# RELIC: Interactive Video World Model with Long-Horizon Memory

**Yicong Hong**[*†], **Yiqun Mei**[*], **Chongjian Ge**[*],
**Yiran Xu**, **Yang Zhou**, **Sai Bi**, **Yannick Hold-Geoffroy**, **Mike Roberts**, **Matthew Fisher**,
**Eli Shechtman**, **Kalyan Sunkavalli**, **Feng Liu**, **Zhengqi Li**, **Hao Tan**

[*]First Authors in Random Order, [†]Project Lead.

A truly interactive world model requires three key ingredients: real-time long-horizon streaming, consistent spatial memory, and precise user control. However, most existing approaches address only one of these aspects in isolation, as achieving all three simultaneously is highly challenging—for example, long-term memory mechanisms often degrade real-time performance. In this work, we present **RELIC**, a unified framework that tackles these three challenges altogether. Given a single image and a text description, **RELIC** enables memory-aware, long-duration exploration of arbitrary scenes in real time. Built upon recent autoregressive video-diffusion distillation techniques, our model represents long-horizon memory using highly compressed historical latent tokens encoded with both relative actions and absolute camera poses within the KV cache. This compact, camera-aware memory structure supports implicit 3D-consistent content retrieval and enforces long-term coherence with minimal computational overhead. In parallel, we fine-tune a bidirectional teacher video model to generate sequences beyond its original 5-second training horizon, and transform it into a causal student generator using a new memory-efficient self-forcing paradigm that enables full-context distillation over long-duration teacher as well as long student self-rollouts. Implemented as a 14B-parameter model and trained on a curated Unreal Engine–rendered dataset, **RELIC** achieves real-time generation at 16 FPS while demonstrating more accurate action following, more stable long-horizon streaming, and more robust spatial-memory retrieval compared with prior work. These capabilities establish **RELIC** as a strong foundation for the next generation of interactive world modeling.

**Date:** December 1st, 2025
**Project Page:** https://relic-worldmodel.github.io/

Adobe

## 1 Introduction

Imagine stepping into a picture and freely exploring or interacting with the world behind it in real time. *World modeling*, which embodies this overarching vision, has gained significant attention for its potential to understand and simulate our three-dimensional physical world (Ha and Schmidhuber, 2018; World Labs, 2025c; Brooks et al., 2024). By enabling interaction between an agent and its surrounding environment, world models can create a wide variety of real-world scenarios, facilitating downstream applications such as autonomous driving (Hu et al., 2023; Tu et al., 2025; Zhou et al., 2025b), spatial intelligence (Parker-Holder et al., 2024; Ball et al., 2025), and robotics (Bardes et al., 2023; Roth et al., 2025; Gu, 2025). Recent advances in video generation thorough diffusion models (Wan et al., 2025; Gao et al., 2025; Lin et al., 2025; Wiedemer et al., 2025; Polyak et al., 2024; Kong et al., 2024) have demonstrated compelling potential to realize such immersive and interactive experiences by learning controllable autoregressive (AR) video models from large-scale synthetic and real-world datasets (Li et al., 2025a; Wu et al., 2025b; Feng et al., 2024; Zhang et al., 2025c; He et al., 2025; Ye et al., 2025; Parker-Holder et al., 2024; Ball et al., 2025; Team et al., 2025; Yu et al., 2025c; Mao et al., 2025; Huang et al., 2025b; Chen et al., 2025b).

There are two essential foundation for building a practical video world model: 1) *real-time long-horizon streaming* and 2) *consistent spatial memory*. Real-time long-horizon streaming requires that video be generated at real-time latency in response to a continuous stream of user controls, such as camera movements or keyboard inputs. Consistent spatial memory, on the other hand, ensures that the model does not forget previously seen or explored scene content. Together, these capabilities enable the model to iteratively construct and maintain a persistent underlying 3D world. In the context of autoregressive (AR) video diffusion models, real-time latency and throughput demand few-step or even one-step

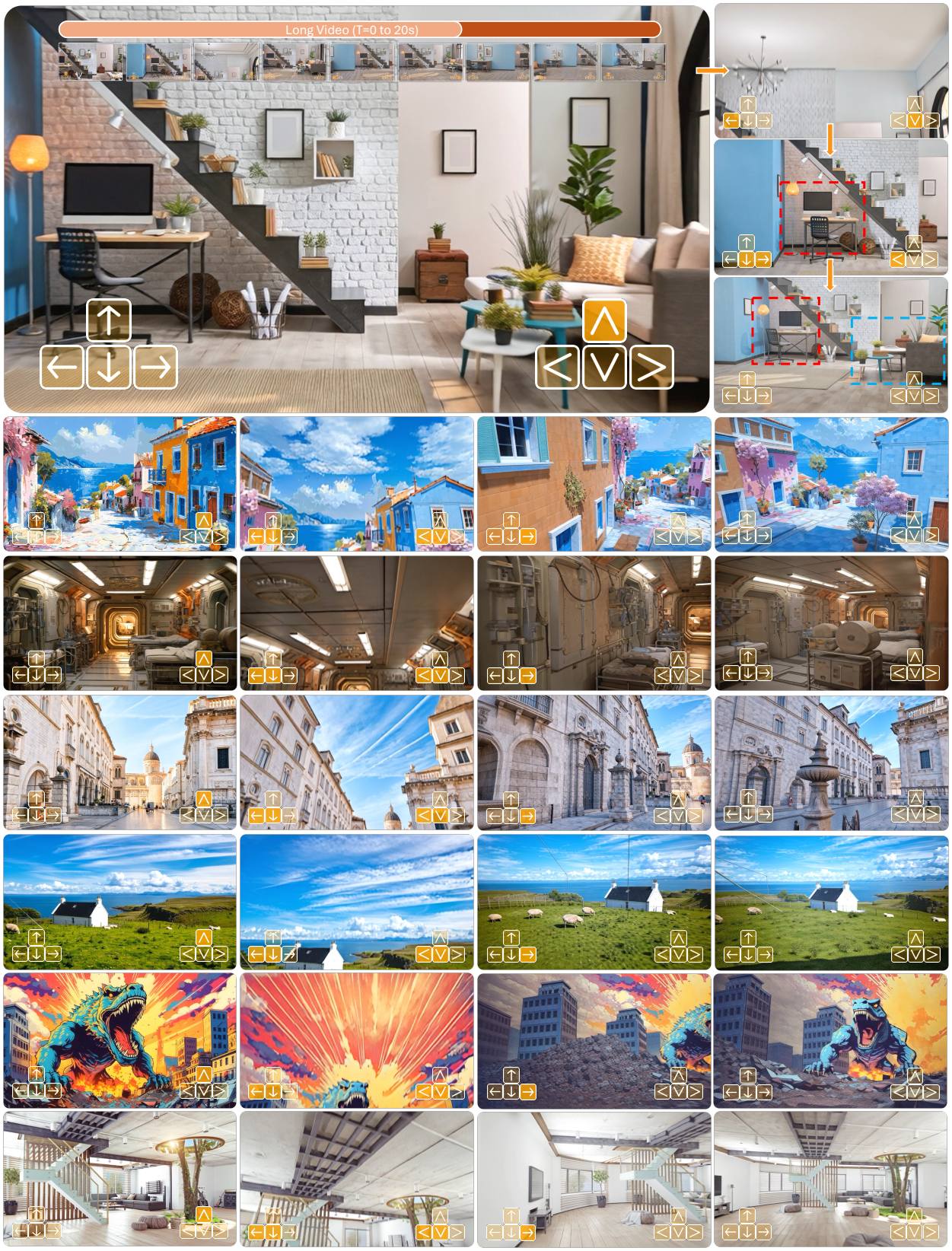



**Figure 1** RELIC is an interactive video world model that allows users to freely explore virtual scenes initialized from an arbitrary first-frame image in real time. Built as a 14B-parameter autoregressive model, RELIC generates videos at 480×832 resolution, 16 FPS, for up to 20 seconds, exhibiting consistent long-horizon spatial memory.

