# LATENT SPATIAL MEMORY FOR VIDEO WORLD MODELS

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




Figure 1: **Geometrically consistent videos generated by LSM-World with latent spatial memory.** Given an input image and a user-specified camera trajectory (left), LSM-World preserves spatial consistency by caching 3D information directly in the *latent* space, rather than as an RGB point cloud. This design enables spatial memory directly in latent space, avoiding the costly render-and-encode round trip required by prior RGB point-cloud memories. LSM-World can faithfully return to previously observed regions even after large camera movements, while achieving up to $\mathbf{10.57}\times$ faster end-to-end video generation and $\mathbf{55}\times$ lower GPU memory usage than RGB-cache baselines.

## ABSTRACT

Video world models that maintain 3D spatial consistency across generated frames typically rely on explicit point-cloud memory constructed in RGB space. This design is both computationally expensive, requiring repeated rendering and VAE encoding, and inherently lossy, as the round trip through pixel space discards rich features of the learned latent representation. In this paper, we introduce *latent spatial memory* for video world models, a persistent 3D cache that stores scene information directly in the diffusion latent space, avoiding pixel-space reconstruction. Building on this, we propose LSM-World, a latent-space spatial memory framework that constructs the memory by lifting latent tokens into 3D via depth-guided back-projection and queries it by synthesizing novel views through direct latent-space warping. This unified formulation eliminates both the information loss of pixel-space reconstruction and the computational burden of repeated encoding and rendering. Experiments show that latent spatial memory achieves the strongest 3D and photometric consistency on WorldScore and the best closed-loop revisit consistency on RealEstate10K, while matching top baselines in overall score at up to $\mathbf{10.57}\times$ faster generation and $\mathbf{55}\times$ lower cache memory.

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

# 1 INTRODUCTION

Large-scale video diffusion models OpenAI (2024); Wan et al. (2025); Kong et al. (2024); Yang et al. (2024); Blattmann et al. (2023) have demonstrated remarkable ability to synthesize photorealistic sequences, motivating their use as world simulators that internalize visual dynamics and generate plausible future observations conditioned on camera trajectories or actions Bruce et al. (2024); Valevski et al. (2024); Alonso et al. (2024). A central challenge in this paradigm is maintaining long-term scene consistency: without explicit spatial memory, even powerful generators accumulate geometric drift, producing frames that are individually convincing but collectively inconsistent when projected into a shared world coordinate system. For example, when revisiting a previously generated region, the visual content is likely to have changed.

A natural remedy is to equip the generator with a persistent 3D memory that tracks what has been observed. As shown in Figure 2, recent world-generation systems Zhao et al. (2026); Huang et al. (2025b); Yu et al. (2025; 2024a); Wang et al. (2025b); Duan et al. (2026); Wu et al. (2025); Chou et al. (2025); Wang et al. (2026c) adopt this strategy by maintaining an explicit point cloud in RGB space that is rendered and re-encoded at every step. While effective at enforcing scene consistency, this pipeline introduces two fundamental bottlenecks. As shown in Figure 2(c), first, the repeated round trip between latent and pixel space is computationally prohibitive, as rendering millions of colored points at full resolution and re-encoding them dominates wall-clock time. Second, the RGB detour does not preserve the model's native latent conditioning features. RGB point-cloud memories render a target-view image and then re-encode it into a latent tensor, which may be distorted by VAE reconstruction error and distribution mismatch in the resulting conditioning features.

To address these bottlenecks, we propose **latent spatial memory**, a persistent 3D cache that stores the diffusion model's latent features at world-space locations instead of RGB colors. As illustrated in Figure 2, an observed frame is first encoded into VAE latent space, and each latent-grid cell is lifted into 3D by depth-guided back-projection. Each memory element thus pairs a world-space coordinate with the corresponding frame latent token. At readout time, the cache is projected directly onto the target camera grid at latent resolution with depth-aware visibility handling, yielding a target-view latent tensor in the same space consumed by the diffusion backbone. This avoids both pixel-resolution rendering of the cache and per-step VAE encoding of the rendered image, eliminating the two main bottlenecks of RGB point-cloud memories.

Building on this representation, we present LSM-World, a video world model that generates long, geometrically consistent rollouts chunk by chunk through an initialize-readout-update latent memory cycle as shown in Figure 3. First, LSM-World initializes the latent spatial memory by encoding the initial frame and back-projecting its latent tokens into world-space coordinates via depth-guided lifting Lin et al. (2026); Wang et al. (2026a;b; 2025a; 2026d). For each subsequent chunk, it reads from memory by projecting the cache onto every target camera pose, producing latent-space feature tensors that are injected into the diffusion backbone through a ControlNet-style side branch Zhang et al. (2023). After decoding the denoised latents into output frames, LSM-World updates the memory by segmenting dynamic objects Carion et al. (2025), estimating depth, and back-projecting the corresponding latent tokens into the cache to preserve geometric coherence. This cycle repeats over chunks to support long-horizon generation.

Extensive experiments validate the benefits of latent spatial memory in both generation quality and efficiency. Across WorldScore Duan et al. (2025) and RealEstate10K Zhou et al. (2018), LSM-World exceeds strong baselines in overall quality while leading on the 3D and photometric consistency metrics. Crucially, it attains this quality while running up to $\mathbf{10.57}\times$ faster end-to-end and using up to $\mathbf{55}\times$ less GPU memory in the 3D cache than RGB point-cloud readout, making world-consistent generation practical for long trajectories.

Our main contributions can be summarized as follows: First, we introduce *latent spatial memory*, a 3D memory for video world models that operates entirely in latent space, avoiding heavy pixel-space conversion. Second, we develop a data-processing pipeline for collecting video clips with latent spatial memory. Building on this, we propose LSM-World, a video world model featuring a ControlNet-style conditioning branch and a two-stage training strategy. Third, LSM-World delivers up to $\mathbf{10.57}\times$ end-to-end generation speedup and $\mathbf{55}\times$ lower GPU memory usage, while matching or exceeding strong baselines on WorldScore and RealEstate10K.

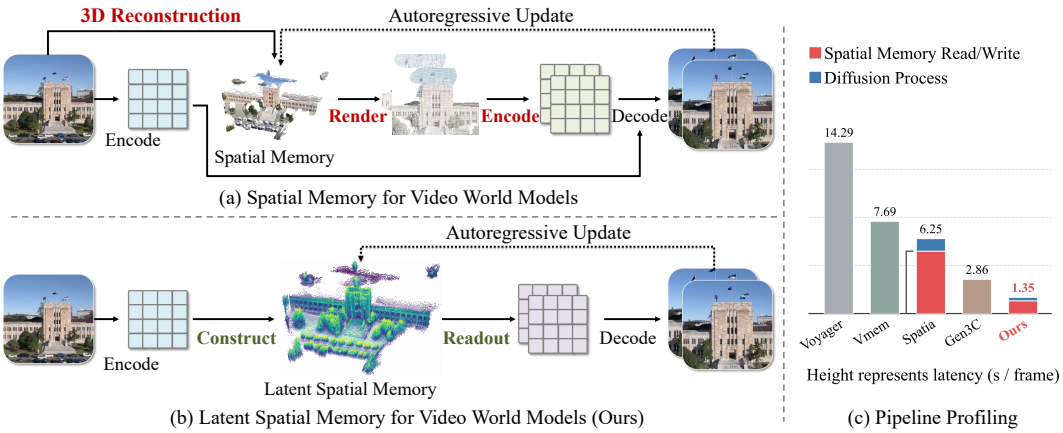


Figure 2: **Latent spatial memory vs. RGB point-cloud-based memory.** (a): prior systems store memory in RGB point clouds and pay a render-and-encode round trip at every conditioning step. (b): latent spatial memory is stored in diffusion latent space and can be read through simple projections, eliminating the per-step pixel-space detour and shrinking the cache footprint by the squared VAE compression factor. (c): pipeline profiling shows that maintaining explicit memory is a major efficiency bottleneck during long-horizon generation.

