LiveWorld: Simulating Out-of-Sight Dynamics in Generative Video World Models

Zicheng Duan[1*], Jiatong Xia[1*], Zeyu Zhang[2*], Wenbo Zhang[1], Gengze Zhou[1], Chenhui Gou[3], Yefei He[4], Feng Chen[1†], Xinyu Zhang[5‡], Lingqiao Liu[1†‡]

[1]Adelaide University, [2]The Australian National University, [3]Monash University, [4]Zhejiang University, [5]University of Auckland

zicheng.duan@adelaide.edu.au, lingqiao.liu@adelaide.edu.au

* Equal contribution. † Project lead. ‡ Corresponding author.

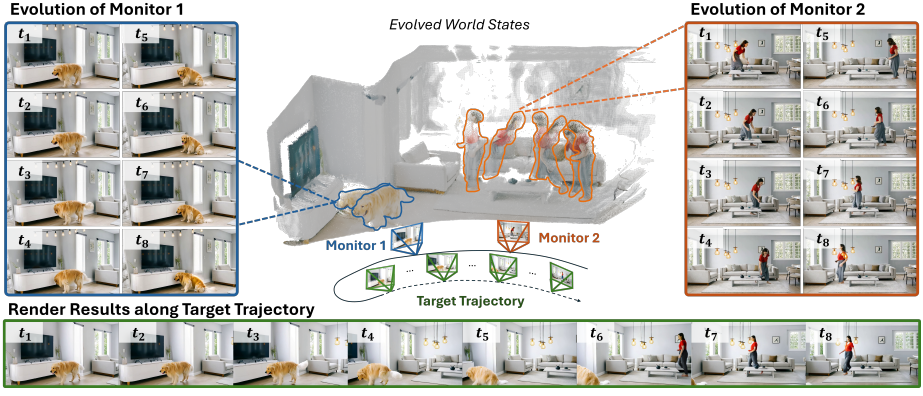

Fig. 1: LiveWorld enables persistent out-of-sight dynamics. Instead of freezing unobserved regions, our framework explicitly decouples world evolution from observation rendering. We register stationary *Monitors* to autonomously fast-forward the temporal progression of active entities (e.g., the dog and the person) in the background. As the observer explores the scene along the target trajectory (green cameras), our state-aware renderer projects the continuously evolved world states to synthesize the final observation. This ensures that dynamic events progress naturally, accurately reflecting the elapsed time even when entities are completely out of the observer's view.

Abstract. Recent generative video world models aim to simulate visual environment evolution, allowing an observer to interactively explore the scene via camera control. However, they implicitly assume that the world only evolves within the observer's field of view. Once an object leaves the observer's view, its state is "frozen" in memory, and revisiting the same region later often fails to reflect events that should have occurred in the meantime. In this work, we identify and formalize this overlooked limitation as the "out-of-sight dynamics" problem, which impedes video world models from representing a continuously evolving world. To address this issue, we propose LiveWorld, a novel framework that extends video world models to support persistent world evolution. Instead of treating the world as static observational memory, LiveWorld models a persistent global state composed of a static 3D background and dynamic entities that continue evolving even when unobserved. To maintain these

unseen dynamics, LiveWorld introduces a monitor-based mechanism that autonomously simulates the temporal progression of active entities and synchronizes their evolved states upon revisiting, ensuring spatially coherent rendering. For evaluation, we further introduce LiveBench, a dedicated benchmark for the task of maintaining out-of-sight dynamics. Extensive experiments show that LiveWorld enables persistent event evolution and long-term scene consistency, bridging the gap between existing 2D observation-based memory and true 4D dynamic world simulation. The baseline and benchmark will be publicly available.

1 Introduction

Recently, there is an increasing demand to build world models that can anticipate future world states based on the current context and control inputs. By modeling the underlying dynamics of the environment, such systems can simulate the progression of a virtual world, offering a powerful platform for applications including agent training [3,4,43], decision-making [13,24], and large-scale synthetic environment generation [7, 11].

Among them, generative video models have emerged as a dominant paradigm for world modeling, leveraging their powerful prior to simulate realistic visual dynamics and enabling users to explore virtual environments through camera control. To maintain temporal consistency during exploration, existing approaches typically condition the generation on historical contexts stored as 2D snapshots in a KV cache [20, 44] or explicitly reconstructed 3D spatial memory [34, 52].

Despite these advancements, current video world models suffer from a fundamental limitation: they conflate the autonomous evolution of the world with camera-dependent rendering. By implicitly collapsing these two distinct processes into a single black-box generator, they are inherently trapped in a *static-world assumption*. Consequently, once an active entity leaves the observer's field of view, its temporal progression is entirely ignored, effectively freezing the entity at its last observed timestamp. We refer to this overlooked phenomenon as the missing of **out-of-sight dynamics**. For instance, if an observer looks away from a dog eating its food, revisiting the same location later will simply retrieve the outdated snapshot of the dog mid-bite, rather than reflecting the elapsed dynamics of it having finished the meal.

To overcome this limitation, we propose **LiveWorld**, a novel video world model framework that explicitly decouples world evolution ($\mathcal{E}$) from observation rendering ($\mathcal{R}$). Recognizing that maintaining a fully dense 4D state of the entire unobserved world is computationally intractable, we introduce a *structured world-state approximation*. We factorize the global world state into two components based on their physical nature: a temporally invariant static background ($\mathcal{M}_{static}$), which is accumulated into a 3D spatial point cloud; and sparsely distributed dynamic entities ($\mathcal{M}_{dyn}$), which explicitly retain their temporal dimensions to continue evolving out of sight.

To seamlessly maintain and evolve this decoupled 4D world, we design a *monitor-driven pipeline*. When a dynamic entity is detected, the system retroac-

tively registers a virtual "Monitor" at its location. Even when the entity is no longer in the observer's field of view, the monitor autonomously fast-forwards its temporal progression, yielding an evolving 4D dynamic point cloud. To render the observer's continuous view, we project both the static environment and the up-to-date dynamic entities onto the target camera trajectory to serve as precise geometric conditioning. Importantly, recognizing that both unobserved temporal evolution and world rendering share the same generative paradigm, we implement both the monitor and the renderer using a *unified state-conditioned video diffusion backbone* that takes the projected states and auxiliary references as input to produce the rendering results and the evolution video. In summary, our main contributions are as follows:

- We rigorously identify and formalize the missing *out-of-sight dynamics* problem in current video world models, highlighting the critical flaw of conflating world evolution with observation rendering.
- We propose **LiveWorld**, a decoupled video world model framework featuring a monitor-centric evolution system and a unified video backbone, which enables the autonomous temporal progression of unobserved entities.
- We develop the first dedicated benchmark, **LiveBench**, specifically designed to quantitatively evaluate long-horizon out-of-sight dynamics and event permanence for video world models.
- Extensive experiments on LiveBench demonstrate that LiveWorld successfully bridges the gap between 2D static memorization and 4D dynamic simulation, significantly outperforming existing baselines.