**Table 1 Comparison with recent video world models.** Note that generalization and memory often trade off against output duration and dynamic level. In particular, most of the models that can produce infinite-length stable videos in training-aligned domains do not generalize to arbitrary scenes or actions. Also, the generation speed largely depends on the model size, GPU hardware, and the degree of parallelism. `T`, `R`, and `E` are translation-, rotation-, and other interaction-related actions, respectively.

| | The Matrix | Genie-2 | GameCraft | Yume | Yan | Matrix-Game 2.0 | Genie-3 | RELIC (ours) |
|---|---|---|---|---|---|---|---|---|
| Data Source | AAA Games | Unknown | AAA Games | Sekai | 3D game | Minecraft+UE+Sekai | Unknown | UE |
| Action Space | 4T4R | 5T4R2E | 4T4R | 4T4R | 7T2R | 4T | 5T4R1E | 6T6R |
| Resolution | 720×1280 | 720×1280 | 720×1280 | 544×960 | 1080×1920 | 352×640 | 704×1280 | 480×832 |
| Speed | 8-16 FPS | Unknown | 24 FPS | 16 FPS | 60 FPS | 25 FPS | 24 FPS | 16 FPS |
| Duration | Infinite | 10-20 sec | 1 min | 20 sec | Infinite | 1 min | 1 min | 20 sec |
| Generalization | ☆ | ☆☆ | ☆☆ | ☆☆ | ☆ | ☆☆ | ☆☆☆ | ☆☆☆ |
| Memory | None | ☆ | ☆ | None | None | None | ☆☆☆ | ☆☆☆ |
| Model Size | 2.7B | Unknown | 13B | 14B | Unknown | 1.3B | Unknown | 14B |

denoising models (Song et al., 2023; Yin et al., 2024b,a, 2025; Lin et al., 2025; Wang et al., 2025b). Long-horizon streaming additionally requires stable autoregressive rollouts without drift (Xie et al., 2025; Ruhe et al., 2024; Chen et al., 2025a; Sun et al., 2025; Kodaira et al., 2025; Huang et al., 2025d; Liu et al., 2025b; Cui et al., 2025; Yang et al., 2025). Achieving consistent memory further requires storing long histories of past generated information—either within the KV cache or through dedicated memory mechanisms (Yu et al., 2025b; Xiao et al., 2025; Ma et al., 2025; Ren et al., 2025; Yu et al., 2025a; Huang et al., 2025c). However, satisfying these requirements simultaneously is challenging: maintaining long-term spatial consistency typically incurs substantially higher computational complexity and memory bandwidth bottleneck, which directly conflicts with the need for real-time responsiveness.

To tackle these fundamental challenges holistically, we present **RELIC**, a 14B action-controlled interactive video world model that supports both long-horizon streaming and efficient long-term memory retrieval. Our framework builds upon the Self-Forcing paradigm (Yin et al., 2025; Huang et al., 2025d), which distills an autoregressive (AR) student video model from a bidirectional teacher using the Distribution Matching Distillation (DMD) technique (Yin et al., 2024b,a). Specifically,

1. To unlock efficient long-horizon streaming and robust spatial memory retrieval, we represent the autoregressive model's memory as a set of highly compressed historical latents, in a similar spirit to (Zhang and Agrawala, 2025), which are encoded with both relative and absolute camera-pose and stored within the KV cache. This design enables implicit 3D scene-content retrieval through viewpoint-aware context alignment, while the high compression ratio allows **RELIC** to retain the entire memory history with high computational efficiency. Our method stands in contrast to approaches that maintain spatial memory through recurrent model updates (Zhang et al., 2025b; Po et al., 2025; Dalal et al., 2025), which are fundamentally constrained by the capacity of the internal model state and often tailored to specific visual domains. It also differs from methods that introduce external memory banks with handcrafted retrieval heuristics (Yu et al., 2025b; Xiao et al., 2025), or that integrate explicit 3D scene representations (Ma et al., 2025; Ren et al., 2025; Yu et al., 2025a; Huang et al., 2025c), which can introduce strong inductive biases and are frequently bottlenecked by reconstruction accuracy and runtime cost.
2. To overcome the limitations of the 5-second short-context training window used in most prior work (Lin et al., 2025; Yang et al., 2025), where a short teacher fails to provide long-horizon consistency or handle drastic viewpoint changes, both of which are essential for world modeling, we fine-tune the teacher model to generate 20-second sequences. This extended temporal horizon enables supervision that enforces spatial and temporal consistency over significantly longer trajectories. However, performing DMD over the full 20-second student rollout during distillation is computationally intractable. To address this, we introduce a replayed back-propagation technique that enables memory-efficient differentiation of student parameters by accumulating cached gradients of the DMD loss in a temporal block-wise fashion over the entire self-rollout.

We curated 350 licensed Unreal Engine–rendered scenes, encompassing approximately 1600 minutes of training video with high-quality text, action, and camera-trajectory annotations. Importantly, we collect trajectories that include both single and mixed actions, as well as trajectories with viewpoint revisitations, enabling the model to learn precise decoupled controls, flexible action composition, and long-range memory retrieval. Overall, our architectural design enables both efficient training and inference: **RELIC** achieves 16 FPS generation throughput on 4 H100 GPUs while maintaining precise action and text following, as well as long-horizon spatial consistency. We believe that the **RELIC** framework provides an effective and flexible foundation for advancing interactive world modeling, paving the way for future capabilities across numerous domains.

## 2 Related Works

***Video World Models.*** Building interactive video world models requires the integration of diverse techniques, spanning autoregressive video generation (Yin et al., 2025; Huang et al., 2025d; Teng et al., 2025; Xie et al., 2025; Henschel et al., 2025; Chen et al., 2025a; Yang et al., 2025; Kodaira et al., 2025) and few-step model distillation (Yin et al., 2024b,a; Song et al., 2023; Geng et al., 2025), alongside breakthroughs in several fundamental challenges, including long-context memory (Po et al., 2025; Song et al., 2025; Zhang et al., 2025b), spatial consistency (Xiao et al., 2025; Huang et al., 2025c; Zhang et al., 2025a; Wu et al., 2025b,a; Yu et al., 2025b), and the generation of interactive entities (Lu et al., 2025; Xia et al., 2025; Kreber and Stueckler, 2025; Luo et al., 2025; Kurai et al., 2025). Recent industrial systems such as Matrix-Game 2.0 (He et al., 2025), Magica 2 (Lab, 2025), RTFM (World Labs, 2025a), and Genie-3 (Ball et al., 2025) have demonstrated remarkable progress toward this goal, yet substantial advances are still required to make world models truly practical.

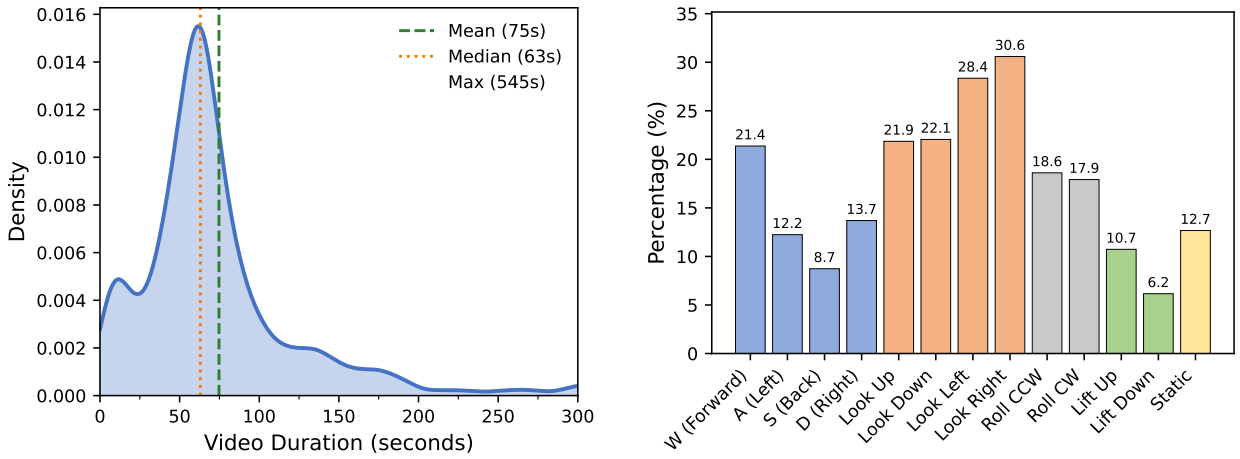
***Long Video Generation.*** The *drifting* phenomenon is a typical issue in long-horizon autoregressive video generation, manifesting as rapid degradation in video quality, color over-saturation, or even static outputs. Early training-free approaches, such as FIFO-Diffusion (Kim et al., 2024), introduce an inference strategy that operates on a window of latent frames with monotonically increasing noise levels. After a fixed number of denoising steps, a clean latent is popped out (and cached), while a new noise sample is pushed in at the other end, enabling continuous video generation. Following this idea and to bridge the train–test discrepancy, PA-VDM (Xie et al., 2025), Rolling Diffusion (Ruhe et al., 2024), SkyReels-v2 (Chen et al., 2025a), AR-Diffusion (Sun et al., 2025), StreamDiT (Kodaira et al., 2025), Rolling Forcing (Liu et al., 2025b), and Wan's Streamer (Wan et al., 2025) incorporate the same noise-scheduling patterns during training, thereby achieving stable, minute-long video generation. These scheduling strategies can be regarded as special cases of Diffusion-Forcing (Chen et al., 2024a), where latents within the modeling window may carry uneven noise levels. A key strength of this framework lies in its flexibility to incorporate context tokens during both training and inference (Song et al., 2025), effectively serving as a form of memory, an essential component for consistent world simulation. Meanwhile, *Teacher-forcing*, inspired by the next-token prediction paradigm of large language models, trains the model to predict new frames conditioned on previously generated and cached outputs (Alonso et al., 2024; Jin et al., 2024; Valevski et al., 2024; Zhou et al., 2025a; Hu et al., 2024; Gao et al., 2024). However, such methods often suffer from severe *drifting* only after a small number of rollouts. To mitigate this issue, Self-Forcing (Huang et al., 2025d) introduces self-rolling mechanisms that allow the model to adapt to its own predictions, analogous to *student-forcing* in embodied navigation, where an agent learns to correct itself when deviating from a target trajectory (Anderson et al., 2018; Hong et al., 2020). Furthermore, APT-2 (Lin et al., 2025) and LongLive (Yang et al., 2025) extend self-rolling training to minute lengths, enabling long-horizon generation.