## 2 Related Work

### 2.1 Video Diffusion Models

Diffusion and flow-matching frameworks Lipman et al. (2022); Esser et al. (2024) have quickly moved from high-fidelity image synthesis to temporally coherent video generation. Early video methods extended image architectures with temporal attention or 3D convolution layers to capture inter-frame dependencies Blattmann et al. (2023); Guo et al. (2023); Chen et al. (2024b). Later systems scaled this paradigm to photorealistic, multi-second outputs by pretraining on large, diverse video corpora OpenAI (2024); Kong et al. (2024); Yang et al. (2024); Xu et al. (2024); Zhou et al. (2024); Fan et al. (2025). A common strategy works in a compressed VAE latent space, which cuts the cost of denoising and enables higher resolutions and longer durations Blattmann et al. (2023); HaCohen et al. (2024); Wan et al. (2025). Recent autoregressive and streaming extensions further target longer sequences Chen et al. (2024a); Henschel et al. (2024); Gu et al. (2025); Chen et al. (2025a); Xie et al. (2024). Despite this progress, most video diffusion models remain fundamentally two-dimensional: frames are synthesized in sequence or in parallel with no explicit model of the underlying 3D scene geometry. As a result, generated videos often show geometric drift, parallax violations, and inconsistent scene structure under large camera motions or long time horizons.

### 2.2 Camera-controllable Video Generation

A parallel line of work conditions video diffusion models on explicit camera trajectories for controllable viewpoint changes. Representative methods use extra control modules He et al. (2024), attention mechanisms Yu et al. (2024b); Zhou et al. (2025), or 3D-aware rendering signals as conditioning inputs Ren et al. (2025). These approaches give fine-grained camera control within a single generation pass, but they keep no persistent scene representation across steps. Each clip is generated independently of earlier content, so revisiting a region or extending a trajectory beyond the model's context window can introduce inconsistencies. Without spatial memory, they are ill-suited to long-horizon, exploratory world generation, where 3D coherence must hold across many sequential rounds.

### 2.3 Spatial Memory for Video Generation

To keep scenes consistent, a growing body of work adds explicit spatial memory that holds a 3D scene representation across generation steps. A representative strategy lifts generated RGB frames

into point clouds using estimated depth, accumulates them into a persistent 3D cache, and renders target-view conditioning images at each step Yu et al. (2024a); Engstler et al. (2024); Yu et al. (2025); Huang et al. (2025b); Zhao et al. (2026); Chen et al. (2025b). More recent systems add surfel-indexed view memory Li et al. (2025) to improve temporal consistency over long sequences. These methods clearly improve multi-view coherence, since the 3D cache anchors each new frame to a shared geometric scaffold. However, existing spatial memory designs work entirely in RGB pixel space, which is computation-intensive and prone to errors that accumulate over repeated encoding and decoding. This motivates our approach, which builds and queries spatial memory entirely in the diffusion model's latent space, preserving representational fidelity while removing the rendering and re-encoding overhead that limits prior methods.

# 3 PRELIMINARIES

A video world model synthesizes a multi-view-consistent sequence $\{I^t\}_{t=1}^T$ from an initial frame $I^0$ and camera trajectory $\{(\mathbf{E}^t, K^t)\}$, using a pretrained latent video diffusion backbone with encoder $\mathcal{E}$ and decoder $\mathcal{D}$. Autoregressive chunk generation can accumulate geometric drift when the camera revisits an observed region. To provide persistent spatial context, prior systems attach an ***RGB point cloud*** Zhao et al. (2026); Huang et al. (2025b); Yu et al. (2025)

$$\mathcal{M}_{\text{rgb}} = \big\{(\mathbf{p}_i, \mathbf{c}_i)\big\}, \qquad \mathbf{p}_i \in \mathbb{R}^3,\ \mathbf{c}_i \in [0,1]^3, \tag{1}$$

that stores the color $\mathbf{c}_i$ of each observed surface point $\mathbf{p}_i$. At target pose $(\mathbf{E}^t, K^t)$, this memory is rasterized into an RGB image and re-encoded into the denoiser input,

$$\hat{\mathbf{z}}^t = \mathcal{E}\big(\text{Rasterize}(\mathcal{M}_{\text{rgb}};\, \mathbf{E}^t, K^t)\big), \tag{2}$$

where $\text{Rasterize}(\cdot)$ denotes $z$-buffered rendering. Pixel-resolution rendering and VAE encoding make each read increasingly expensive as memory grows, while the RGB detour introduces reconstruction error and latent-distribution mismatch.

# 4 METHOD

## 4.1 OVERVIEW

As shown in Figure 3, LSM-World initializes persistent latent memory $\mathcal{M}$ and iteratively performs readout-update cycles over video chunks. **Initialization:** the initial frame $I^0$ is encoded by $\mathcal{E}$ and lifted into 3D space via a downsampled depth map, seeding $\mathcal{M}$ at latent grids (Section 4.2). **Readout and denoising:** For each chunk, $\mathcal{M}$ is projected onto the target view as control signals, which are fed into the diffusion backbone through a ControlNet-style side branch, keeping spatial memory working entirely in latent space (Section 4.3). We provide visualizations of memory readouts in A. **Memory update:** the newly generated frames are back-projected to update $\mathcal{M}$ in the rollout (Section 4.4). Crucially, $\mathcal{M}$ is queried directly in latent space, eliminating the redundant render-and-encode loop required by RGB point-cloud memories and reducing both readout cost and memory footprint.

## 4.2 LATENT SPATIAL MEMORY INITIALIZATION

We represent the memory as a set of latent-attributed 3D points

$$\mathcal{M} = \big\{(\mathbf{p}_i, \mathbf{f}_i)\big\}, \qquad \mathbf{p}_i \in \mathbb{R}^3,\ \mathbf{f}_i \in \mathbb{R}^C, \tag{3}$$

where each point $i$ pairs a world-space coordinate $\mathbf{p}_i$ with a latent feature $\mathbf{f}_i$ drawn directly from the VAE encoder, matching the native input space of the diffusion backbone.