2 Related Works

2.1 Video World Models

World modeling aims to construct an evolving environment to simulate the real world. Existing works approach this problem from different perspectives. [41] builds explicit geometric-consistent 3D representations, while JEPA [1,5] learns abstract state transitions in latent space. With the rapid progress of video generation models [6,14,27,35], Generative Video World Models [7,11,24] have become a dominant paradigm as they achieve more scalable and realistic world modeling. Typically, a video world model predicts future frames conditioned on historical context and control signals. First, to leverage historical content, early methods [6,14,27,33,35] concatenate anchor frames at the sequence start. CausVID [47] pioneeringly distills fixed-size self-attention into causal attention, enabling autoregressive next-frame prediction with KV cache [7,10,11,20,21,32,44,46,48,51] to store arbitrary history tokens. Next, to support controllable exploration, recent works [15–18,29,43,45] incorporate camera trajectory embeddings. Combined with cached history, these methods allow revisiting previously observed regions. However, the cached frame tokens merely capture 2D visual snapshots of past regions at the exact time they were observed. This inherently relies on a static-world assumption, completely failing to maintain out-of-sight dynamics.

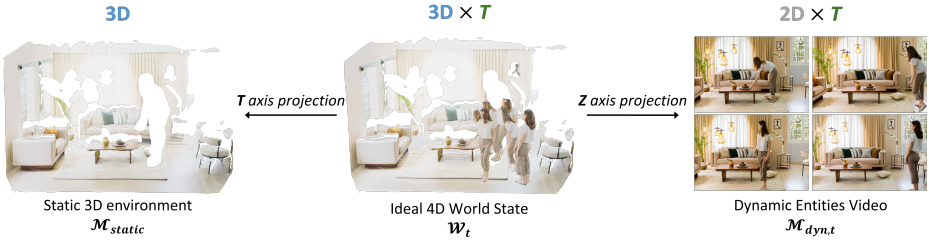

Fig. 2: World State Formulation. We approximate the intractable 4D world state $\mathcal{W}_t$ by decoupling it into two trackable representations: a temporally-invariant static 3D environment $\mathcal{M}_{static}$ via T-axis projection, and 2D video sequences of dynamic entities $\mathcal{M}_{dyn,t}$ via Z-axis projection.

2.2 Explicit Spatial Memory

To achieve precise camera control and maintain long-term geometry during exploration, recent methods [22, 23, 30, 34, 39, 40, 52, 53] have been developed conditioned on explicit 3D spatial memory. These approaches maintain an explicitly reconstructed spatial representation as a form of global memory—such as a point cloud combined with camera parameters predicted by feed-forward estimators [26, 28, 31, 36–38]. By injecting this structural representation into the video generation model, they ensure rigorous geometric consistency. Despite these advancements in spatial tracking, the temporal dimension of historical locations remains neglected. The registered 3D representations store merely the static 3D spatial structure of the scene at the moment of observation. Consequently, they still cannot capture the *out-of-sight dynamics* within previously visited areas.

2.3 Out-of-sight Dynamics

An ideal world model should capture the continuous evolution of the entire explored space, forming a unified 4D spatio-temporal field where scene states evolve consistently over time. This poses a fundamental challenge to existing video world models. Current approaches [7,10,11,20–23,30,32,34,39,40,44,46,48,51–53] only update states within the camera's visible region, while content outside the current view is merely stored as historical observations in memory (e.g., KV cache or spatial memory), remaining frozen at their last observed timestamps. In this work, we formalize this overlooked problem within the video world model paradigm. Building upon explicit spatial memory, we propose a decoupled framework to continuously maintain and evolve this out-of-view representation, thereby bridging the gap between static scene memorization and true 4D dynamic world simulation.

3 Methods

We propose **LiveWorld**, a video world model framework designed to maintain and continuously evolve the out-of-sight dynamics of the currently unobserved world. Unlike previous video world models that follow an observer-centric paradigm—conflating world evolution with rendering and freezing out-of-view

regions into static snapshots—our system explicitly decouples temporal evolution from spatial rendering. By implementing a monitor-centric pipeline, we autonomously model the temporal progression of active entities, narrowing the gap between the 2D video world modeling and the 4D dynamic world.

3.1 Problem Formulation

Decoupling World Evolution from Observation Rendering. The world continues to evolve even when it is not observed. That is to say, an ideal world model should maintain a latent global world state $\mathcal{W}_t$ at each time step t, which specifies the underlying 4D scene at that moment. Since this state is view-independent, while an observed frame is only a camera-dependent projection, world modeling naturally decomposes into two processes: **(1) state evolution**, which updates the world over time, and **(2) rendering**, which maps the current state to an observation under a condition $C_t = \{C_t^{\text{cam}}, C_t^{\text{text}}\}$ (where C_t^{cam} is the camera pose and C_t^{text} is the semantic prompt). Formally,

$$\mathcal{W}_t = \mathcal{E}(\mathcal{W}_{<t}), \quad F_t = \mathcal{R}(\mathcal{W}_t, C_t). \tag{1}$$

where $\mathcal{E}$ denotes the evolution engine and $\mathcal{R}$ denotes the conditioned renderer.

However, existing video world models do not maintain such an explicit world state. Instead, they compress the evolving 4D world into a history of 2D observations and directly predict the next frame from previously observed frames $\mathrm{F}_{<t} = \{F_{t-1}, F_{t-2}, \ldots, F_0\}$ and control signals C_t:

$$F_t = \mathcal{V}_\theta(\mathrm{F}_{<t}, C_t). \tag{2}$$

Under this formulation, world evolution and rendering are implicitly collapsed into a single black-box generator $\mathcal{V}_\theta$. The key limitation is that $\mathrm{F}_{<t}$ contains only camera-dependent visual snapshots rather than the full underlying state $\mathcal{W}_{<t}$. In other words, the continuous 4D world is flattened into a sequence of 2D observations, and once a region leaves the field of view, the model has no explicit state to update, so its temporal progression is ignored and the region remain frozen at its last observed timestamp. We refer to this missing temporal progression as **out-of-sight dynamics**.