***Data for World Exploration.*** Building a world exploration model fundamentally requires high-quality data that are dynamic, long-horizon (spanning minutes), and paired with accurate action annotations, a combination that is rare on the web and expensive to obtain (Li et al., 2025b; Wang et al., 2025a; Che et al., 2024; Feng et al., 2024; He et al., 2025). Large-scale datasets such as Sekai (Li et al., 2025b) offer abundant real-world walking and drone videos, but their action distributions are highly imbalanced and often compositional, which makes precise control learning difficult. Although AAA game datasets provide clean and reliable action–video pairs (Feng et al., 2024; Li et al., 2025a; He et al., 2025), their strong stylistic bias and limited scene diversity mean that models trained on such narrow-domain data, including specific game environments (Ye et al., 2025; Zhang et al., 2025c) or videos with repetitive camera and scene patterns (Zhang and Agrawala, 2025; Henschel et al., 2025), often produce stable long-horizon rollouts but fail to generalize to broader and more dynamic environments. In this work, we observe that when starting from a strong pretrained backbone, a small amount of carefully curated, control-precise data collected in Unreal Engine is sufficient to improve controllability while preserving generalization capability.

## 3 Data Curation for Interactive Video World Model

### 3.1 Data Overview

To support training a real-time, long-horizon interactive video world model, we construct a large-scale synthetic video dataset rendered entirely in Unreal Engine (UE). The dataset is designed to provide (i) diverse and complex




**(a)** Video duration distribution of our curated UE dataset.

**(b)** Action distribution of our curated UE dataset

**Figure 2 Dataset statistics visualization.** Left: video duration distribution; Right: action distribution.

3D environments and (ii) precise control over camera trajectory. Our final curated dataset contains over 1400 human-controlled camera trajectories collected from 350 high-quality licensed 3D scenes spanning both indoor (*e.g.*, homes, offices) and outdoor environments (*e.g.*, forests, mountains, streets). After applying the filtering procedures described in section 3.2, we obtain more than **1600 minutes** of high-fidelity 720p video sequences The clip duration distribution is shown in figure 2a, with an average of ∼75 seconds and a maximum of up to 9 minutes. We derive action labels from the corresponding camera motions (section 3.3), and sample them so that distribution of each action is well-balanced for training model. We now summarize the data curation pipeline, filtering strategy, and annotation process below.

## 3.2 Data Processing and Filtering

We begin by curating 350 photorealistic static UE scenes covering a wide range of indoor and outdoor layouts. Human operators navigate each scene using a collision-constrained camera controller to ensure physically plausible movement. During navigation, we record continuous 6-DoF camera trajectories—including positions, orientations, and the corresponding timestamps, which are then rendered into high-quality 720p video sequences using the UE renderer.

This synthetic capture pipeline is designed to address the fundamental limitations of existing real-world navigation datasets. Prior work has primarily trained models on real-world video corpora, such as FOV walking datasets and drone-based navigation videos (Wang et al., 2025a; Li et al., 2025b). However, these sources suffer from three key limitations. (1) **Imbalanced action distributions**: real-world videos are dominated by forward motion, with very limited lateral or rotational movement, which we observe prevents models from learning diverse egocentric movement behaviors. (2) **Overly coupled action behavior**: real-world videos often contain tightly coupled actions, such as turning while moving forward, which makes it difficult for the model to learn disentangled single-action control. (3) **Lack of revisitation of past viewpoints**: real-world videos rarely return to previously visited locations over long horizons, weakening the model's ability to learn long-context spatial recall and memory-based reasoning or retrieval for consistent world generation.

By contrast, our UE-rendered camera trajectories are intentionally curated to be both well-balanced in sampled action space and designed to frequently revisit locations and scene content in long-horizon settings, directly addressing the aforementioned limitations. Moreover, we observe that raw UE-rendered videos may contain various types of artifacts, including (1) overly fast or jittery camera motion, (2) unstable or collision-prone trajectories, and (3) rendering issues such as over-exposure or missing textures. These observations motivate a new filtering pipeline described below:

- **Camera Motion.** Trajectories exhibiting unnatural camera-motion patterns, such as excessive panning speed, abrupt rotations, or inconsistent velocity profiles introduced by human operators, are manually removed. This ensures that the camera dynamics remain smooth and physically plausible, preventing the model from learning unrealistic motion behavior.

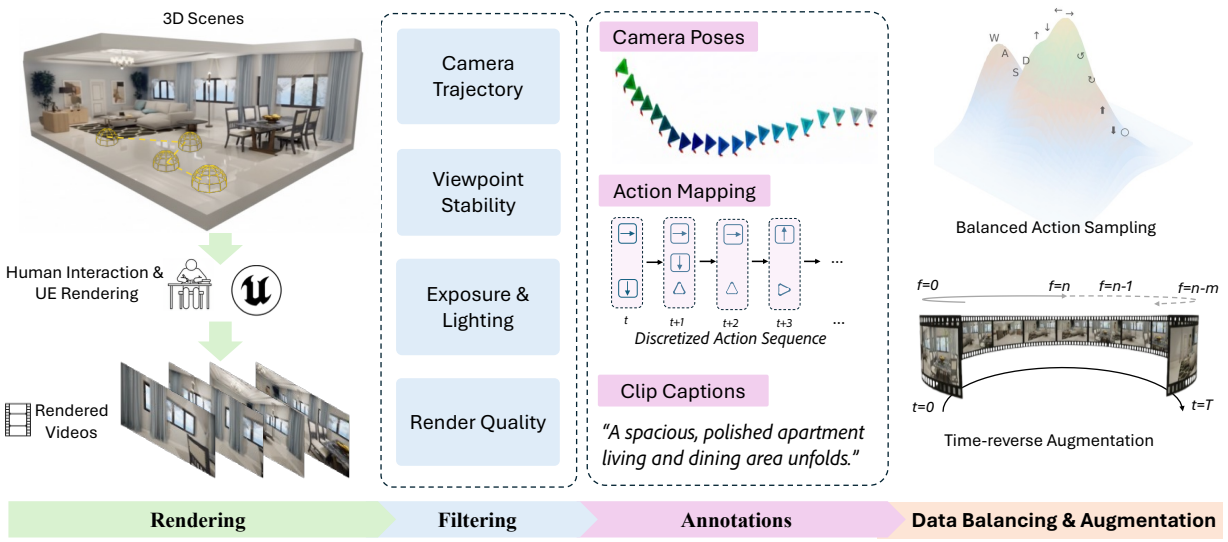



**Figure 3 The data curation pipeline in RELIC.** Given a set of 3D scenes, we manually collect thousands of camera trajectories and generate high-quality video-action-text triplets through a series of data filtering, captioning, and balancing steps.

- **Viewpoint Stability.** Segments exhibiting micro-jitters, oscillatory drift, or near-collision paths are excluded. Filtering out these unstable videos prevents high-frequency noise from contaminating the training signal.
- **Exposure and Lighting Problems.** Each clip is reviewed for exposure-related anomalies, including overexposed highlights, under-lit interiors, or illumination flicker caused by renderer settings or inconsistencies in scene assets.
- **Render Quality.** Clips with missing textures, geometry popping, incomplete meshes, or other rendering defects are discarded. Ensuring high-fidelity rendering is essential for high-quality world generation and video synthesis.