**Construction.** Given a latent frame $\mathbf{z} \in \mathbb{R}^{C\times h\times w}$, metric depth map $D$, intrinsics $K$, and camera pose $\mathbf{E}$ from a feed-forward reconstructor Lin et al. (2026), we first downsample $D$ to the latent grid and rescale $K$ accordingly; in what follows we use $D$ and $K$ to refer to their latent-resolution versions. Each latent cell $(u, v)$ is then back-projected into world space, thereby producing one latent memory element for each cell:

$$\mathbf{p}_{uv} = \pi^{-1}(u, v, D(u,v);\, K, \mathbf{E}), \qquad \mathbf{F}_{uv} = \mathbf{z}[:, v, u], \tag{4}$$

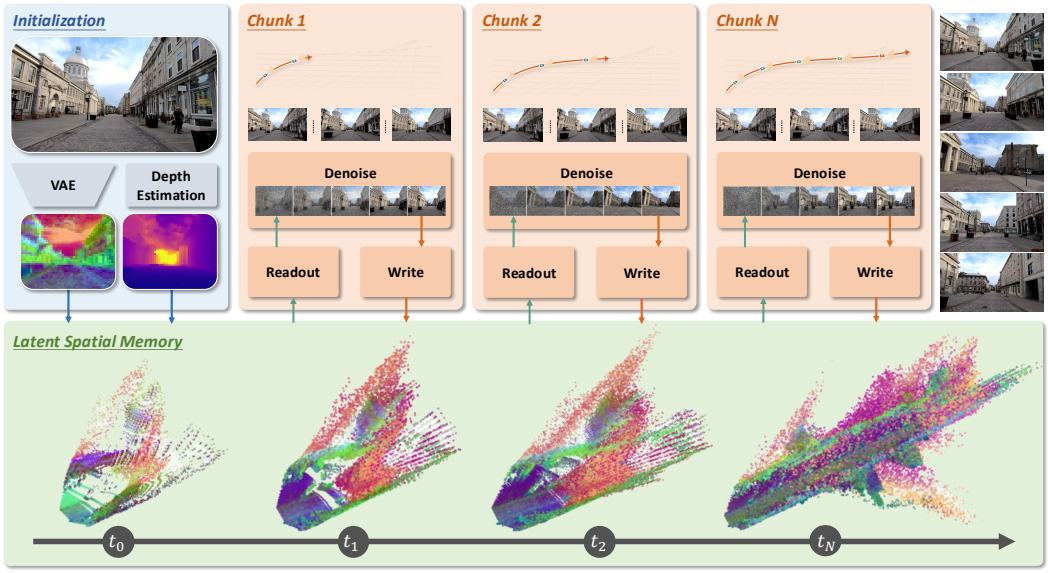


Figure 3: **Overview of LSM-World.** LSM-World initializes a 3D latent memory from $I_0$ by encoding it into VAE latents and lifting them with depth-guided back-projection. At each target view, the memory is read through a latent-resolution projection, and generation proceeds chunk by chunk: each decoded output chunk is re-estimated for depth, re-encoded into latents, and back-projected to extend the memory throughout the rollout.

where $\pi^{-1}$ denotes standard pinhole back-projection (see the accompanying appendix for details) and $\mathbf{F}_{uv} \in \mathbb{R}^C$ denotes the latent token stored at the memory element anchored at $\mathbf{p}_{uv}$. The initial memory is seeded by applying Equation 4 to the encoded latent of $I^0$, and is subsequently updated at every generation step by the autoregressive memory-update procedure described below.

### 4.3 Latent Spatial Memory Readout

Given a target view $(\mathbf{E}^t, K^t)$, we query the latent memory $\mathcal{M}$ by projecting all memory points onto the target camera grid at latent resolution. For each target cell, we retain the frontmost projected point using $z$-buffering and retrieve its associated latent token:

$$i^t(u,v) = \underset{i \in \Omega^t(u,v)}{\arg\min} \left[\mathbf{E}^t \mathbf{p}_i\right]_z, \qquad \hat{\mathbf{z}}^t(u,v) = \mathbf{F}_{i^t(u,v)}. \tag{5}$$

Here $\Omega^t(u,v)$ denotes the set of memory points that project to latent cell $(u,v)$ with positive depth under the target view $(\mathbf{E}^t, K^t)$, and $[\cdot]_z$ extracts the depth coordinate. Cells with $\Omega^t(u,v) = \emptyset$ are zero-filled. We also produce a binary visibility mask $\mathbf{m}^t \in \{0,1\}^{h \times w}$ indicating which cells receive at least one projected point. This mask allows the denoiser to distinguish genuinely unseen regions from observed regions whose latent feature is zero.

This readout preserves the stored conditioning signal: when the memory is constructed from a source frame and queried from the same view, Equation 5 retrieves the corresponding source-view latent tokens on visible cells, up to discretization and occlusion. To generate each chunk, the latent memory readouts $\hat{\mathbf{z}}^t$ and visibility masks $\mathbf{m}^t$ are concatenated and passed to a ControlNet-style side branch Zhang et al. (2023), which injects memory features into the video diffusion backbone. Besides, segment-aware rotary encodings are also leveraged to mark noisy target, clean preceding, and clean reference frames in a single forward pass. Because the readouts already lie in the backbone's latent space, no redundant render-and-encode loops are needed, unlike in the pixel-space pipelines used by RGB point-cloud memories.

Table 1: **Evaluation Results on WorldScore Duan et al. (2025).** The **Average Score** column is the average of the Static Score and Dynamic Score, while the remaining metrics are computed according to the WorldScore benchmark protocol.

| Method | **Average Score** | **Static Score** | **Dynamic Score** | Camera Ctrl | Object Ctrl | Content Align | 3D Const | Photo Const | Style Const | Subject Quality | Motion Acc | Motion Mag | Motion Smooth |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Models with 3D Cache* | | | | | | | | | | | | | |
| WonderJourney | 54.19 | 63.75 | 44.63 | 84.60 | 37.10 | 35.54 | 80.60 | 79.03 | 62.82 | 66.56 | - | - | - |
| InvisibleStitch | 51.95 | 61.12 | 42.78 | **93.20** | 36.51 | 29.53 | 88.51 | 89.19 | 32.37 | 58.50 | - | - | - |
| WonderWorld | 61.79 | 72.69 | 50.88 | 92.98 | 51.76 | 71.25 | 86.87 | 85.56 | 70.57 | 49.81 | - | - | - |
| Voyager | 66.08 | **77.62** | 54.53 | 85.95 | 66.92 | 68.92 | 81.56 | 85.99 | 84.89 | **71.09** | - | - | - |
| FlashWorld | 60.23 | 70.85 | 49.60 | 84.43 | 50.28 | 56.54 | 85.87 | 86.72 | 79.36 | 52.75 | - | - | - |
| LucidDreamer | 59.84 | 70.40 | 49.28 | 88.93 | 41.18 | **75.00** | 90.37 | 90.20 | 48.10 | 58.99 | - | - | - |
| Spatia | 69.73 | 72.63 | 66.82 | 75.66 | 52.32 | 69.95 | 86.40 | 89.10 | 80.09 | 54.86 | 54.83 | 24.75 | 80.26 |
| *General Video Models* | | | | | | | | | | | | | |
| VideoCrafter2 | 50.03 | 52.57 | 47.49 | 28.92 | 39.07 | 72.46 | 65.14 | 61.85 | 43.79 | 56.74 | 47.12 | 30.40 | 29.39 |
| EasyAnimate | 52.25 | 52.85 | 51.65 | 26.72 | 54.50 | 50.76 | 67.29 | 47.35 | 73.05 | 50.31 | 75.00 | 31.16 | 40.32 |
| Allegro | 53.64 | 55.31 | 51.97 | 24.84 | 57.47 | 51.48 | 70.50 | 69.89 | 65.60 | 47.41 | 54.39 | **40.28** | 37.81 |
| CogVideoX-I2V | 60.64 | 62.15 | 59.12 | 38.27 | 40.07 | 36.73 | 86.21 | 88.12 | 83.22 | 62.44 | 69.56 | 26.42 | 60.15 |
| Vchitect-2.0 | 40.38 | 42.28 | 38.47 | 26.55 | 49.54 | 65.75 | 41.53 | 42.30 | 25.69 | 44.58 | 33.59 | 33.81 | 21.31 |
| LTX-Video | 55.99 | 55.44 | 56.54 | 25.06 | 53.41 | 39.73 | 78.41 | 88.92 | 53.50 | 49.08 | **76.22** | 29.95 | 71.09 |
| Wan2.1 | 55.21 | 57.56 | 52.85 | 23.53 | 40.32 | 45.44 | 78.74 | 78.36 | 77.18 | 59.38 | 54.27 | 33.26 | 38.05 |
| LSM-World (Ours) | **70.36** | 73.60 | **67.11** | 55.36 | **74.17** | 42.09 | **92.21** | **93.95** | **96.91** | 60.50 | 51.36 | 24.18 | **80.36** |