Structured World-State Approximation To address this, **LiveWorld** restores an explicit separation between *world evolution* and *camera-conditioned rendering*. Directly maintaining a fully explicit 4D world state is impractical; therefore, we structurally approximate $\mathcal{W}_t$ based on a simple intuition: the static scene is temporally invariant, while temporal changes are concentrated in sparse dynamic entities. Accordingly, we decompose the world state into two components as illustrated in Fig. 2. **(1) Static background.** We collapse the time-invariant background of the world along the temporal axis into a static 3D scene representation $\mathcal{M}_{static}$. For **(2) dynamic entities**, their states continuously evolve over time. To maintain their up-to-date representations $\mathcal{M}_{dyn,t}$ (especially when they are out of the observer's sight), we introduce an explicit

evolution function G_θ^{evo} serving as $\mathcal{E}$ in Eq. 1. It takes historical frames $\mathrm{F}_{<t}$ containing active entities and simulates their continuous temporal progression, yielding the evolved dynamic representation at time t:

$$\mathcal{M}_{dyn,t} = G_\theta^{\text{evo}}(\mathrm{F}_{<t}) \tag{3}$$

The up-to-date world state is then approximated as:

$$\mathcal{W}_t \approx \{\mathcal{M}_{static}, \mathcal{M}_{dyn,t}\} \tag{4}$$

Under this formulation, the video generator no longer implicitly handles hidden temporal evolution. Instead, it serves strictly as a state-aware renderer G_θ^{render} (serving as $\mathcal{R}$ in Eq. 1) that projects the composed world state under the control signal C_t:

$$F_t = G_\theta^{\text{render}}(\mathcal{W}_t, C_t). \tag{5}$$

Implementing this decoupled formulation requires two components: (1) an explicit evolution mechanism G_θ^{evo} to update unobserved active regions over time (Sec. 3.3), and (2) a state-aware generative process G_θ^{render} that renders the composed 3D/2D world state into a coherent observation (Sec. 3.4).

3.2 Unified State-Conditioned Video Backbone

While the state-aware renderer G_θ^{render} and the evolution engine G_θ^{evo} are conceptually distinct, they share a fundamental generative paradigm: both synthesize future visual content conditioned on previous world states and external control signals. Motivated by this structural commonality, we propose a unified state-conditioned video diffusion model G_θ as an abstract interface model. Moreover, although we formulate state evolution at an atomic time step t in Sec. 3.1; In practice, we follow foundational video diffusion models [6, 27, 35] that generate video chunks in discrete autoregressive rounds. Therefore, we instantiate the generative step to cover a temporal window of T frames. In the following sections, we use the notation $t : t + T$ to denote the generation of a T-frame sequence starting from timestamp t. Specifically, G_θ comprises a latent Video Diffusion Transformer (DiT) [35] backbone, augmented with a dual-injection conditioning design to process explicit state projections and detailed appearances respectively:

Explicit state conditioning via state adapter. To inject the maintained world states into the generation process, we employ a *state adapter* initialized from [25]. This module functions as a ControlNet [49], taking a pixel-level projection tensor $\mathbf{P}_{t:t+T} \in \mathbb{R}^{T\times H\times W\times C}$ that represents the explicitly projected state for the target generation window. By injecting these signals into the DiT backbone, the state adapter imposes strict, explicit pixel-level guidance for the generated frames, ensuring consistency with the underlying world state.

Incorporating complementary appearance references. Because the projected state $\mathbf{P}_{t:t+T}$ primarily serves as structural and positional guidance and may lack fine-grained visual details, inspired by [9, 19, 52], we register learnable LoRA parameters to the DiT backbone to accept concatenated historical reference frames.

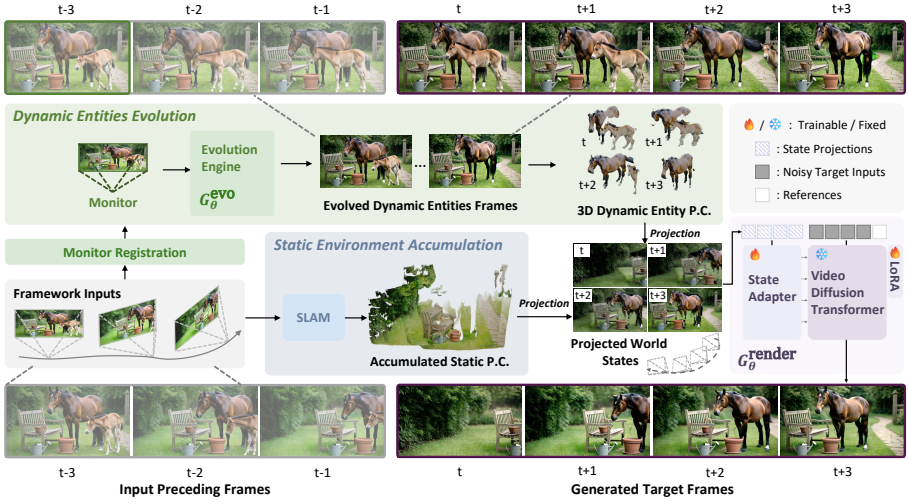

Fig. 3: LiveWorld overview. Our system explicitly decouples world modeling into two processes. **(1) Static Accumulation (Blue):** Temporally-invariant backgrounds are fused into a static 3D point cloud via SLAM. **(2) Dynamic Evolution (Green):** Stationary monitors use the Evolution Engine G_θ^{evo} to fast-forward the out-of-sight progression of active entities, lifting them into 4D point clouds. **(3) State-aware Rendering (Purple):** Both representations are projected onto the target camera trajectory. This geometric projection, alongside appearance references, guides the renderer G_θ^{render} to synthesize coherent observations reflecting the elapsed dynamics.

These references typically include a temporal anchor (e.g., the immediately preceding frames) to maintain motion continuity, and appearance anchors (retrieved from past frames) to fill in dense visual textures. We concatenate these reference frames along the token axis to the input of the DiT backbone. By flexibly configuring the input projection $\mathbf{P}_{t:t+T}$ and the appearance references, this unified backbone can be seamlessly instantiated for entirely different roles. As detailed below, we utilize this shared architecture to perform autonomous unobserved evolution (Sec. 3.3) and observer-centric spatial rendering (Sec. 3.4).

Unified Backbone Interface. The shared architecture G_θ synthesizes a T-frame video chunk $V_{t:t+T}$, guided by three abstract conditioning inputs: an explicit state projection $\mathbf{P}$ (which encapsulates the geometric camera control C^{cam}), supplementary appearance references $\mathbf{A}$, and the text prompt C^{text}.

$$V_{t:t+T} = G_\theta(\mathbf{P}_{t:t+T},\, \mathbf{A},\, C^{\text{text}}_{t:t+T}) \tag{6}$$

3.3 Evolving World States

In this section, we introduce the maintenance of the world state formulated in Eq. 4 through a continuous, multi-round update process. In each generation

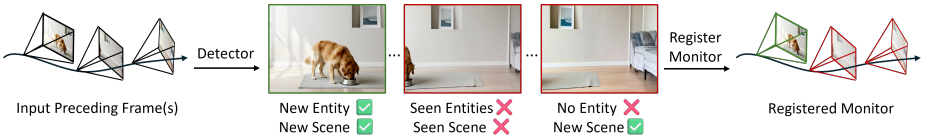

Fig. 4: Given one or multiple preceding frames from the previous round, we first detect if the scene visited by the observer contains active dynamic entities, using off-the-shelf VLMs and segmentors. Following a positive detection, we further validate if the entity and scene are already registered by existing monitors.

round spanning the temporal window t to $t+T$, the system iteratively prepares the up-to-date world state $\mathcal{W}_{t:t+T}$ by executing two fundamental updates: **(1)** accumulating newly observed static regions into the temporally-invariant 3D representation $\mathcal{M}_{static}$, and **(2)** autonomously updating the registered active entities via the evolution function G_θ^{evo} to simulate their out-of-sight dynamics over this window into $\mathcal{M}_{dyn,t:t+T}$.