## 3.3 Data Annotations

***Camera Pose Annotations.*** For every rendered video clip $(f_1, f_2, ..., f_T)$, we record precise, frame-aligned camera pose annotations directly derived from the 6-DoF trajectories provided by Unreal Engine (UE). These camera annotations include full 6-DoF camera poses, which consist of absolute camera positions $(\mathbf{P}_1, \mathbf{P}_2, ...\mathbf{P}_T)$ and world-to-camera camera orientations $(R_1, R_2, ..., R_T)$. These complete camera annotations are essential ingredients for training interactive video world models because they allow accurate mappings between input action controls and the corresponding dynamic responses from the agent and environment. Furthermore, full 6-DoF camera poses also support effective long-context spatial content retrieval, as shown in the experiment section.

***Action Annotations.*** Although Unreal Engine records continuous 6-DoF camera trajectories, the control interface of our world model is typically structured around user actions, inputs that real users can realistically provide (*e.g.*, moving forward, panning, or looking up), rather than continuous camera poses. To bridge this gap, we convert continuous 6-DoF camera-pose sequences into per-frame action labels $\mathcal{A} \in \mathbb{R}^{13}$, *i.e.*, 13-DOF input-control format described in section 4.1.2. Our action-annotation pipeline derives per-frame actions $\mathcal{A}_t$ by comparing the relative camera poses between adjacent frames $(f_t, f_{t+1})$ in the video (see algorithm 1 for details).

Specifically, we first compute the relative camera-translation vector $\Delta\mathbf{P}_t^c$ in the camera's egocentric coordinate system using the camera positions in world coordinates $\mathbf{P}_t$ and the world-to-camera rotation matrix $R_t$ at time $t$:

$$\Delta\mathbf{P}_t^c = R_t(\mathbf{P}_{t+1} - \mathbf{P}_t). \tag{1}$$

The $x$, $y$, and $z$ components of $\Delta\mathbf{P}_t^c$ constitute the translational 3-DOF component of our action labels. This vector represents the per-frame camera displacement, *i.e.*, the instantaneous motion between two consecutive frames. Importantly, we observe that different 3D scenes may exhibit different inherent scales, analogous to the scale ambiguity discussed in the Structure-from-Motion (SfM) literature. To ensure consistent behavior across environments, we

**Algorithm 1** Action Label Extraction from UE Camera Trajectories

**Require:** Annotation JSON $\mathcal{J}$
**Ensure:** Action sequence $\mathcal{A}_{1:T}$ (including, movement action $\mathbf{A}^{\mathrm{m}}$, and rotate action $\mathbf{A}^{\mathrm{r}}$)

**function** READUEACTIONS($\mathcal{J}$)
  Load frames $\{f_t\}_{t=1}^{T}$, extract $\mathbf{P}_t$ and $R_t$
  $\Delta\mathbf{P}_t^{\mathrm{c}} \leftarrow R_t(\mathbf{P}_{t+1} - \mathbf{P}_t)$
  $\bar{d} \leftarrow$ mean movement magnitude over non-zero $\Delta\mathbf{P}_t^{\mathrm{c}}$
  **for** $t = 1$ to $T - 1$ **do**
    $\mathbf{A}_t \leftarrow$ INFERPAIRACTION($\Delta\mathbf{P}_t^{\mathrm{c}}, \bar{d}, t$)
  **end for**
  Prepend static action; **return** $\mathbf{A}_{1:T}$
**end function**
**function** INFERPAIRACTION($\Delta\mathbf{P}_t^{\mathrm{c}},\ \bar{d},\ t$)
  **// Translational motion in camera coordinates**
  $(d_f, d_s, d_z) \leftarrow \Delta\mathbf{P}_t^{\mathrm{c}}/\bar{d}$
  Assign $\mathbf{A}^{\mathrm{m}}$ : {`w/s`, `a/d`, `lifting_up/down`} from $d_f, d_s, d_z$
  **// Camera rotation**
  $\Delta R_t^c \leftarrow R_{t+1}(R_t)^{\top}$
  ($\Delta$yaw, $\Delta$pitch, $\Delta$roll) $\leftarrow$ Euler decomposition of $R_{\text{rel}}$
  Assign $\mathbf{A}^{\mathrm{r}}$ : {`camera_up/down`, `left/right`, `roll_ccw/cw`} from the angles
  Mark `static` if no action activated
  **return** action vector $(\mathbf{A}^m, \mathbf{A}^r)$
**end function**

normalize each 3D displacement vector by the average displacement magnitude of the entire clip (computed across all consecutive frame pairs), which characterizes the typical motion scale of that trajectory. This normalization interprets each $\Delta\mathbf{P}_t^c$ as a relative motion ratio, indicating how the current displacement compares to the typical displacement within the clip. During inference, the average displacement magnitude (a coefficient $\gamma$) can be adjusted to control the overall motion scale of the generated videos.

Similarly, we annotate 6-DoF rotational actions by computing the relative camera rotation $\Delta R_t^c$ between adjacent frames:

$$\Delta R_t^c = R_{t+1}(R_t)^T, \tag{2}$$

We then convert $\Delta R_t^c$ into yaw, pitch, and roll Euler angles following UE's intrinsic rotation convention. These three angles constitute the rotational portion of the action labels. Moreover, if all translational and rotational components of a relative pose are near zero, we assign the action label at the current time step to `static`.

Each frame is ultimately assigned a 13-dimensional action vector that mirrors the model's input-control interface. This structured representation enables the model to learn a consistent mapping from per-timestep discrete controls to the corresponding dynamic responses of the agent and environment. To improve action-following capability across a wide range of user-input scenarios and to reduce distributional bias during long-horizon generation, we intentionally collect a diverse and well-balanced set of actions during data acquisition, unlike many real-world datasets (Li et al., 2025b; Mao et al., 2025; Wang et al., 2025a) that are dominated by forward or dolly-in motions.

***Segmented Captions*** Long videos in our UE dataset can span multiple minutes, making a single global text caption insufficient to describe the content of the entire video. To reduce visual–caption misalignment, we split each long video into 5-second segments and generate a short, high-level caption for each segment using GPT-5 (OpenAI, 2024). During training, when the sampled video length exceeds 5 seconds, we use the text description corresponding to the first segment within the sampled video as the caption prompt for that training sample.

Because text descriptions can sometimes contradict user-intended action inputs—for example, a user may wish to move forward in the environment while the text inadvertently describes the camera moving backward or panning—we thus attempt to avoid any contradiction between text conditioning and user-input actions at inference time. To achieve this, captions are explicitly constrained to describe only static, scene-level attributes. Specifically, we prompt GPT-5

to suppress descriptions of camera motion and fine-grained object motion, as we empirically found that common vision–language models (Bai et al., 2023; Yuan et al., 2025; Chen et al., 2024b) tend to overemphasize such dynamics when captioning a video. We observe that this strategy enables more precise joint action and text control for AR video generation.

### 3.4 Data Augmentation

Although our curated camera trajectories deliberately encourage revisitation of past viewpoints, random segment sampling during training does not guarantee that most sampled sequences contain sufficient signal for the model to learn long-context memory retrieval. Therefore, to strengthen the model's spatial long-context modeling capability, we propose a simple yet effective time-reverse augmentation technique that injects controlled time-reversal structure into the training clips.

Specifically, as shown in figure 3 given a sampled training video segment of length $T$, we uniformly sample a pivot index $t^*$ from the second half of the clip, $t^* \sim \mathcal{U}(T/2, , T)$, and construct a palindrome-style training sequence by concatenating the forward segment $f_{1:t^*}$ with its time-reversed counterpart $f_{t^*:(2t^*-T)}$. This augmentation produces a clip in which the visual content naturally "looks back" in time, encouraging the model to learn effective spatial recall and to leverage long-horizon memory.

## 4 RELIC World Model

Our goal is to generate a video stream $(\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_T)$ from an input RGB image $f_0$ and a text description $c_{\text{text}}$, given a stream of action-control inputs (*e.g.*, keyboard or mouse commands) $(\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_T)$. The model must preserve spatial and temporal consistency over long horizons while running in real time with interactive responsiveness to user inputs.

Following recent work (Huang et al., 2025d; Yin et al., 2025; Shin et al., 2025), we propose a two-stage pipeline that distills a few-step autoregressive (AR) video diffusion model from a bidirectional video diffusion teacher model, where both models are conditioned on the additional action-control signal. However, unlike prior work, which typically focuses either on real-time performance (Lin et al., 2025) or on long-term generation (Yang et al., 2025; Liu et al., 2025a), achieving both capabilities simultaneously—along with long-horizon memory—is substantially more challenging, as these requirements often conflict. In particular, long-horizon spatial memory demands additional computation and GPU memory to store, transfer, and reason over past tokens, which introduces significant FLOPs and memory-bandwidth bottlenecks for real-time applications.