### 4.4 Autoregressive Memory Update

After each chunk is generated, LSM-World updates $\mathcal{M}$ with the newly observed scene content. We estimate depth and camera parameters for the generated frames using a feed-forward reconstructor Lin et al. (2026) and back-project the corresponding latent tokens into the memory using Equation 4. Then, the updated memory can be denoted as:

$$\mathcal{M} \leftarrow \mathcal{M} \cup \left\{(\mathbf{p}_{uv}, \mathbf{F}_{uv})\right\}_{(u,v)\in\Lambda^t}, \tag{6}$$

where $\Lambda^t$ contains latent cells with valid depth outside dynamic objects and sky regions, detected by an open-vocabulary entity extractor Yang et al. (2025) and a video segmenter Carion et al. (2025). This filtering prevents transient or geometrically unreliable content from contaminating the persistent memory. The current chunk latents are also carried forward as short-term temporal context for conditioning the next generated chunk.

### 4.5 Efficient Adaptation to Existing Diffusion Models

We adapt a pretrained camera-controllable video diffusion transformer Wan et al. (2025); Zhao et al. (2026) in two stages. In the first stage, we freeze the backbone and VAE and train only the ControlNet-style side branch, aligning latent memory readouts with the backbone feature space without perturbing the pretrained generative prior. In the second stage, we freeze the side branch and attach rank-32 LoRA adapters Hu et al. (2022) ($\alpha = 32$) to the backbone. We optimize only these adapters, enabling lightweight adaptation to the memory condition while preserving the backbone's appearance and motion priors. Both stages use the flow-matching objective Lipman et al. (2022) on the target frames.

## 5 Experiments

### 5.1 Experimental Setup

**Datasets and baselines.** We train on a corpus of videos from RealEstate10K Zhou et al. (2018), with depth and camera poses annotated by Huang et al. (2025a); Lin et al. (2026) and dynamic regions removed as in Section 4.4. Evaluation is performed on WorldScore Duan et al. (2025), which reports ten metrics covering controllability, consistency, quality, and motion, and on RealEstate10K, which provides paired ground truth for novel view synthesis and supports the closed-loop protocol of Spatia Zhao et al. (2026). On WorldScore we compare against scene generators based on RGB point




Figure 4: **Closed-loop revisit comparison on RealEstate10K.** In the closed-loop test, the camera trajectory gradually returns to its starting point. The comparison between the last frame and the input frame shows that LSM-World maintains strong consistency under the revisit setting.

Table 2: **Evaluation Results on RealEstate10K Zhou et al. (2018).** We report novel-view synthesis and closed-loop results. Following Spatia Zhao et al. (2026), closed-loop evaluation measures the similarity between the initial frame and the final frame after generating a return trajectory.

| Method | Novel View Synthesis | | | Closed-Loop | | |
|---|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | $\text{PSNR}_C$ ↑ | $\text{SSIM}_C$ ↑ | $\text{LPIPS}_C$ ↓ |
| SEVA Zhou et al. (2025) | 13.07 | 0.515 | 0.445 | - | - | - |
| VMem Li et al. (2025) | 14.62 | 0.522 | 0.426 | - | - | - |
| ViewCrafter Yu et al. (2024b) | 15.78 | 0.580 | 0.396 | 14.79 | 0.481 | 0.365 |
| FlexWorld Chen et al. (2025b) | 16.25 | 0.593 | 0.370 | 12.20 | 0.428 | 0.598 |
| Voyager Huang et al. (2025b) | 17.79 | 0.636 | 0.297 | 17.66 | 0.540 | 0.380 |
| Spatia Zhao et al. (2026) | **18.58** | <u>0.646</u> | <u>0.254</u> | <u>19.38</u> | <u>0.579</u> | **0.213** |
| LSM-World (Ours) | <u>18.38</u> | **0.779** | **0.250** | **20.05** | **0.825** | <u>0.228</u> |

clouds Yu et al. (2024a); Engstler et al. (2024); Yu et al. (2025); Huang et al. (2025b) and state-of-the-art foundation video generators Chen et al. (2024b); Xu et al. (2024); Zhou et al. (2024); Yang et al. (2024); Fan et al. (2025); HaCohen et al. (2024); Wan et al. (2025); Zhao et al. (2026). On RealEstate10K we additionally compare against the view memory baselines SEVA Zhou et al. (2025) and VMem Li et al. (2025), and against 3D-aware video generators ViewCrafter Yu et al. (2024b) and FlexWorld Chen et al. (2025b).

**Implementation and evaluation.** Our backbone is Wan2.2-TI2V-5B Wan et al. (2025), whose VAE has a compression ratio of $4\times16\times16$ and latent channel count $C = 48$. Each generation chunk contains nine latent frames at resolution $44 \times 80$, corresponding to 33 RGB frames at $704 \times 1280$. Training was conducted on 32 A100 GPUs with a global batch size of 64. The ControlNet-style side branch is initialized from its corresponding blocks of Wan2.2 and trained with an AdamW optimizer, bfloat16 mixed precision, and the flow-matching objective Lipman et al. (2022) on target frames. At inference, we use the UniPC flow scheduler with 40 sampling steps. We evaluate generation quality using the WorldScore Average Score and its constituent metrics, as well as PSNR, SSIM, LPIPS, and the closed-loop metrics $\text{PSNR}_C$, $\text{SSIM}_C$, and $\text{LPIPS}_C$ on RealEstate10K following Spatia. Efficiency is measured on a single NVIDIA H100 using wall-clock time and peak GPU memory for one cache read across progressively longer rollouts.