Accumulating the static environment $\mathcal{M}_{static}$ We represent the static environment $\mathcal{M}_{static}$ as an accumulated background point cloud. For each historically observed frame in $\mathrm{F}_{<t}$, the background is segmented [8] and incrementally fused into a global static point cloud via a feed-forward SLAM framework Stream3R [28], enabling continuous online memory accumulation.

Evolving dynamic entities $\mathcal{M}_{dyn}$ We implement the evolution function $\mathcal{E}$ through a *monitor-driven dynamic evolution* system. Instead of modeling the entire unobserved world, we dynamically allocate stationary virtual agents, termed *Monitors*, to track localized active regions.

Defining and registering monitors. Monitors are a set of generative agents placed at different world positions along the past camera trajectory, aiming to continuously evolve the previously observed scenes containing dynamic entities. They are powered by shared modules: a VLM-based detector [2,8] for entity detection, and the evolution engine G_θ^{evo}. Specifically, as illustrated in Fig. 4, before generating the new chunk $F_{t:t+T}$, we inspect the previously generated frames $F_{t-T:t}$ using the detector. If an unseen dynamic entity is detected along the observer's trajectory, and its spatial overlap with existing monitored regions falls below a certain threshold, a new monitor is registered at that specific pose (denoted as its *anchor pose*). The monitor then takes the observed frame as its *anchor frame*. To maintain computational efficiency, we limit the number of active monitors to M, discarding the one farthest from the observer when this limit is exceeded.

Monitor-driven dynamic evolution. After registration, we take the instantiated evolution engine G_θ^{evo} to simulate ongoing out-of-sight dynamics in the upcoming round for each monitor. First, to fix the monitor at the desired position without camera movements, recalling the state injection formulation (Eq. 6), We instantiate the explicit state condition $\mathbf{P}_{t:t+T}$ of G_θ^{evo} using the static background from the anchor frame, and condition G_θ^{evo} on text prompts $C_{t:t+T}^{text}$ detailing the anticipated actions of the entities to evolve the foreground entities, then cropped entities $\mathbf{A}^{\text{entity}}$ from the anchor frame for the additional appearance references.

These inputs are processed through G_θ^{evo} to generate a local video depicting the continued dynamics of the entity over the interval $t : t+T$.

Asynchronous temporal synchronization. Since a new entity may emerge mid-round (e.g., first observed by the main camera at $t_a \in (t-T, t)$), its initial state is asynchronous with the current global timestamp t. To synchronize it before the next evolution round, the evolution engine G_θ^{evo} first synthesizes the missing local frames from t_a to t. This aligns the entity's state with the global timeline, preparing it for the upcoming $t : t+T$ evolution.

Instantiation I: Evolution Engine (G_θ^{evo}). We instantiate G_θ to synthesize the monitor's local video $v_{t:t+T}^{\text{monitor}}$, which is subsequently lifted to form $\mathcal{M}_{dyn,t:t+T}$. The inputs are specialized for localized event evolution:

$$v_{t:t+T}^{\text{monitor}} = G_\theta^{\text{evo}}(\mathbf{P}_{t:t+T}^{\text{bg}_{anc}}, \mathbf{A}^{\text{entity}}, C_{t:t+T}^{\text{text}}) \tag{7}$$

where $\mathbf{P}_{t:t+T}^{\text{bg}_{anc}}$ is the repeated static anchor frame background, $\mathbf{A}^{\text{entity}}$ is the cropped entity reference, and $C_{t:t+T}^{\text{text}}$ is the action prompt.

Integrating dynamic memory. Finally, equipped with the known monitor anchor pose and per-frame depth, the monitor unprojects the 2D dynamic foreground from $v_{t:t+T}^{\text{monitor}}$ back into the 3D world space. This lifting process yields a localized, temporally evolving 4D Monitor Point Cloud. This explicit 4D representation constitutes the concrete output $\mathcal{M}_{dyn,t:t+T}$ defined in Eq. 3, providing the up-to-date dynamic memory to compose $\mathcal{W}_{t:t+T}$.

3.4 Rendering World States

With the updated world state $\mathcal{W}_{t:t+T}$ prepared (including the static point cloud $\mathcal{M}_{static}$ and the evolved $\mathcal{M}_{dyn,t:t+T}$), the final step is to synthesize the observer's visual experience. Here, we instantiate the unified backbone G_θ (Sec. 3.2) into its second role: the state-aware renderer G_θ^{render}.

Unlike the evolution engine that operates on stationary monitor poses, G_θ^{render} synthesizes the scene from a continuously moving observer trajectory. Specifically, we project both the static 3D environment and the evolved dynamic monitor frames into the observer's novel camera views to construct the global state projection $\mathbf{P}_{t:t+T}$. Guided by this projection and historical reference frames, G_θ^{render} autoregressively renders the final observation video $F_{t:t+T}$, completing the generation loop.

Projecting the evolved world states At each generation round spanning t to $t+T$, the updated static point cloud $\mathcal{M}_{static}$ and the evolved dynamic 4D point clouds $\mathcal{M}_{dyn,t:t+T}$ are used to derive the explicit state projection $\mathbf{P}_{t:t+T}$ for the state adapter of G_θ^{render}:

$$\mathbf{P}_{t:t+T} = \text{Proj}(\{\mathcal{M}_{static}, \mathcal{M}_{dyn,t:t+T}\}, C_{t:t+T}^{cam}) \in \mathbb{R}^{T\times H\times W\times C} \tag{8}$$

As a result, generation strictly conditioned on the state projection $\mathbf{P}_{t:t+T}$ enables explicit camera control while consistently reflecting the spatial evolution of both the static environment and dynamic events throughout the sequence.

Reference frames retrieval for appearance guidance To supplement dense visual details through the registered LoRA of G_θ^{render}, we retrieve the latest preceding frame F_{t-1} as a temporal anchor for motion continuity and if $C_{t:t+T}^{cam}$ revisits previously explored regions, we retrieve the corresponding eariliest historical frames, containing least visual drifting, from $\mathrm{F}_{<t}$ as appearance anchors. This strategy ensures high visual fidelity and texture consistency across continuous explorations.

