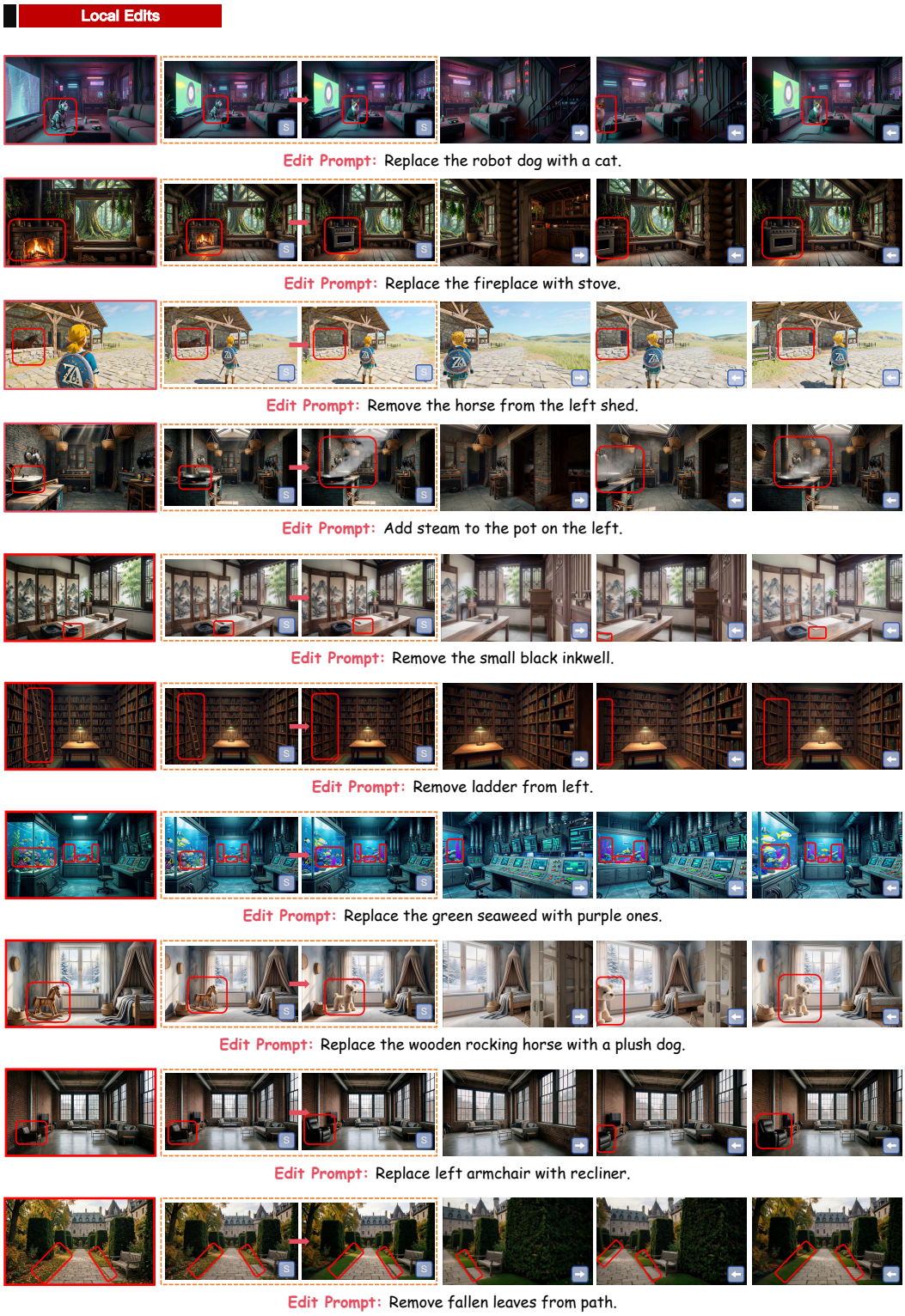

Figure 9: Additional results under local edits, showing localized semantic updates with preserved scene geometry and consistent behavior under revisited views.