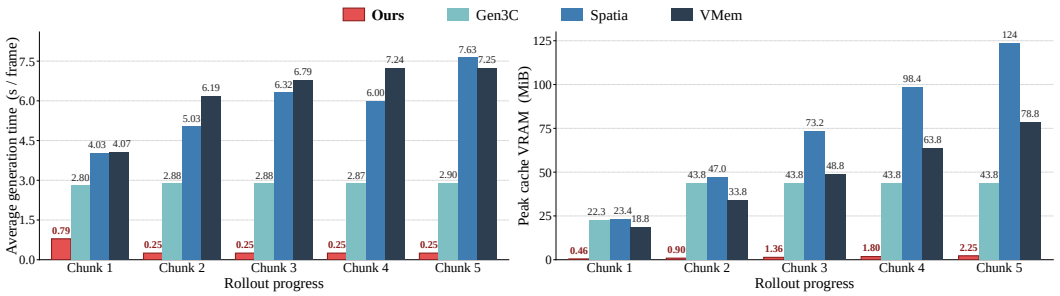


Figure 5: **Efficiency scaling with rollout progress.** Average cache-read time (*left*) and peak cache footprint (*right*) measured on a single NVIDIA H100 across five rollout chunks. After the first chunk pays for a one-off setup pass, LSM-World settles at a per-frame cost of $0.25$ s and a cache footprint that grows by less than $0.5$ MiB per chunk. RGB-cache baselines require an order of magnitude more memory and one-to-two orders of magnitude more time per frame.




Figure 6: **Open-domain video comparison.** Generations on out-of-domain outdoor and natural scenes that differ substantially from the training distribution. LSM-World generalizes effectively, producing temporally smooth and 3D-consistent rollouts under aggressive camera motion, whereas RGB point-cloud baselines exhibit stretched textures on unseen layouts and severe visual artifacts.

### 5.2 Main Results

We first demonstrate that LSM-World matches or exceeds prior work on generation quality, then show how it does it far more efficiently.

**World generation on WorldScore.** Table 1 summarizes results on WorldScore. LSM-World attains the highest Average Score of all compared baselines, improving over RGB point-cloud memory and clearly outperforming foundation video generators that lack a persistent spatial memory. The advantage is strongest on the dynamic partition, and LSM-World leads on 3D and photometric consistency on the static axis. We attribute this to keeping spatial memory in the video diffusion feature space, avoiding the information loss of converting to and from RGB. Figure 6 shows side-by-side generations on out-of-domain prompts, where LSM-World maintains 3D coherence on scenes that lie far from the RealEstate10K training distribution while RGB-cache and memory-free baselines exhibit visible drift.

**Novel view synthesis and closed-loop consistency.** Table 2 reports results on RealEstate10K. In the standard novel view synthesis setting, LSM-World achieves the best SSIM and LPIPS, while remaining close to Spatia on PSNR, surpassing both view memory baselines and all 3D-aware video generators. The improvements in SSIM and LPIPS indicate that latent spatial memory effectively preserves structural and textural information. The closed-loop setting is particularly informative because it amplifies long-horizon drift. Figure 4 visualizes this return trajectory. LSM-World achieves

Table 3: **Ablation studies and depth-source sensitivity on a WorldScore split.** Each row disables a single component of LSM-World or swaps the depth predictor, while all other settings remain fixed.

| Variant / depth source | Avg ↑ | Static ↑ | Dynamic ↑ | 3D Cons ↑ | Photo Cons ↑ |
|---|---|---|---|---|---|
| *Component ablations* | | | | | |
| LSM-World (full) | **70.36** | **73.60** | **67.11** | **92.21** | **93.95** |
| Explicit RGB Point Cloud | <u>67.71</u> | <u>70.49</u> | <u>64.93</u> | <u>90.75</u> | <u>91.10</u> |
| Feature Upsampling | 60.85 | 62.41 | 59.28 | 84.90 | 79.81 |
| No Dynamic Object Filter | 61.20 | 62.69 | 59.70 | 80.88 | 76.10 |
| *Depth source ablations* | | | | | |
| MapAnything Keetha et al. (2025) | 69.66 | 72.78 | 66.53 | 91.89 | 93.32 |
| UniDepth Piccinelli et al. (2024) | 69.13 | 72.15 | 66.10 | 91.63 | 92.79 |

the best $\text{PSNR}_C$ and $\text{SSIM}_C$ under this protocol, while remaining second-best on $\text{LPIPS}_C$, indicating that latent spatial memory anchors the generator to a coherent geometric representation even after the camera has left. We provide more visual comparisons in Appendix B, confirming that latent spatial memory can match or even surpass a well-tuned RGB cache while being far more efficient.

**Efficiency.** At the quality level established above, latent spatial memory delivers its main payoff: it eliminates the per-step pixel-space round trip. Figure 5 plots the wall-clock time of a single cache read and the peak GPU memory required to maintain the cache, against RGB-cache baselines that re-render the accumulated point cloud and re-encode it through the VAE at every step. Both quantities grow slowly for latent spatial memory, because the read operates at the native latent resolution and the cache is smaller by a factor of $s^2$ per spatial dimension. The RGB cache instead grows rapidly along both axes, since per-step rendering scales with $HW$ rather than $hw$ and the dense point cloud expands roughly linearly with the number of anchored frames. At matched rollout length, end-to-end video generation is $10.57\times$ faster than the RGB pipeline and uses $55\times$ less GPU memory for the cache.

### 5.3 Ablation Studies

We isolate the contribution of each component of LSM-World on a split of WorldScore, as reported in Table 3. **Latent vs. RGB memory.** Replacing the latent cache with an RGB point cloud from the same source frames lowers the Average Score and weakens 3D and photometric consistency, supporting the benefit of preserving native latent features. **Feature downsampling vs. geometry downsampling.** Upsampling latent features to pixel resolution before lifting degrades 3D consistency; matching geometry to the native latent grid is both cheaper and more faithful to the generator. **Dynamic object filtering.** Removing the dynamic-object masks degrades long-horizon stability, as stale content persists in memory and is projected into later chunks. **Depth source.** Swapping DepthAnything 3 for MapAnything or UniDepth yields only a modest drop, indicating that the method remains robust across the tested depth estimators.

## 6 Conclusion

We propose *latent spatial memory*, which operates entirely and efficiently in the diffusion latent space, and build LSM-World upon it. Across WorldScore and RealEstate10K, LSM-World achieves state-of-the-art or competitive quality while delivering up to $10.57\times$ faster generation and $55\times$ lower cache memory than RGB-cache baselines.

**Limitations and future work.** The dynamic-region filter excludes moving entities from persistent memory because their geometry is unreliable; extending the cache to preserve dynamic objects is an important direction for future work.

### AI USE STATEMENT

AI tools were used to support literature research and to correct grammar and improve the language of the manuscript. The authors reviewed and verified all AI-assisted content and remain fully responsible for the paper.

### REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide detailed descriptions of the training procedure, dataset usage, and computational resources in the appendix C. The supplementary materials also include a complete anonymized codebase.

## REFERENCES

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

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

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, 2024a.

Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Juncheng Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengchen Ma, et al. Skyreels-v2: Infinite-length film generative model. *arXiv preprint arXiv:2504.13074*, 2025a.

Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7310–7320, 2024b.

Luxi Chen, Zihan Zhou, Min Zhao, Yikai Wang, Ge Zhang, Wenhao Huang, Hao Sun, Ji-Rong Wen, and Chongxuan Li. Flexworld: Progressively expanding 3d scenes for flexiable-view synthesis. *arXiv preprint arXiv:2503.13265*, 2025b.