Our **RELIC** model is designed to address these challenges by introducing several key innovations into the framework. First, we redesign the bidirectional video diffusion teacher model to support long-duration (20-second) generation with strong action-following ability (section 4.1.1). To enable low-latency interactive streaming, we convert the teacher into an AR video-diffusion model and devise a spatial-aware memory mechanism that efficiently compresses historical video tokens within the KV cache. We further combine this with a training strategy that induces emergent memory retrieval from action–video data (section 4.2). Next, we introduce a new variant of the self-forcing training paradigm that enables memory-efficient distillation of a few-step AR model even under the teacher's 20-second long context (section 4.4). Lastly, we incorporate additional runtime optimizations that bring the model inference to real-time (section 4.5).

### 4.1 Action-Conditioned Teacher for Long Video Generation

#### 4.1.1 Base Architecture

Our framework is built upon Wan-2.1 (Wan et al., 2025), a bidirectional video diffusion model pretrained for for 5-second text-to-video and image-to-video generation. The base model consists primarily of a spatio-temporal variational autoencoder (ST-VAE) and diffusion transformers (DiTs). The ST-VAE employs a 3D causal architecture that maps between high-dimensional video space and a lower-dimensional latent token space, achieving an $8\times$ spatial compression ratio and a $4\times$ temporal compression ratio. Its temporally causal design, together with an internal feature-cache mechanism, plays a crucial role in enabling our real-time video streaming capabilities.

The video DiTs include patchifying and unpatchifying layers along with a series of transformer blocks. Input text descriptions are encoded with umT5 (Chung et al., 2023) and integrated into the model via cross-attention, while

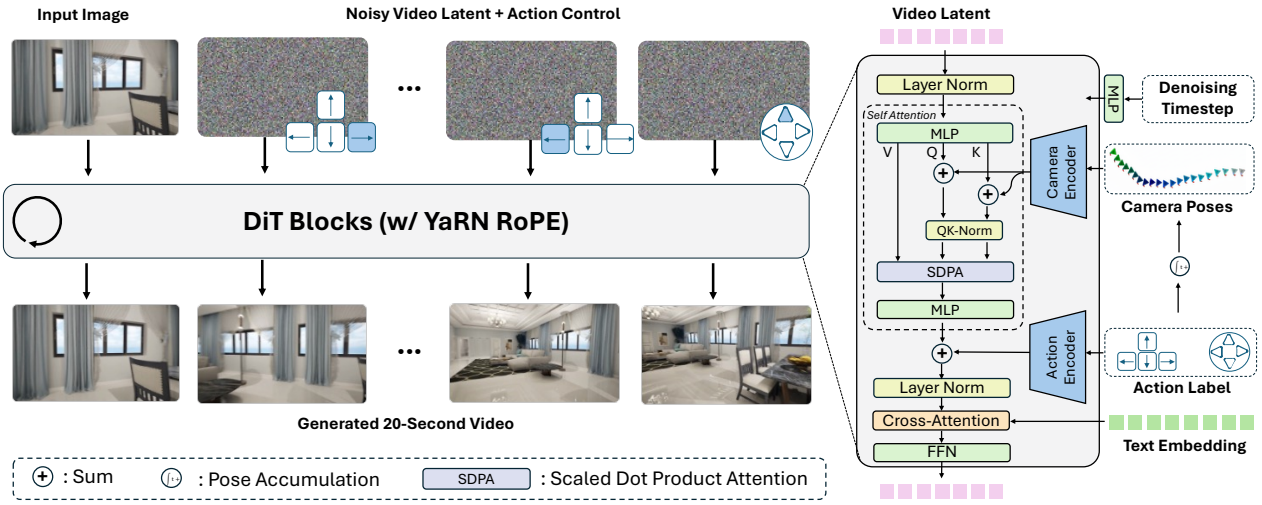



**Figure 4 Model Pipeline.** Starting from an input image and a sequence of noisy video latents, our DiT-based architecture generates a 20-second video conditioned on text, action labels, and camera poses. Each DiT block integrates YaRN-RoPE, SDPA with QK-Norm, and cross-attention to conditioning tokens. Camera and action information are embedded through dedicated encoders, and conditioning is injected throughout the denoising process to produce spatially consistent and action-aligned video frames.

denoising timestep embeddings are injected through a shared MLP module. Our **RELIC** adopts the 14B-parameter Wan-2.1 text-to-video DiT model due to its strong ability to understand 3D structure, generate consistent scene content under large camera motions, and retrieve long-term memory. Since our goal is to accept both text and an input image as conditioning signals, during both training and inference, we feed the clean latent corresponding to the input frame into the model and always set its noise level to zero when it is concatenated with other noisy video latents.

To enable precise keyboard control, long-duration video streaming, and long-horizon spatial memory—the three key capabilities required for a video world model—we must first construct a robust action-conditioned *teacher* model capable of high-quality long-form video generation from streaming action inputs. In the following section, we describe our design choices for integrating action-control signals into the base architecture and extending its generation horizon beyond the original 5-second limit.

### 4.1.2 Action Control

***Action Space.*** We design a 13-degree action space $\mathcal{A} \in \mathbb{R}^{13}$ for **RELIC**, enabling full 6-DoF camera viewpoint control. Specifically, $\mathcal{A}$ specifies the magnitudes of six translational motions—`Dolly In` $\uparrow$, `Dolly Out` $\downarrow$, `Truck Left` $\leftarrow$, `Truck Right` $\rightarrow$, `Pedestal Up`, and `Pedestal Down`—and six rotational motions—`Tilt Up` $\wedge$, `Tilt Down` $\vee$, `Pan Left` $<$, `Pan Right` $>$, `Roll Clockwise`, and `Roll Counter-Clockwise`—between consecutive frames in the video, along with a static action `[Static]` representing no camera movement. Each action is represented as a non-negative scalar value rather than a binary flag, thus it can encode the relative translational and rotational velocities induced by user inputs.

***Action Conditioning .*** To enhance action following and spatial memory retrieval, our **RELIC** model incorporates not only the 13-DoF relative action controls but also the derived 6-DoF absolute camera poses as additional conditioning signals. Because the action vector $\mathcal{A} \in \mathbb{R}^{13}$ represents relative camera motion—*i.e.*, the translational velocity $\Delta \mathbf{P}_t^c$ and rotational velocity $\Delta R_t^c$ between frames at times $t$ and $t+1$—we obtain the absolute camera poses $\mathbf{P}_t \in \mathbb{R}^3$ and $R_t \in SO(3)$ by integrating the relative motions:

$$\mathbf{P}_t = \sum_{i=1}^{t} (R_i)^T \Delta \mathbf{P}_i^c, \quad R_t = \prod_{i=1}^{t} \Delta R_i^c \tag{3}$$

Using continuous-valued action encoding allows the model to represent motion strengths consistent with observed frame transitions, enabling it to accommodate videos captured under different unit velocities. During inference, we stream a continuous sequence of user actions to the model from keyboard inputs, represented as a multi-hot vector scaled by a predefined coefficient $\gamma$, which modulates the magnitude of camera motion in the generated video.

We embed both the relative actions $\mathcal{A}$ and the absolute camera poses $(\mathbf{P}_t, R_t)$ using two dedicated encoders and inject their embeddings into every transformer block through distinct pathways, as illustrated in figure 4. In particular, each encoder consists of a temporal patchifying module followed by an MLP modulation layer that temporally compresses the control signal by $4\times$. Relative action embeddings are added directly to the latents after the self-attention layer, while absolute camera pose embeddings are added to the *query* $(Q)$ and *key* $(K)$ projections before the scaled dot-product attention (SDPA), with the *value* $(V)$ projections left unchanged. This design reflects the distinct computational roles of the two signals: relative actions guide the model in generating frame-to-frame scene transitions consistent with user control, whereas absolute camera poses act as proxies for retrieving spatial content across viewpoints and time.

#### 4.1.3 Long-Horizon Training

The original Wan-2.1 model is pretrained to generate 5-second videos (81 frames) at 16 FPS. Although recent work (Lin et al., 2025; Yang et al., 2025) has demonstrated that using a teacher model with a short context window can distill a student model capable of streaming long video sequences, we argue that a 5-second video training context length is typically not sufficient to enable the capability for long-term memory retrieval or be robust to significant camera viewpoint changes. The model must be trained directly under a long-horizon setting to learn how to restore spatial scene contents previously seen through the memory. Therefore, the first requirement is to enable the teacher model having the capacity for long-duration video generation, and we start to fine-tune our action-conditioned video diffusion model on our action-video dataset (section 3) to extend its generation duration to 20-seconds (317 frames). In particular, we employ a curriculum learning strategy: the model is first trained on 5-second videos for 5,000 iterations, followed by 10-second videos for 1,000 iterations, and finally by 20-second videos for another 4,000 iterations. To facilitate more rapid adaptation to longer sequences, we apply the YaRN technique (Peng et al., 2023) to extend the Rotary Positional Embeddings (RoPE) (Su et al., 2024) for the query and key tokens in each self-attention layer of the DiT block.