Instantiation II: Observer Renderer (G_θ^{render}). We instantiate G_θ to render the global observation chunk $F_{t:t+T}$ based on the composed world state and the text control:

$$F_{t:t+T} = G_\theta^{\text{render}}(\mathbf{P}_{t:t+T}^{\text{global}}, \mathbf{A}^{\text{history}}, C_{t:t+T}^{\text{text}}) \tag{9}$$

where $\mathbf{P}_{t:t+T}^{\text{global}} = \text{Proj}(\{\mathcal{M}_{static}, \mathcal{M}_{dyn,t:t+T}\}, C_{t:t+T}^{\text{cam}})$ integrates both static and out-of-sight dynamics, and $\mathbf{A}^{\text{history}}$ provides retrieved spatial textures.

3.5 Model Training

The unified backbone G_θ is trained with the flow matching objective [12]. Given a clean target latent $\mathbf{z}_0$ encoded by a frozen VAE, noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, and timestep $t \sim \mathcal{U}[0, 1]$, the training loss is:

$$\mathcal{L} = \mathbb{E}_{\mathbf{z}_0, \boldsymbol{\epsilon}, t} \left\| \mathbf{v}_\theta(\mathbf{z}_t, t, \mathbf{P}, \mathbf{A}, C) - (\boldsymbol{\epsilon} - \mathbf{z}_0) \right\|^2 \tag{10}$$

where $\mathbf{z}_t = (1-t)\mathbf{z}_0 + t\boldsymbol{\epsilon}$. The loss is computed only on target frames; preceding and reference frames serve as clean conditioning tokens. We adopt a two-stage strategy: **Stage 1** trains the state adapter with the backbone frozen, and **Stage 2** freezes the adapter and fine-tunes LoRA modules on the backbone's attention layers for appearance reference integration. All encoders remain frozen throughout. Training data construction can be found in the Appendix.

4 Experiments

4.1 Introducing LiveBench

Benchmark Construction. We aim to quantitatively evaluate the long-horizon maintenance of dynamic events by pairing diverse scene images with procedurally generated multi-round camera trajectories and text-driven scripts.

Scene image curation. We curate 100 diverse scene images featuring various foreground entities and backgrounds. Using a VLM [2] for prompt composition and a text-to-image model [42], we generate photorealistic 480×832 images, strictly enforcing sharp focus, and clean, depth-unambiguous backgrounds.

Trajectory design. After estimating scene geometry via Stream3R [28], we generate camera trajectories that alternate between leaving and revisiting the initial viewpoint. We define two families: **(i) Same-Pose Revisit** ($A \to B \to A \to B \to A$ over four rounds), where the camera returns to its original pose, and **(ii) Different-Pose Revisit** ($A \to B \to C$), yielding revisits from novel viewpoints. We scale trajectories to fit individual scene sizes, generate 4 variants per scene (two families $\times$ left/right), each spanning 4 rounds (260 frames at 16 FPS).

Event scripts. A VLM generates per-step event scripts conditioned on the scene, ensuring physically plausible motions with explicit spatial displacements. In total, LiveBench comprises 100 scenes and 400 evaluation sequences.

Quantitative Evaluation Metrics. Due to the lack of ground-truth videos for out-of-sight dynamics, we design a reference-based and VLM-driven protocol.

Spatial Memory and Identity. For same-pose revisits, we evaluate static background consistency against the initial frame (excluding dynamic regions) using PSNR, SSIM, and LPIPS [50]. For dynamic entities, we compute the Chamfer Distance between the generated dynamic point clouds and the monitor's predictions in 3D world space. Additionally, across both revisit scenarios, we employ a Masked Bag-of-Features (BoF) strategy using foreground-masked DINOv2 tokens (DINO_{fg}) to robustly evaluate identity preservation under severe pose variations.

Event Progression and Consistency. Since pixel metrics struggle with out-of-sight transitions, we utilize VLM-based binary VideoQA to verify if the generated actions and revisited states align with the text scripts (VQA-Acc). Finally, temporal smoothness of the evolved local dynamics is measured via adjacent-frame CLIP similarity (CLIP_F).

4.2 Experimental Setup

Implementation Details. We build upon Wan2.1-14B-T2V [35] with the state adapter initialized from Wan2.1-VACE-14B and rank-64 LoRA on the backbone attention layers. Stage 1 trains the state adapter for 10k steps; Stage 2 fine-tunes LoRA for 5k steps, both at lr 1×10^{-4} with cosine decay. Global batch size is 16 across 16 NVIDIA H200 GPUs in bf16 FSDP. We set the maximum active monitors M=3 and drop text prompts with probability 0.2.

Other Evaluations. Beyond initial-frame revisits in LiveBench, we conduct human evaluation for complex scenarios involving late-appearing entities and concurrent out-of-sight dynamics. We summon a new foreground entity via text prompts mid-generation. Evaluators then assign binary scores across three criteria: Presence, Identity (Id._{fg}), and Event Consistency. From these, we report two metrics: **Event Succ.** (an individual event satisfies all three criteria) and **Full Succ.** (both concurrent events succeed simultaneously), strictly measuring the capacity for persistent multi-event simulation.

Table 1: Quantitative comparison on LiveBench. Background (bg) metrics measure spatial memory against the first scene frame, foreground (fg) metrics measure entity identity preservation, and VQA-Acc evaluates the success of out-of-sight event progression guided by the text prompt. Columns in blue highlight the **second revisit** performance. † denotes our implementation.

	Same Pose Revisit												Different Pose Revisit			
	$PSNR_{bg}$ ↑		$SSIM_{bg}$ ↑		$LPIPS_{bg}$ ↓		CD_{fg} ↓		$DINO_{fg}$ ↑		VQA-Acc↑		$DINO_{fg}$ ↑		VQA-Acc↑	
	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd
MG-2	16.321	16.131	0.512	0.502	0.565	0.629	6.631	7.429	0.335	0.198	7.737	5.012	0.230	0.122	5.111	4.132
GC-1	17.637	16.012	0.571	0.523	0.421	0.572	2.107	6.236	0.527	0.262	20.125	10.273	0.475	0.191	18.799	8.397
Spatia†	**20.132**	19.020	0.672	0.649	**0.297**	**0.310**	4.031	5.122	0.440	0.416	19.205	14.655	0.392	0.363	18.723	13.212
Ours	20.071	**19.983**	**0.679**	**0.650**	0.301	0.330	**0.068**	**0.135**	**0.760**	**0.721**	**59.063**	**54.620**	**0.691**	**0.632**	**52.829**	**49.478**

4.3 Main Results

Initial frame revisiting on LiveBench. Tab. 1 presents the quantitative results for single-event revisiting. We choose state-of-the-art open-sourced camera-conditioned world models that support long horizon generation as comparison baselines, namely Matrix-Game-2.0 [17], Hunyuan-GameCraft-1.0 [29], and Spatia [52]. We analyze the performance across four crucial dimensions:

Spatial Background Maintenance. Benefiting from explicitly accumulated 3D point clouds, both our method and Spatia effectively maintain the static environment, achieving superior background metrics ($PSNR_{bg}$, $SSIM_{bg}$, $LPIPS_{bg}$). Without explicit spatial memory, Matrix-Game-2.0 (MG-2) and GameCraft-1 (GC-1) struggle during the first revisit. By the second long-horizon revisit (cyan columns), their backgrounds collapse completely with severe artifacts (Fig. 5), causing a precipitous metric drop.