Yu-Cheng Chou, Xingrui Wang, Yitong Li, Jiahao Wang, Hanting Liu, Cihang Xie, Alan Yuille, and Junfei Xiao. Captain safari: A world engine with pose-aligned 3d memory. *arXiv preprint arXiv:2511.22815*, 2025.

Haoyi Duan, Hong-Xing Yu, Sirui Chen, Li Fei-Fei, and Jiajun Wu. Worldscore: A unified evaluation benchmark for world generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27713–27724, October 2025.

Zicheng Duan, Jiatong Xia, Zeyu Zhang, Wenbo Zhang, Gengze Zhou, Chenhui Gou, Yefei He, Feng Chen, Xinyu Zhang, and Lingqiao Liu. Liveworld: Simulating out-of-sight dynamics in generative video world models. *arXiv preprint arXiv:2603.07145*, 2026.

Paul Engstler, Andrea Vedaldi, Iro Laina, and Christian Rupprecht. Invisible stitch: Generating smooth 3d scenes with depth inpainting. In *Arxiv*, 2024.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Weichen Fan, Chenyang Si, Junhao Song, Zhenyu Yang, Yinan He, Long Zhuo, Ziqi Huang, Ziyue Dong, Jingwen He, Dongwei Pan, et al. Vchitect-2.0: Parallel transformer for scaling up video diffusion models. *arXiv preprint arXiv:2501.08453*, 2025.

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

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.

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.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024.

Roberto Henschel, Levon Khachatryan, Hayk Poghosyan, Daniil Hayrapetyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Streamingt2v: Consistent, dynamic, and extendable long video generation from text. *arXiv preprint arXiv:2403.14773*, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations (ICLR)*, 2022.

Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, et al. Vipe: Video pose engine for 3d geometric perception. *arXiv preprint arXiv:2508.10934*, 2025a.

Tianyu Huang, Wangguandong Zheng, Tengfei Wang, Yuhao Liu, Zhenwei Wang, Junta Wu, Jie Jiang, Hui Li, Rynson WH Lau, Wangmeng Zuo, et al. Voyager: Long-range and world-consistent video diffusion for explorable 3d scene generation. *arXiv preprint arXiv:2506.04225*, 2025b.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17191–17202, 2025.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal feed-forward metric 3D reconstruction, 2025. arXiv preprint arXiv:2509.13414.

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.

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.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *International Conference on Learning Representations (ICLR)*, 2026.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

OpenAI. Sora. `https://openai.com/sora/`, 2024.

Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. UniDepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837*, 2024.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Weijie Wang, Yeqing Chen, Zeyu Zhang, Hengyu Liu, Haoxiao Wang, Zhiyuan Feng, Wenkang Qin, Zheng Zhu, Donny Y. Chen, and Bohan Zhuang. Volsplat: Rethinking feed-forward 3d gaussian splatting with voxel-aligned prediction. *arXiv preprint arXiv:2509.19297*, 2025a.

Weijie Wang, Jiagang Zhu, Zeyu Zhang, Xiaofeng Wang, Zheng Zhu, Guosheng Zhao, Chaojun Ni, Haoxiao Wang, Guan Huang, Xinze Chen, Yukun Zhou, Wenkang Qin, Duochao Shi, Haoyun Li, Yicheng Xiao, Donny Y. Chen, and Jiwen Lu. Drivegen3d: Boosting feed-forward driving scene generation with efficient video diffusion. *arXiv preprint arXiv:2510.15264*, 2025b.

Weijie Wang, Qihang Cao, Sensen Gao, Donny Y Chen, Haofei Xu, Wenjing Bian, Songyou Peng, Tat-Jen Cham, Chuanxia Zheng, Andreas Geiger, et al. Feed-forward 3d scene modeling: A problem-driven perspective. *arXiv preprint arXiv:2604.14025*, 2026a.

Weijie Wang, Donny Y Chen, Zeyu Zhang, Duochao Shi, Akide Liu, and Bohan Zhuang. Zpressor: Bottleneck-aware compression for scalable feed-forward 3dgs. *Advances in Neural Information Processing Systems*, 38:113407–113436, 2026b.

Weijie Wang, Xiaoxuan He, Youping Gu, Yifan Yang, Zeyu Zhang, Yefei He, Yanbo Ding, Xirui Hu, Donny Y Chen, Zhiyuan He, et al. World-r1: Reinforcing 3d constraints for text-to-video generation. *arXiv preprint arXiv:2604.24764*, 2026c.

Weijie Wang, Zimu Li, Jinchuan Shi, Zeyu Zhang, Botao Ye, Marc Pollefeys, Donny Y. Chen, and Bohan Zhuang. Trisplat: Simulation-ready feed-forward 3d scene reconstruction. *arXiv preprint arXiv:2605.26115*, 2026d.

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.

Desai Xie, Zhan Xu, Yicong Hong, Hao Tan, Difan Liu, Feng Liu, Arie Kaufman, and Yang Zhou. Progressive autoregressive video diffusion models. *arXiv preprint arXiv:2410.08151*, 2024.

Jiaqi Xu, Xinyi Zou, Kunzhe Huang, Yunkuo Chen, Bo Liu, MengLi Cheng, Xing Shi, and Jun Huang. Easyanimate: A high-performance long video generation method based on transformer architecture. *arXiv preprint arXiv:2405.18991*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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.

Hong-Xing Yu, Haoyi Duan, Junhwa Hur, Kyle Sargent, Michael Rubinstein, William T Freeman, Forrester Cole, Deqing Sun, Noah Snavely, Jiajun Wu, et al. Wonderjourney: Going from anywhere to everywhere. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6658–6667. IEEE, 2024a.

Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T Freeman, and Jiajun Wu. Wonderworld: Interactive 3d scene generation from a single image. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5916–5926. IEEE, 2025.

Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024b.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

Jinjing Zhao, Fangyun Wei, Zhening Liu, Hongyang Zhang, Chang Xu, and Yan Lu. Spatia: Video generation with updatable spatial memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Jensen Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. *arXiv preprint arXiv:2503.14489*, 2025.

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.

Yuan Zhou, Qiuyue Wang, Yuxuan Cai, and Huan Yang. Allegro: Open the black box of commercial-level video generation model. *arXiv preprint arXiv:2410.15458*, 2024.



(a) RGB Observations (b) Latent Memory Readouts

Figure 7: **Latent memory readout.** (a) The RGB input frame. (b) The target-view memory read out from the corresponding latent spatial memory, VAE-decoded for visualization.

## A Geometric Details

**Validity of Latent-Space Warping** LSM-World builds a target view by moving latent tokens on the grid (Figure 7), instead of warping pixels and re-encoding them with $\mathcal{E}$. A latent token is what the encoder produces from a local patch of the image. Moving such a token to a new cell is valid for two reasons. First, the warp acts only on token positions. The readout in Equation 5 copies each frontmost token as is: it never blends a token's channels, and it never blends neighboring tokens. Second, the encoder is fully convolutional with stride $s$, so shifting the input image by $s$ pixels shifts the latent by one cell. A token is therefore a local descriptor of the surface below it, and it stays a valid input when placed at another cell. We train the conditioning branch and the LoRA adapters on the warped readouts, so the backbone is able to understand them.