### 4.2 Autoregressive Student for Real-Time Streaming

Designing our final interactive video world model entails three core challenges: **(1) Memory**, *i.e.* the ability to recall and faithfully re-render previously generated scenes when the camera agent revisits earlier viewpoints; **(2) Streaming inference**, to enable low-latency, real-time interactive exploration; and **(3) Long-horizon generation**, to provide sufficiently extended temporal context for user navigation and discovery.

To address these challenges, we distill our 20-second bidirectional teacher model into a few-step, memory-aware causal student video diffusion model. Our student model is also built upon Wan2.1-14B (Wan et al., 2025), but replaces bidirectional attention with block-wise causal attention and generates latent frames in a block-causal manner, following similar principles as recent autoregressive video diffusion approaches (Yin et al., 2025; Huang et al., 2025d; Yang et al., 2025). We describe our memory mechanisms and training strategies below, and refer readers to these prior works for the full mathematical formulation of autoregressive video generation.

### 4.3 Memory

Most recent autoregressive (AR) video models adopt causal attention over a short sliding window to enable efficient long-video streaming (Yang et al., 2025; Shin et al., 2025). While this strategy effectively reduces inference latency and improves throughput, it inherently limits the model's ability to retrieve long-range information, which is crucial for consistent world modeling. A straightforward solution to restore such long-horizon consistency is to retain all past tokens in the KV cache during AR inference. However, both the KV-cache memory footprint and the per-token attention cost scale linearly with sequence length. Consequently, as video length increases during AR generation, both computation and communication overhead grow proportionally, making real-time streaming inference particularly challenging.

To address this dilemma, we introduce a memory-compression mechanism, illustrated in figure 6. Given a newly denoised latent at index $i$, our KV cache is composed of two branches: a rolling-window cache and a compressed long-horizon spatial memory cache. The rolling-window cache stores uncompressed KV tokens for recent video latent between indices $i - w$ and $i$, where $w$ denotes the sliding-window size. We maintain a small rolling window to prevent

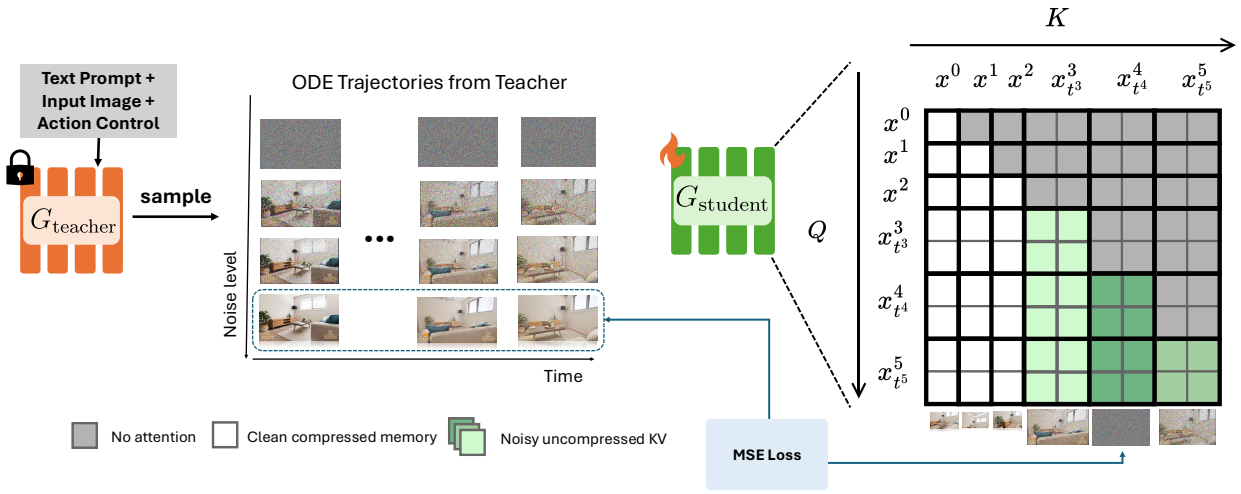



**Figure 5 ODE initialization.** We convert a bidirectional video diffusion model into a causal generator by initializing the student on a set of ODE trajectories obtained from the teacher. To achieve this, we adopt a hybrid forcing strategy that combines teacher forcing and diffusion forcing (mask shown on the right).

the model from relying solely on short-term patterns and encourage effective learning from the compressed long-range memory. The compressed long-horizon spatial memory cache, in contrast, stores spatially downsampled KV tokens for video latent from the beginning of the sequence up to index $i - w$, following a predefined compression schedule. In practice, we adopt an empirically balanced configuration that interleaves spatial downsampling factors of $1\times$ (no compression), $2\times$, and $4\times$ across latents. This design is motivated by the observation that VAE-encoded latent spaces exhibit substantial spatial redundancy, allowing much of the original information to remain recoverable after moderate spatial compression. As a result, the model can still reconstruct high-fidelity content from the compressed long-range context. On average, our strategy reduces the total token count by approximately $4\times$ (*e.g.*, from 120K to 30K), which in turn yields a proportional $4\times$ reduction in both KV-cache memory and attention FLOPs.

## 4.4 Distillation Framework

We build our distillation framework on prior self-forcing (Huang et al., 2025d), an effective AR distillation paradigm designed to reduce exposure bias. Self-forcing simulates inference-time AR rollouts during training by predicting new chunks conditioned on the model's own previously generated tokens (*i.e.*, using generated history rather than ground-truth context).

Many recent long-video distillation pipelines (Liu et al., 2025a; Yang et al., 2025; Cui et al., 2025; Lin et al., 2025) train a long-horizon student model using a teacher that generates only short video segments (*e.g.*, 5-second windows from the Wan-2.1 teacher). However, this design decomposes long videos into loosely coupled short segments, limiting the student's ability to retrieve and reason over long-range spatial memory. In contrast, we distill from a bidirectional teacher trained on much longer 20-second clips, enabling a direct alignment between the student's long-video distribution and the teacher's long-video distribution.

### 4.4.1 ODE Initialization with Hybrid Forcing

Following recent AR video distillation frameworks (Yin et al., 2025; Huang et al., 2025d), we adopt the same ODE-initialization procedure to adapt the bidirectional teacher into a causal model and to enable the student to utilize long-horizon spatial memory. The student is initialized with the teacher's weights and trained to regress precomputed ODE trajectories at the four denoising time steps used during distillation and inference.

Interestingly, we found that using teacher forcing or diffusion forcing alone produces suboptimal initialization for causal distillation. We therefore introduce an improved ODE initialization strategy that combines both paradigms to facilitate fast convergence. Specifically, given a training sequence of $B$ latent blocks, we divide them into two chunks: the first $B - K$ blocks contain clean, spatially compressed latents, while the remaining $K$ blocks contain uncompressed