Dynamic Entity Preservation. Crucially, LiveWorld uniquely preserves out-of-sight dynamic objects. Our decoupled Evolution Engine persistently updates world dynamics, while explicit state projection ensures perfect foreground alignment upon re-observation, achieving drastically better geometric (CD_{fg}) and semantic ($DINO_{fg}$) consistency. Baselines fundamentally fail here; despite history caching and prompt conditioning, they cannot guarantee foreground consistency across temporal gaps.

Event Progression. Our decoupled architecture ensures highly successful text-script completion (VQA-Acc). By continuously evolving dynamic objects in the background, our renderer accurately captures their logically progressed states upon revisiting. Conversely, baselines entangle foreground motion and camera control within a single generator, causing camera movements to easily disrupt event generation and fail designated scripts.

Different Pose Revisit. Our decoupled design's advantages amplify under novel revisiting viewpoints. While we maintain consistent entity identities ($DINO_{fg}$) and event alignment, baselines suffer further degradation due to exacerbated artifacts and failed camera control from novel angles.

Fig. 5: A comparison result with the latest state-of-the-art methods on LiveBench. With the camera view repeatedly moving rightwards and backwards, our methods stand out alone to successfully maintain long-horizon (260 frames) out-of-sight dynamics, while others fail. Different colors correspond to different evolving prompts of the event.

Table 2: User study on late-appearing event revisit success rate, higher is better (%).

Method	Presence		Id._{fg}		Consistency		Event Succ.		Full Succ.
	E1	E2	E1	E2	E1	E2	E1	E2	
w/o Event Evo	40	39	32	31	20	15	2	3	0
Ours	**92**	**70**	**80**	**61**	**67**	**53**	**42**	**35**	**26**

Late-appear Event Revisiting. Since baselines struggle to initialize multiple entities via text, we benchmark against our camera-control variant (*w/o Event Evo*). As Table 2 shows, our method robustly maintains parallel events, achieving 92% Presence for the primary event (E1). E2 Presence drops to 70% as text-to-video randomness occasionally fails to trigger the entity, causing cascading metric declines. Nevertheless, our approach vastly outperforms the baseline with 42% and 35% **Event Succ.** for E1 and E2. Crucially, on the strict scene-level **Full Succ.** metric (requiring both events to succeed simultaneously), our model secures 26% while the baseline completely collapses (0%). This confirms that explicit evolution is indispensable for multi-event modeling.

4.4 Ablation Studies

To validate the necessity of each core component, we conduct comprehensive ablation studies in Tab. 3.

Effect of Event Evolution. Removing the evolution engine degrades our system into a pure camera control model. While background scores remain competitive, it completely fails to preserve out-of-sight entities, drastically dropping foreground (CD_{fg}, $DINO_{fg}$) and event completion (VQA-Acc) metrics.

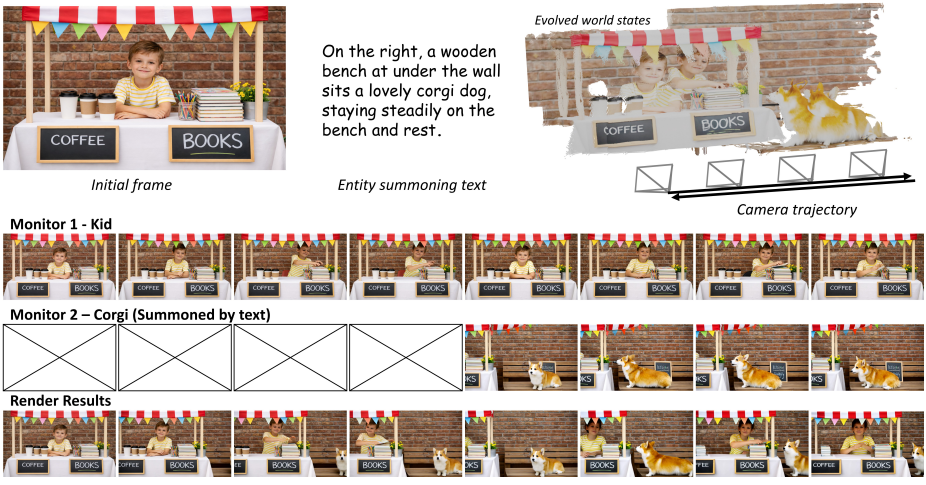

Fig. 6: A demonstration of the late-appearing event revisiting. While we have the initial image containing the kid, we allow the renderer to generate the corgi following the text prompt. The monitor for the corgi is registered after the camera has overlap under the threshold with the monitor of the initial frame. The results showcase a perfect synchronization between the renderer and each monitor.

Table 3: Ablation studies of our proposed components. We validate the necessity of the evolution engine, spatial memory, and reference frames. The column in blue demonstrates the performance of a **second revisit**.

Method	Same Pose Revisit												Different Pose Revisit			
	$PSNR_{bg}$ ↑		$SSIM_{bg}$ ↑		$LPIPS_{bg}$ ↓		CD_{fg} ↓		$DINO_{fg}$ ↑		VQA-Acc↑		$DINO_{fg}$ ↑		VQA-Acc↑	
	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd
w/o Event Evo.	20.005	**18.995**	0.671	0.647	**0.299**	**0.315**	4.215	5.431	0.425	0.401	18.512	13.985	0.380	0.350	18.105	12.855
w/o Spa. Mem.	17.512	16.895	0.550	0.531	0.495	0.550	5.820	6.915	0.395	0.285	12.350	8.510	0.315	0.185	10.125	6.850
w/o Ref. Frame	18.520	17.105	0.610	0.545	0.385	0.490	1.520	4.850	0.615	0.410	38.510	22.150	0.550	0.320	32.105	18.550
Full	**20.071**	18.983	**0.679**	**0.650**	0.301	0.330	**0.068**	**0.135**	**0.760**	**0.721**	**59.063**	**54.620**	**0.691**	**0.632**	**52.829**	**49.478**

Effect of Spatial Memory. Disabling spatial memory causes catastrophic camera control failure. Injecting historical references without proper spatial grounding misaligns the attention LoRA on novel views, resulting in severe ghosting, spatial jittering, and degraded background metrics.