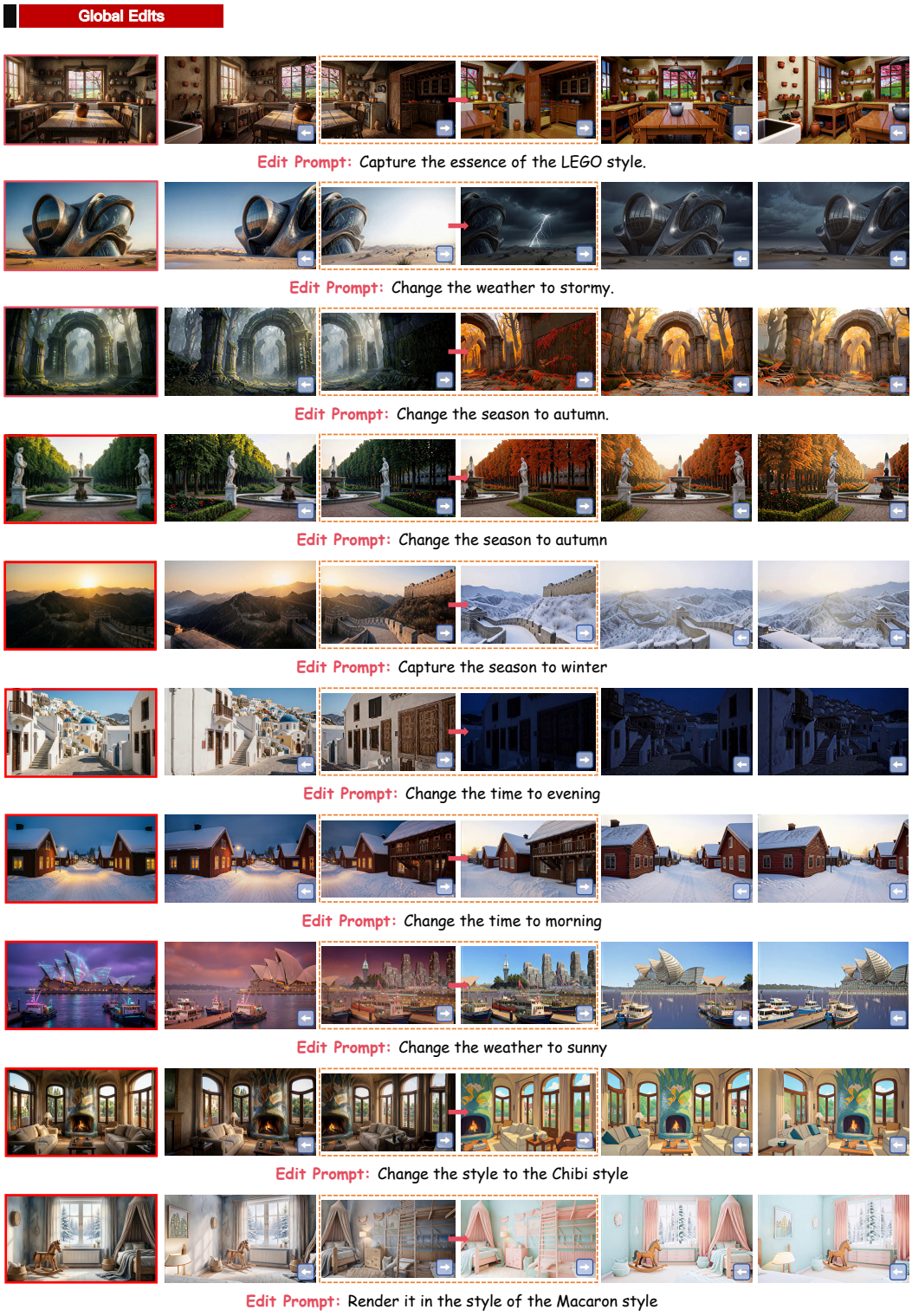

Figure 10: Additional results under global edits, showing coherent propagation of global semantic attributes (e.g., style, illumination, and season) across revisited views with stable scene geometry.

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 introduction clearly state our main contributions: consistent video generation across edits, a disentangled multi-modal context memory that separates semantic appearance from geometric structure, an edit-aware memory update and retrieval strategy, and a memory-guided video generation model. These claims are directly supported by the method and experiments in Sections 1, 3, and 4.

   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: See the appendix section "Limitations and Future Work".

   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 core assumptions underlying our method are: (1) semantic appearance and geometric structure exhibit different temporal behaviors under edits, with appearance changing more frequently while geometry is often reusable; (2) global edits mainly invalidate scene-level semantic appearance while largely preserving geometry; and (3) local edits affect only bounded spatial regions, allowing selective memory invalidation and reuse. Based on these assumptions, the paper specifies the proposed disentangled multi-modal context memory, the edit-aware memory update and retrieval strategy, and the memory-guided video generation formulation in Section 3.

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 and supplement provide the implementation and experimental details needed to reproduce the main results. The source code, UE-Mem dataset, and trained model are publicly available, together with instructions for environment setup, training, inference, and evaluation.

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: [Yes]

   Justification: The source code and reproduction instructions are publicly available at `https://github.com/YS-IMTech/PermaVid`. The UE-Mem dataset is available at `https://huggingface.co/datasets/ysmikey/PermaVid_datasets`, and the trained model is available at `https://huggingface.co/ysmikey/prismworld_ckpt_backup`.

   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: These details are provided in the paper and supplement, and the released code and documentation provide the corresponding environment setup, training, inference, and evaluation procedures.

   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: [N/A]

Justification: Our metrics use standard benchmarks for the tasks we evaluate. These metrics do not include statistical significance.

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: Section 4.1 reports the training setup, including the two-stage training procedure, the model backbone, the video resolution and length, and the use of 32 NVIDIA H200 GPUs for the two-stage training.

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: We reviewed the Code of Ethics and confirm that this research conforms to it.

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: See Sec. 5.

    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: [Yes]

    Justification: Our training and evaluation data is publicly available. Safeguards of the base model, i.e., WAN2.1, are inherited.

    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: See text.

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: The released UE-Mem dataset, source code, and trained model are accompanied by documentation covering environment setup, data format, training, inference, evaluation, and model access.

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 involve crowdsourcing experiments or research with human subjects, so there are no participant instructions, screenshots, or compensation details applicable to this work.

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 involve human subjects research, and therefore no study participants were exposed to risk and no IRB approval was required for this work.

    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: [Yes]

    Justification: LLMs were exclusively used for writing, editing, or formatting purposes.

    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.