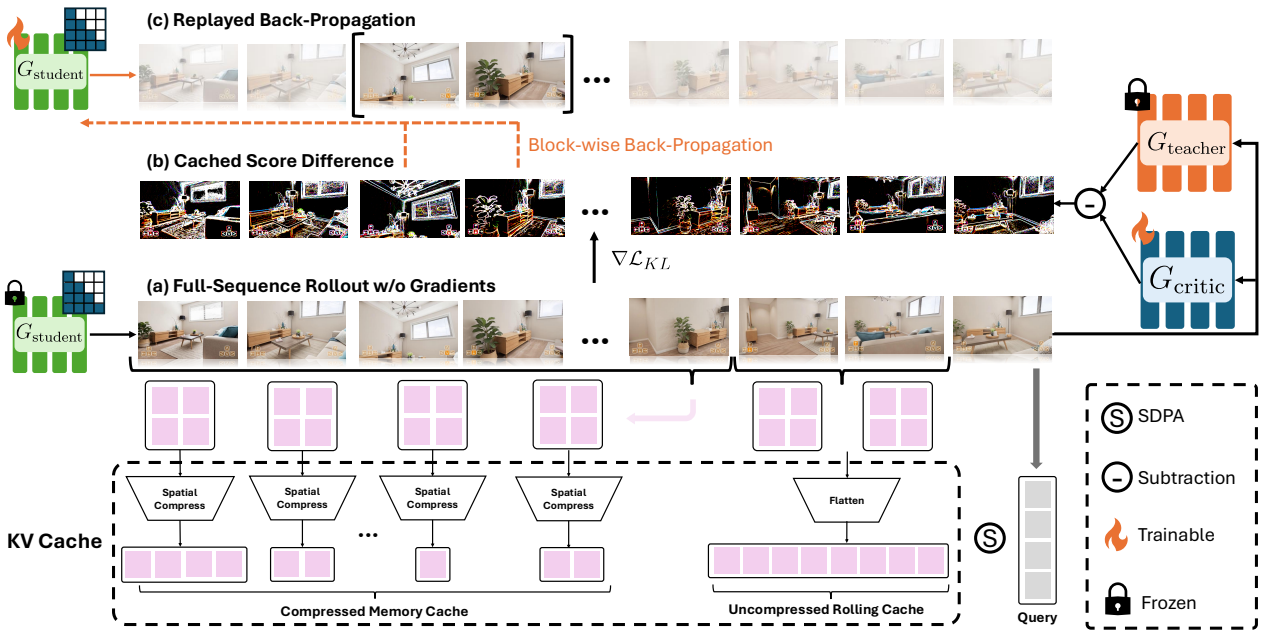



**Figure 6 Long video distillation with replayed back-propagation.** Given a 20-second bidirectional teacher, we distill it into a fast autoregressive student model via self-forcing (Huang et al., 2025d). We achieve memory-efficient distillation via replayed back-propagation. Specifically, (a) we first let the student model generate the full sequence via self-rollout with gradients disabled. (b) Then we compute and cache the DMD score difference maps over the entire predicted sequence using the critic and teacher models. (c) We re-run the student forward pass with auto-differentiation enabled and back-propagate the corresponding cached score difference maps to accumulate parameter gradients. Parameters are updated once after the full replay.

latents. For each block in the second chunk, we add noise at varying scales and condition the block causally on (i) past uncompressed latents within the second chunk, as in diffusion forcing, and (ii) the clean compressed latents from the first chunk, as in teacher forcing, as illustrated in figure 5. This hybrid strategy provides a stronger warm-up for long-horizon memory retrieval compared to either forcing method alone.

### 4.4.2 Long-Video Distillation with Replayed Back-propagation

Self-forcing (Huang et al., 2025d) employs the Distribution Matching Distillation (DMD) loss (Yin et al., 2024a,b), which minimizes the reverse KL divergence between the diffused data distribution and the student distribution across sampled timesteps $u$. The gradient of the objective can be approximated by the difference between the real-data and generated-data score functions, $s^{\text{data}}$ and $s^{\text{gen}}$:

$$\nabla_\theta \mathcal{L}_{KL} \approx -\mathbb{E}_u \left[ \int \left( s^{\text{data}}(\Psi(G_\theta(\epsilon, c_{\text{text}}), u) - s^{\text{gen}}(\Psi(G_\theta(\epsilon, c_{\text{text}}), u)) \right) \frac{dG_\theta(\epsilon, , c_{\text{text}})}{d\theta} \, d\epsilon \right], \tag{4}$$

where $\Psi$ denotes the forward diffusion process, $\epsilon$ is Gaussian noise, and $G_\theta$ is the student generator.

The original self-forcing implementation becomes prohibitively memory-intensive for long-video distillation: the student must roll out an entire long video with autograd enabled, and the DMD loss must be back-propagated through the full computation graph. Since the graph grows linearly with video length, this quickly becomes intractable for long rollout scenarios.

To address this limitation, we introduce a *replayed back-propagation* technique that stores only a small computation graph corresponding to a single generation block during self-rollout. This idea is related in spirit to prior work on neural rendering (Zhang et al., 2022), but our method incorporates substantial modifications to support distillation. As shown in figure 6, we first generate the entire predicted sequence of $L$ video latents using $G_\theta$ with autograd disabled,

$$\hat{\mathbf{x}}_{0:L} = \text{stop-grad}(G_\theta(\epsilon_{0:L})),$$

and compute the score-difference maps using frozen real and fine-tuned fake score models:

$$\Delta\hat{s}_{0:L} = s^{\text{data}}(\hat{\mathbf{x}}_{0:L}) - s^{\text{gen}}(\hat{\mathbf{x}}_{0:L})$$

Next, we replay the AR rollout block by block. For each block index $l$, we re-run the student AR forward pass with autograd enabled, conditioned on the previously generated context, and then back-propagate the corresponding cached score-difference map to update the gradients:

$$\nabla_\theta \mathcal{L}_{KL} \approx \sum_{l=1}^{L} -\Delta\hat{s}_l \frac{\partial G_\theta}{\partial \theta}. \tag{5}$$

After processing block $i$, its computation graph is immediately freed before moving to the next block. Parameters are updated once after the full replay. This approach shifts back-propagation from full-sequence differentiation to block-wise differentiation, reducing peak GPU memory from that of an entire rollout to that of a single video-latent block, while still capturing gradients that reflect the full-length video distribution of the teacher.

### 4.5 Runtime Efficiency Optimization

To provide users with a real-time experience, we optimize our codebase for efficient inference. Since inference latency and throughput are largely bounded by GPU memory bandwidth and CPU speed, we tailor our optimizations accordingly. We first apply `torch.compile` to reduce kernel launch overhead and memory costs, *e.g.*, for RMSNorm, RoPE embeddings, and modulation layers. We also use a KV cache in self-attention to avoid recomputing historical context, and we store the cache in FP8 E4M3 format to halve memory usage and reduce memory transfer time during inference. We also employ FlashAttention v3 (Shah et al., 2024) with FP8 kernels to improve performance on NVIDIA Hopper GPUs. Finally, we manually fuse and reflow certain PyTorch operations based on profiling results to further reduce overhead.

After minimizing memory and CPU costs, we utilize parallelization to shard computation and memory loads across multiple GPUs. We adopt a parallelization strategy similar to that used in our long-sequence training (section 5.1); specifically, all linear layers and cross-attention modules are parallelized over the sequence dimension (sequence/context parallelism), while self-attention operators are parallelized over attention heads (tensor parallelism). We use NCCL `All-to-All` operations to switch tensor layouts between these two parallelization schemes. For example, when transitioning from sequence parallelism to tensor-parallel attention, an `All-to-All` operation scatters along the heads dimension and gathers along the sequence-length dimension simultaneously. We use tensor parallelism for self-attention because it enables sharding of the KV cache across devices, ensuring that each GPU stores only the heads it is responsible for computing.

## 5 Experiments

In this section, we detail our implementation setup, demonstrate the capabilities of **RELIC**, and compare its performance against existing methods.

### 5.1 Implementation details.

Training a 20-second 14B base model can be challenging as the model needs to process more than 120K tokens for a forward pass. This creates high pressure on GPU memory. To deal with this, we use a combination of existing techniques in both bidirectional long-horizon training and auto-regressive distillation stages: (i) employ Fully Sharded Data Parallel (FSDP) (Zhao et al., 2023) to shard training batch, model parameters, gradients and optimization states over GPUs; (ii) leverage sequence parallelism (Li et al., 2023) to scatter the sequence; and (iii) use tensor parallelism (Shoeybi et al., 2019) to distribute attention heads across GPUs. This linearly reduces the memory requirement by the number of available computes. For the student model, our memory spatial compression configuration is empirically set to $S = [1, 4, 2, 4, 4, 4, 2, 4, 4, 2, 4, 4, 4, 2, 4, 4, 2, 4]$ to fit a 20-second context into the original pre-trained token context length corresponding to a 5-second clip. We apply this configuration recurrently, where the compression ratio for latent frame $i$ is selected as $s_i = S[i \bmod \text{len}(S)]$. During distillation, we progressively increase the rollout length to keep training stable and facilitate convergence. Specifically, we first roll out 5-second sequences for 250 training iterations

**Table 2 Quantitative comparison.** We compare RELIC with recent representative open-source world models on **20s video clips**. †We compute Subject Consistency, Background Consistency, Motion Smoothness, Dynamic Degree, Aesthetic Quality, Imaging Quality, and then calculate the average score over them.

| Model | Visual quality ↑ | | | Action accuracy (RPE ↓) | |
|---|---|---|---|---|---|
| | Average Score† | Image Quality | Aesthetic | Trans | Rot |
| Matrix-Game-2.0 (He et al., 2025) | 0.7447 | 0.6551 | 0.4931 | 0.1122 | 1.48 |
| Hunyuan-GameCraft (Li et al., 2025a) | 0.7885 | **0.6737** | 0.5874 | 0.1149 | 1.23 |
| RELIC (ours) | **0.8015** | 0.6665 | **0.5967** | **0.0906** | **1.00** |

using the intermediate 5-second teacher, then extend the rollout to 10 seconds with the 10-second teacher for another 150 iterations, and finally increase the horizon to 20 seconds with the 20-second teacher for the last 150 iterations. We further optimize decoding-time efficiency by replacing the original VAE with the same Tiny VAE used in MotionStream (Shin et al., 2025). Our final model is trained on 32 H100 GPUs, each with 80GB of memory.