Effect of Reference Frames. Omitting historical references deprives the model of dense visual textures, destabilizing the background. This spatial instability triggers a cascading temporal collapse of the scene, severely degrading all metrics, especially foreground and event alignment, during the 2nd long-horizon revisit.

5 Conclusion

We formalize the *out-of-sight dynamics* problem in video world models, where unobserved regions incorrectly freeze at their last seen state. To overcome this,

we propose LiveWorld, a framework that explicitly decouples continuous world evolution from view-dependent rendering. By factorizing the environment into a static 3D background and utilizing a monitor-centric pipeline to autonomously fast-forward unobserved active entities, LiveWorld achieves tractable 4D modeling. Along with LiveBench, our dedicated benchmark, LiveWorld bridges the gap between static 2D memorization and persistent 4D dynamic simulation.

References

1. Assran, M., Bardes, A., Fan, D., Garrido, Q., Howes, R., Muckley, M., Rizvi, A., Roberts, C., Sinha, K., Zholus, A., et al.: V-jepa 2: Self-supervised video models enable understanding, prediction and planning. arXiv preprint arXiv:2506.09985 (2025) 3
2. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., Zhu, K.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025) 8, 10
3. Bai, Y., Tran, D., Bar, A., LeCun, Y., Darrell, T., Malik, J.: Whole-body conditioned egocentric video prediction. arXiv preprint arXiv:2506.21552 (2025) 2
4. Bar, A., Zhou, G., Tran, D., Darrell, T., LeCun, Y.: Navigation world models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 15791–15801 (2025) 2
5. Bardes, A., Garrido, Q., Ponce, J., Rabbat, M., LeCun, Y., Assran, M., Ballas, N.: Revisiting feature prediction for learning visual representations from video. arXiv:2404.08471 (2024) 3
6. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023) 3, 6
7. Bruce, J., Dennis, M., Edwards, A., Farquhar, J., Grefenstette, E., Harb, J., Kostrikov, I., Lai, R., Lan, C., Laskin, M., et al.: Genie: Generative interactive environments. arXiv preprint arXiv:2402.15391 (2024) 2, 3, 4
8. Carion, N., Gustafson, L., Hu, Y.T., Debnath, S., Hu, R., Suris, D., Ryali, C., Alwala, K.V., Khedr, H., Huang, A., Lei, J., Ma, T., Guo, B., Kalla, A., Marks, M., Greer, J., Wang, M., Sun, P., Rädle, R., Afouras, T., Mavroudi, E., Xu, K., Wu, T.H., Zhou, Y., Momeni, L., Hazra, R., Ding, S., Vaze, S., Porcher, F., Li, F., Li, S., Kamath, A., Cheng, H.K., Dollár, P., Ravi, N., Saenko, K., Zhang, P., Feichtenhofer, C.: Sam 3: Segment anything with concepts (2025), https://arxiv.org/abs/2511.16719 8
9. Chong, Z., Dong, X., Li, H., Zhang, S., Zhang, W., Zhang, X., Zhao, H., Jiang, D., Liang, X.: Catvton: Concatenation is all you need for virtual try-on with diffusion models. arXiv preprint arXiv:2407.15886 (2024) 6
10. Cui, J., Wu, J., Li, M., Yang, T., Li, X., Wang, R., Bai, A., Ban, Y., Hsieh, C.J.: Self-forcing++: Towards minute-scale high-quality video generation. arXiv preprint arXiv:2510.02283 (2025) 3, 4

11. DeepMind, G.: Genie3 official page (2025), https://deepmind.google/models/genie/, accessed: 2026-02-22 2, 3, 4
12. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first international conference on machine learning (2024) 10
13. Gao, R., Chen, K., Xie, E., Hong, L., Li, Z., Yeung, D.Y., Xu, Q.: MagicDrive: Street view generation with diverse 3d geometry control. In: International Conference on Learning Representations (2024) 2
14. Guo, Y., Yang, C., Rao, A., Liang, Z., Wang, Y., Qiao, Y., Agrawala, M., Lin, D., Dai, B.: Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. International Conference on Learning Representations (2024) 3
15. He, H., Xu, Y., Guo, Y., Wetzstein, G., Dai, B., Li, H., Yang, C.: Cameractrl: Enabling camera control for text-to-video generation. arXiv preprint arXiv:2404.02101 (2024) 3
16. He, H., Yang, C., Lin, S., Xu, Y., Wei, M., Gui, L., Zhao, Q., Wetzstein, G., Jiang, L., Li, H.: Cameractrl ii: Dynamic scene exploration via camera-controlled video diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13416–13426 (2025) 3
17. He, X., Peng, C., Liu, Z., Wang, B., Zhang, Y., Cui, Q., Kang, F., Jiang, B., An, M., Ren, Y., Xu, B., Guo, H.X., Gong, K., Wu, C., Li, W., Song, X., Liu, Y., Li, E., Zhou, Y.: Matrix-game 2.0: An open-source, real-time, and streaming interactive world model. arXiv preprint arXiv:2508.13009 (2025) 3, 12
18. Hong, Y., Mei, Y., Ge, C., Xu, Y., Zhou, Y., Bi, S., Hold-Geoffroy, Y., Roberts, M., Fisher, M., Shechtman, E., et al.: Relic: Interactive video world model with long-horizon memory. arXiv preprint arXiv:2512.04040 (2025) 3
19. Huang, L., Wang, W., Wu, Z.F., Shi, Y., Dou, H., Liang, C., Feng, Y., Liu, Y., Zhou, J.: In-context lora for diffusion transformers. arXiv preprint arxiv:2410.23775 (2024) 6
20. Huang, X., Li, Z., He, G., Zhou, M., Shechtman, E.: Self forcing: Bridging the train-test gap in autoregressive video diffusion. arXiv preprint arXiv:2506.08009 (2025) 2, 3, 4
21. Huang, Y., Guo, H., Wu, F., Zhang, S., Huang, S., Gan, Q., Liu, L., Zhao, S., Chen, E., Liu, J., et al.: Live avatar: Streaming real-time audio-driven avatar generation with infinite length. arXiv preprint arXiv:2512.04677 (2025) 3, 4
22. HunyuanWorld, T.: Hunyuanworld 1.0: Generating immersive, explorable, and interactive 3d worlds from words or pixels. arXiv preprint (2025) 4
23. HunyuanWorld, T.: Hy-world 1.5: A systematic framework for interactive world modeling with real-time latency and geometric consistency. arXiv preprint (2025) 4
24. Jiang, C.M., Masotto, X., Sun, B.: The waymo world model: A new frontier for autonomous driving simulation. https://waymo.com/blog/2026/02/the-waymo-world-model-a-new-frontier-for-autonomous-driving-simulation (feb 2026) 2, 3
25. Jiang, Z., Han, Z., Mao, C., Zhang, J., Pan, Y., Liu, Y.: Vace: All-in-one video creation and editing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17191–17202 (2025) 6
26. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., Luiten, J., Lopez-Antequera,