**Conventions.** Extrinsics $\mathbf{E}_t \in \mathrm{SE}(3)$ map world points into camera $t$, and we denote the camera-to-world inverse by $\mathbf{E}_t^{-1}$. Intrinsics follow the standard pinhole form with focal lengths $(f_x, f_y)$ and principal point $(c_x, c_y)$. The VAE has spatial stride $s$ dividing the pixel resolution $H \times W$ into the latent resolution $h \times w = (H/s) \times (W/s)$, with $s = 16$ throughout. Latent cells are indexed by $(u, v) \in \{0, \dots, w-1\} \times \{0, \dots, h-1\}$ with pixel-centre homogeneous coordinates $[u + \frac{1}{2}, v + \frac{1}{2}, 1]^\top$.

**Latent-resolution intrinsics.** The latent-resolution intrinsic matrix used by the lifting and readout operations in the main paper is obtained from the pixel-resolution intrinsics $K$ by scaling both axes by the corresponding stride ratio,

$$K^\ell = \mathrm{diag}(w/W,\, h/H,\, 1)\, K. \tag{7}$$

Both focal lengths and the principal point are rescaled by the same per-axis ratio, so that perspective projection remains consistent after the depth map is downsampled to latent resolution.

**Pinhole back-projection.** The lifting operator $\pi^{-1}$ maps a latent cell $(u, v)$ with depth $d = D(u, v)$ into a world point through ray-casting in camera coordinates followed by the camera-to-

world transform,

$$\tilde{\mathbf{u}} = [u + \tfrac{1}{2},\, v + \tfrac{1}{2},\, 1]^\top,$$
$$\pi^{-1}(u, v, d;\, K^\ell, \mathbf{E}) = \mathbf{E}^{-1} \begin{bmatrix} d\,(K^\ell)^{-1}\tilde{\mathbf{u}} \\ 1 \end{bmatrix}\Big|_{1:3}. \tag{8}$$

where the trailing subscript denotes selection of the first three coordinates.

**Projection onto the latent grid.** For a memory point $\mathbf{p}_i$, its camera-space position at target view $t$ is the first three coordinates of $\mathbf{E}_t[\mathbf{p}_i^\top, 1]^\top$, which we denote $\mathbf{q}_i \in \mathbb{R}^3$. Its position on the latent grid is then

$$\pi^\ell(\mathbf{q}_i) = \big(\lfloor x \rfloor, \lfloor y \rfloor\big), \qquad [x, y, 1]^\top = K^\ell\, \mathbf{q}_i / [\mathbf{q}_i]_z. \tag{9}$$

The candidate set used in the main paper's readout operation is then

$$\Omega_t(u, v) = \big\{ i : \pi^\ell(\mathbf{q}_i) = (u, v),\ [\mathbf{q}_i]_z > 0 \big\}, \tag{10}$$

and the admissible cell set $\Lambda_t$ used in the memory update is defined on the same grid, restricted to cells with finite positive depth that lie outside the dynamic-object and sky masks.

## B Additional Experimental Analysis

In this section, we provide analysis that clarifies *why* latent spatial memory is efficient, *how* it behaves as the rollout horizon grows, and *what* the stored tokens carry compared with an RGB cache.

**Cost of cache reads.** Let $N$ denote the cache size at a given step. Reading an RGB cache costs $\Theta(N \log N + HW) + \Theta(\Phi_{\mathcal{E}}(H, W))$ per conditioning step, where $\Phi_{\mathcal{E}}$ is the FLOP count of one VAE encoder pass. Reading a latent cache costs $\Theta(N \log N + hw)$: the rasterization term shrinks by the squared VAE stride, and the per-step encoder term disappears entirely because the readout is already in the backbone's input space. The peak memory of the cache itself follows the ratio $s^2 \cdot (3/C)$ between the RGB and latent representations, which together with the per-step pixel-resolution buffer accounts for the order-of-magnitude gap reported in the main paper. The gap widens with the rollout horizon: the $N \log N$ sort begins to dominate the latent cache much later than the pixel cache, and the RGB baseline exhausts GPU memory on trajectories that LSM-World completes comfortably.

**Per-step timing breakdown.** We decompose the per-step cost of the RGB pipeline into rasterization, VAE encoder forward, and depth estimation plus back-projection. On a $257$-frame rollout, rasterization and the encoder together account for more than $98\%$ of the per-step RGB cost, and both are absent from the conditioning loop of LSM-World, since a single latent-resolution projection replaces them. The decoder, which the RGB pipeline calls at every conditioning step to produce the image on which the cache is rasterized, is invoked in LSM-World only once per chunk to materialize output frames and to feed the depth and segmentation modules; it never appears on the per-step critical path.

**Depth downsampling for cache construction.** The lifting operation requires the dense depth map and the latent grid to share a resolution, so depth must be downsampled before lifting. Because depth is piecewise smooth with discontinuities at silhouettes, this choice is not neutral: bilinear interpolation smears foreground over background at edges and can spawn phantom points; nearest-neighbor preserves edges but aliases fine structure; area pooling over-smooths occluding contours; median pooling preserves edges but is biased on slanted surfaces. We evaluate all four with every other component fixed. Table 4 reports the fraction of empty latent cells in the projected cache ("hole rate"). Among the four downsampling choices, bilinear interpolation gives the lowest hole rate at $42.53\%$. We therefore adopt bilinear as the default, since it provides the best projected-cache coverage among the tested variants.

**What the cache stores.** The latent tokens are expected to carry semantic and textural structure that three color channels cannot express. We visualize this by projecting each per-point feature vector onto its top three principal components and coloring the memory by the resulting RGB. Coherent semantic clusters such as walls, floors, windows, and furniture emerge that are not recoverable from an RGB point cloud built on the same frames. This is the qualitative mechanism behind the main




Figure 8: **Video comparison on RealEstate10K.** Each block shows one RealEstate10K trajectory, with rows corresponding to Voyager, Spatia, VMem, and LSM-World (Ours), and columns showing uniformly sampled frames over time. Across indoor and outdoor scenes, LSM-World preserves sharper structure and more stable appearance under camera motion, while baselines exhibit geometry drift and accumulated artifacts.

paper's ablation results, where swapping the latent cache for an RGB cache with the backbone and training recipe held fixed weakens both 3D and photometric consistency.

## C Implementation Details

**Backbone.** The backbone is Wan2.2-TI2V-5B Wan et al. (2025), whose VAE has spatial stride $s = 16$, temporal stride $4$, and channel count $C = 48$. Each generation chunk covers a $9 \times 44 \times 80$ latent tensor corresponding to 33 RGB frames at $704 \times 1280$. The transformer has hidden dimension $3072$, feed-forward dimension $14336$, $24$ attention heads, $30$ blocks, text context $512$ tokens, and uses RMS Q/K normalization together with cross-attention normalization. The VAE is frozen throughout.

Table 4: **Depth downsampling for cache construction.** Evaluated on a held-out RealEstate10K split with every other component of LSM-World kept fixed.

| Downsampling $D \to D^{\ell}$ | Hole Rate $\downarrow$ |
|---|---|
| Bilinear (default) | 42.53 |
| Nearest-neighbor | 47.78 |
| Area pooling | 53.72 |
| Median pooling | 52.22 |

**Algorithm 1** One rollout of LSM-World.

**Require:** initial frame $I_0$; camera trajectory $\{(\mathbf{E}_t, K_t)\}_{t=0}^{T}$ with $\mathbf{E}_0$ fixed to the world frame

$\mathbf{z}_0 \leftarrow \mathcal{E}(I_0)$
$D_0 \leftarrow \textsc{DepthAnything3}(I_0)$
$M_0 \leftarrow \text{SAM3}(\textsc{Qwen3-VL}(I_0)) \cup \textsc{sky}(I_0)$
$\mathcal{M} \leftarrow \text{Lift}(\mathbf{z}_0, D_0, K_0, \mathbf{E}_0, \Lambda_0)$
$\tau \leftarrow 0$; $\mathcal{O} \leftarrow \{I_0\}$ {collected output frames}
**while** $\tau < T$ **do**
  sample latent chunk $W = \{\tau + 1, \ldots, \tau + |W|\}$
  **for** $t \in W$ **do**
    $\hat{\mathbf{z}}_t, \mathbf{m}_t \leftarrow$ readout of $\mathcal{M}$ at $(\mathbf{E}_t, K_t)$
  **end for**
  $\mathcal{H}_W \leftarrow \{(\hat{\mathbf{z}}_t, \mathbf{m}_t) : t \in W\}$
  $\{\mathbf{z}_t\}_{t \in W} \leftarrow \text{Backbone}(\mathcal{H}_W, \text{preceding}, \text{reference})$
  **for** $t \in W$ **do**
    $I_t \leftarrow \mathcal{D}(\mathbf{z}_t)$; append $I_t$ to $\mathcal{O}$
    $D_t \leftarrow \textsc{DepthAnything3}(I_t)$
    $M_t \leftarrow \text{SAM3}(\textsc{Qwen3-VL}(I_t)) \cup \textsc{sky}(I_t)$
    $\tilde{\mathbf{z}}_t \leftarrow \mathcal{E}(I_t)$
    $\Delta\mathcal{M}_t \leftarrow \text{Lift}(\tilde{\mathbf{z}}_t, D_t, K_t, \mathbf{E}_t, \Lambda_t)$
    $\mathcal{M} \leftarrow \mathcal{M} \cup \Delta\mathcal{M}_t$
  **end for**
  $\tau \leftarrow \tau + |W|$
**end while**

**return** $\mathcal{O}$

**Conditioning branch.** The latent readout $\hat{\mathbf{z}}_t$ is injected through a ControlNet-style side branch whose layout mirrors the VACE Jiang et al. (2025) blocks of Wan2.2. Segment-aware rotary positional encoding tags each frame as noisy target, clean preceding, or clean reference inside a single forward pass; at inference, the denoised latents of the previous chunk become the preceding frames of the next.

**Training.** Training has two flow-matching Lipman et al. (2022) stages on the target frames. Stage one updates only the side branch at learning rate $10^{-5}$ with the backbone and VAE frozen. Stage two unlocks rank-32 LoRA adapters Hu et al. (2022) on the $\{\texttt{q}, \texttt{k}, \texttt{v}, \texttt{o}\ \texttt{ffn}$ projections of every self-attention layer and optimizes them at learning rate $10^{-4}$. Optimization uses AdamW ($\beta = (0.0, 0.999)$, weight decay $10^{-3}$), bfloat16 mixed precision under FSDP sharding, gradient checkpointing, and text-dropout probability $0.2$. At inference we use the UniPC flow scheduler with $40$ steps and classifier-free guidance disabled; efficiency is reported on a single NVIDIA H100.

**Data.** Training clips are drawn from RealEstate10K. Each clip is processed by ViPE Huang et al. (2025a) for per-frame depth, intrinsics, and extrinsics, and by a Qwen3-VL-2B Yang et al. (2025) entity extractor followed by SAM3 Carion et al. (2025) to obtain dynamic-object and sky masks. Cells inside the mask are excluded from $\Lambda_t$ during memory update so that only geometry compatible with the rigid-scene assumption enters the persistent memory. Frames, latents, depth, and camera parameters are stored in an LMDB-backed dataset, so no re-encoding occurs during training.

Rollouts are produced in chunks of nine latent frames with one-frame overlap to preserve temporal continuity, and the cache is grown once per chunk by re-encoding the decoded frames and lifting them into the memory. Algorithm 1 summarizes one complete rollout.

## D HUMAN EVALUATION

To complement automatic metrics, we design a blinded human study that measures the properties most directly affected by persistent spatial memory. Each trial presents the input image, the prescribed camera trajectory, and a generated clip, including a short segment that returns to an earlier viewpoint. Raters score each criterion independently on a scale from 0 (unusable) to 5 (excellent).

The protocol follows questions in Table 5, which separates camera following from other dimensions, and makes the revisit consistency segment explicit so that long-horizon memory is evaluated rather than only local frame quality.

Table 5: **User-study criteria.** All criteria use a 0–5 scale, where higher is better.

| Criterion | Question |
|---|---|
| Scene Revisit Consistency | After leaving and returning to an earlier region, do the same surfaces and objects reappear with the same layout, identity, and spatial relationships? |
| Camera Following | Does the generated motion follow the prescribed camera path without unintended camera drift? |
| Appearance Fidelity | Does the same surface retain its color, texture, illumination, and fine detail across viewpoints and when revisited? |
| Temporal Smoothness | Are transitions between adjacent frames smooth and free of flicker, ghosting, or abrupt changes? |
| Overall Quality | How convincing and usable is the clip in general, considering realism, detail, consistency, and visible artifacts? |

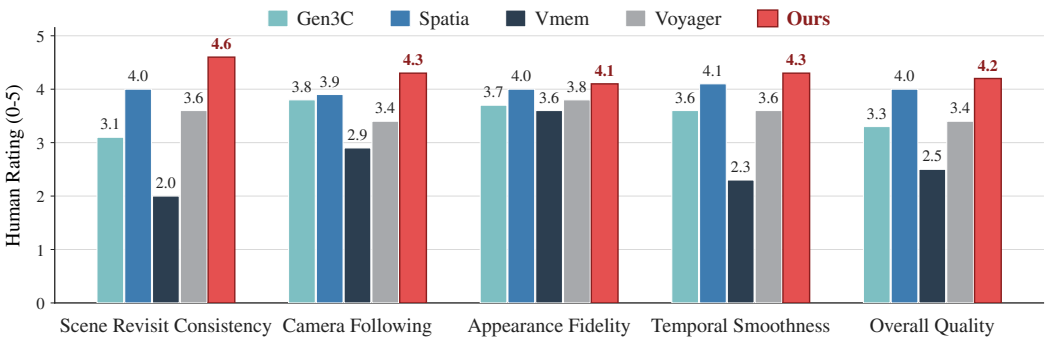


Figure 9: **Human-study ratings.** Mean ratings for the five criteria on a 0–5 scale; higher is better.

We sampled scenes randomly from RealEstate10K and WorldScore, and evaluated all methods under identical inputs and trajectories. Clips were anonymized and shown in a randomized order. We collected ratings from 25 participants over 30 scenes, with each clip receiving at least five independent ratings. As shown in Figure 9, LSM-World achieves the highest mean score on all five dimensions, including 4.6 for scene revisit consistency, 4.3 for camera following, 4.1 for appearance fidelity, 4.3 for temporal smoothness, and 4.2 for overall quality.