M., Bulò, S.R., Richardt, C., Ramanan, D., Scherer, S., Kontschieder, P.: MapAnything: Universal feed-forward metric 3D reconstruction. In: International Conference on 3D Vision (3DV). IEEE (2026) 4
27. Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al.: Hunyuanvideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603 (2024) 3, 6
28. Lan, Y., Luo, Y., Hong, F., Zhou, S., Chen, H., Lyu, Z., Yang, S., Dai, B., Loy, C.C., Pan, X.: STream3R: Scalable sequential 3D reconstruction with causal transformer. In: ICLR (2026) 4, 8, 11
29. Li, J., Tang, J., Xu, Z., Wu, L., Zhou, Y., Shao, S., Yu, T., Cao, Z., Lu, Q.: Hunyuan-gamecraft: High-dynamic interactive game video generation with hybrid history condition (2025), https://arxiv.org/abs/2506.17201 3, 12
30. Li, R., Torr, P., Vedaldi, A., Jakab, T.: Vmem: Consistent interactive video scene generation with surfel-indexed view memory. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 25690–25699 (2025) 4
31. Lin, H., Chen, S., Liew, J.H., Chen, D.Y., Li, Z., Shi, G., Feng, J., Kang, B.: Depth anything 3: Recovering the visual space from any views. arXiv preprint arXiv:2511.10647 (2025) 4
32. Liu, K., Hu, W., Xu, J., Shan, Y., Lu, S.: Rolling forcing: Autoregressive long video diffusion in real time. arXiv preprint arXiv:2509.25161 (2025) 3, 4
33. Song, K., Chen, B., Simchowitz, M., Du, Y., Tedrake, R., Sitzmann, V.: History-guided video diffusion. arXiv preprint arXiv:2502.06764 (2025) 3
34. Sun, W., Zhang, H., Wang, H., Wu, J., Wang, Z., Wang, Z., Wang, Y., Zhang, J., Wang, T., Guo, C.: Worldplay: Towards long-term geometric consistency for real-time interactive world model. arXiv preprint (2025) 2, 4
35. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., Wang, J., Zhang, J., Zhou, J., Wang, J., Chen, J., Zhu, K., Zhao, K., Yan, K., Huang, L., Feng, M., Zhang, N., Li, P., Wu, P., Chu, R., Feng, R., Zhang, S., Sun, S., Fang, T., Wang, T., Gui, T., Weng, T., Shen, T., Lin, W., Wang, W., Wang, W., Zhou, W., Wang, W., Shen, W., Yu, W., Shi, X., Huang, X., Xu, X., Kou, Y., Lv, Y., Li, Y., Liu, Y., Wang, Y., Zhang, Y., Huang, Y., Li, Y., Wu, Y., Liu, Y., Pan, Y., Zheng, Y., Hong, Y., Shi, Y., Feng, Y., Jiang, Z., Han, Z., Wu, Z.F., Liu, Z.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025) 3, 6, 11
36. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: CVPR (2025) 4
37. Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3d perception model with persistent state. In: CVPR (2025) 4
38. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: CVPR (2024) 4
39. Wang, Z., Cho, J., Li, J., Lin, H., Yoon, J., Zhang, Y., Bansal, M.: Epic: Efficient video camera control learning with precise anchor-video guidance. arXiv preprint arXiv:2505.21876 (2025) 4
40. Wang, Z., Lin, H., Yoon, J., Cho, J., Zhang, Y., Bansal, M.: Anchorweave: World-consistent video generation with retrieved local spatial memories. arXiv preprint arXiv:2602.14941 (2026) 4
41. World Labs Team: Marble: A multimodal world model. https://www.worldlabs.ai/blog/marble-world-model (nov 2025) 3
42. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., ming Yin, S., Bai, S., Xu, X., Chen, Y., Chen, Y., Tang, Z., Zhang, Z., Wang, Z., Yang, A., Yu, B., Cheng, C.,

Liu, D., Li, D., Zhang, H., Meng, H., Wei, H., Ni, J., Chen, K., Cao, K., Peng, L., Qu, L., Wu, M., Wang, P., Yu, S., Wen, T., Feng, W., Xu, X., Wang, Y., Zhang, Y., Zhu, Y., Wu, Y., Cai, Y., Liu, Z.: Qwen-image technical report (2025), https://arxiv.org/abs/2508.02324 10

43. Xiang, J., Gu, Y., Liu, Z., Feng, Z., Gao, Q., Hu, Y., Huang, B., Liu, G., Yang, Y., Zhou, K., et al.: Pan: A world model for general, interactable, and long-horizon world simulation. arXiv preprint arXiv:2511.09057 (2025) 2, 3
44. Yang, S., Huang, W., Chu, R., Xiao, Y., Zhao, Y., Wang, X., Li, M., Xie, E., Chen, Y., Lu, Y., Chen, S.H.Y.: Longlive: Real-time interactive long video generation (2025) 2, 3, 4
45. Ye, D., Zhou, F., Lv, J., Ma, J., Zhang, J., Lv, J., Li, J., Deng, M., Yang, M., Fu, Q., et al.: Yan: Foundational interactive video generation. arXiv preprint arXiv:2508.08601 (2025) 3
46. Yi, J., Jang, W., Cho, P.H., Nam, J., Yoon, H., Kim, S.: Deep forcing: Training-free long video generation with deep sink and participative compression. arXiv preprint arXiv:2512.05081 (2025) 3, 4
47. Yin, T., Zhang, Q., Zhang, R., Freeman, W.T., Durand, F., Shechtman, E., Huang, X.: From slow bidirectional to fast autoregressive video diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 22963–22974 (June 2025) 3
48. Yu, J., Bai, J., Qin, Y., Liu, Q., Wang, X., Wan, P., Zhang, D., Liu, X.: Context as memory: Scene-consistent interactive long video generation with memory retrieval. arXiv preprint arXiv:2506.03141 (2025) 3, 4
49. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models (2023) 6
50. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018) 11
51. Zhang, Z., Chang, S., He, Y., Han, Y., Tang, J., Wang, F., Zhuang, B.: Blockvid: Block diffusion for high-quality and consistent minute-long video generation. arXiv preprint arXiv:2511.22973 (2025) 3, 4
52. Zhao, J., Wei, F., Liu, Z., Zhang, H., Xu, C., Lu, Y.: Spatia: Video generation with updatable spatial memory. arXiv preprint arXiv:2512.15716 (2025) 2, 4, 6, 12
53. Zheng, X., Liu, Y., Wu, C.H., Zhang, F., Zheng, H., Zhou, W., Mayol-Cuevas, W.W., Shen, J.: Spatialmem: Unified 3d memory with metric anchoring and fast retrieval. arXiv preprint arXiv:2601.14895 (2026) 4