## 5.2 Capabilities Showcase

RELIC achieves high-quality, diverse, and precisely controllable video generation while maintaining strong spatial consistency over long horizons. figure 1, figure 7, and figure 9 illustrate these capabilities across a wide range of scenes and styles, and we refer readers to the videos on our project page for full results.

***Diversity.*** RELIC generalizes far beyond real indoor or outdoor environments. Starting from a single initial frame, it can explore worlds depicted in oil paintings, comic illustrations, vector art, low-poly renders, and various other stylized domains (figure 7(a)). Notably, the model naturally exhibits correct distance awareness—faraway elements move more slowly than nearby objects—and demonstrates strong 3D shape understanding as the camera moves around subjects. This broad generalization capability enables RELIC to be applied across a wide range of creative and visual settings.

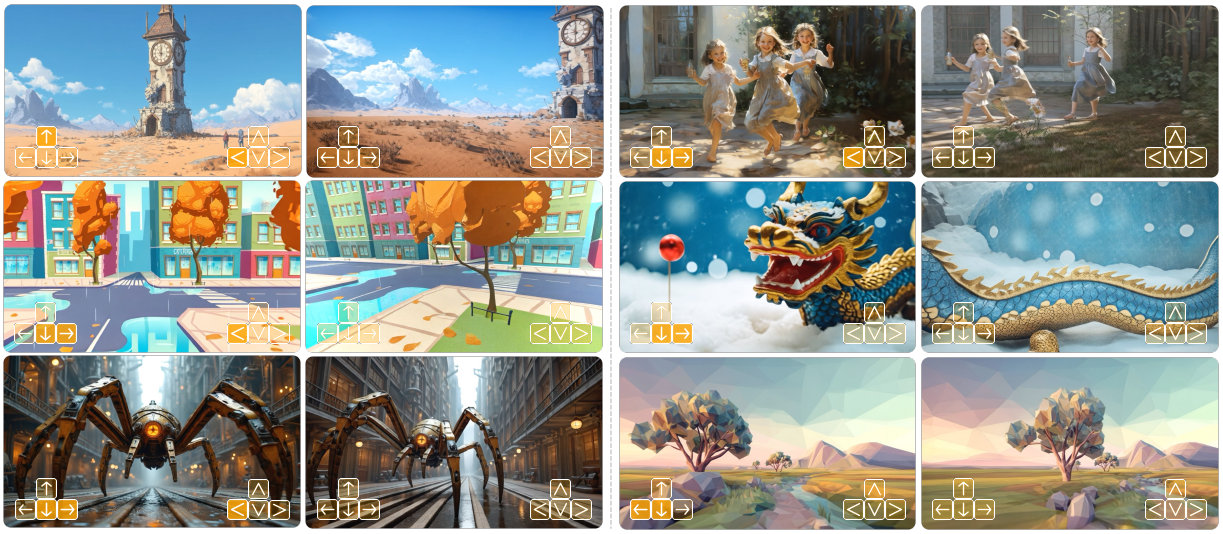
***Long-horizon Memory.*** Our RELIC model enables robust spatial memory retrieval even under large camera movements, as shown in the first row of figure 1 and figure 9. The model can accurately recover previously generated scene content with minimal loss of detail, even when that content has remained outside the camera's field of view for an extended period. This is particularly noteworthy given that our model relies solely on compressed historical video tokens and contexts from absolute camera pose, without introducing any explicit 3D scene representation, handcrafted memory heuristics, or auxiliary hyper-networks.

***Adjustable Velocity.*** Because camera actions are represented as continuous relative velocities, users can freely control the exploration speed by adjusting the displacement coefficient $\lambda$. As shown in figure 7(b), RELIC supports a wide spectrum of translational and rotational velocities while consistently producing high-quality, temporally stable outputs.

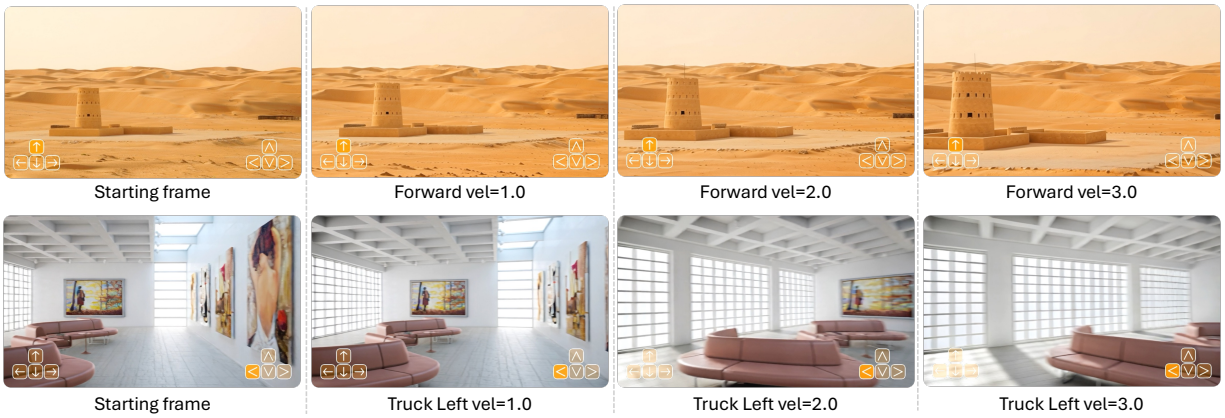
***Multi-Key Control.*** RELIC responds reliably to complex multi-key inputs that combine both translation and rotation actions, enabling rich and intuitive interaction with the generated world. This high degree of motion freedom allows users to explore the scenes in real time with precision and flexibility.

## 5.3 Quantitative Comparison

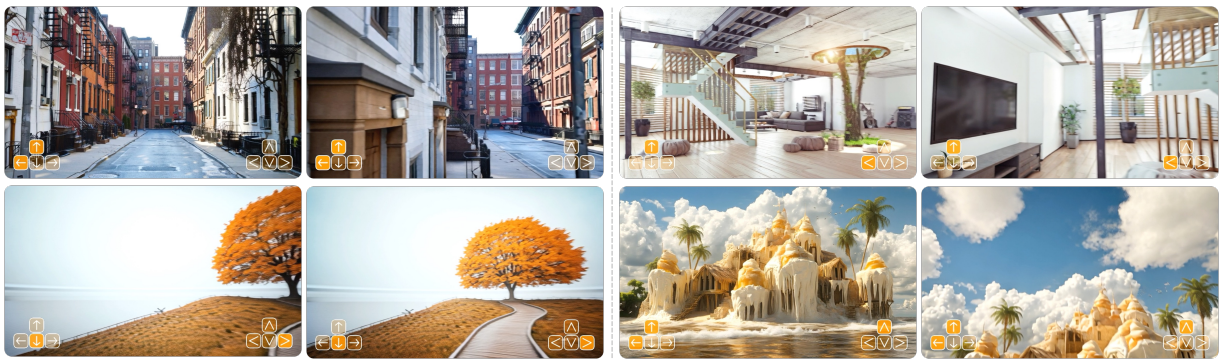
We construct a benchmark test set of 220 images sourced from Adobe Stock. The set spans both realistic scenes (landscapes, urban environments, indoor spaces) and non-realistic scenes (cartoons, vector art, oil paintings). The 220 images are randomly partitioned into 11 groups. For each group, we evaluate all baseline models using a predefined action script, resulting in 220 generated videos per baseline. The output duration is fixed at 20 seconds. We further evaluate each baseline from two aspects: visual quality and action accuracy. We compare with two state-of-the-art baselines: Matrix-Game-2.0 (He et al., 2025) and Hunyuan-GameCraft (Li et al., 2025a).



(a) Generalize to diverse Art Styles




(b) Adjustable Velocity



(c) Multi-key Control

**Figure 7** RELIC generates high-quality, diverse, and controllable videos while maintaining strong spatial consistency over long horizons. The figure showcases its generalization across varied artistic styles, its ability to follow adjustable camera velocities, and its support for complex multi-key control.

***Visual quality.*** We evaluate visual quality using selected dimensions from VBench (Huang et al., 2024), with results summarized in table 1. **RELIC** achieves the strongest overall performance among all compared baselines. Although trained at 480p resolution, it performs comparably to Hunyuan-GameCraft (Li et al., 2025a), which is trained on 720P videos, in terms of Image Quality. Notably, it attains higher Aesthetics scores and significantly outperforms Matrix-Game 2.0 (He et al., 2025) across metrics.

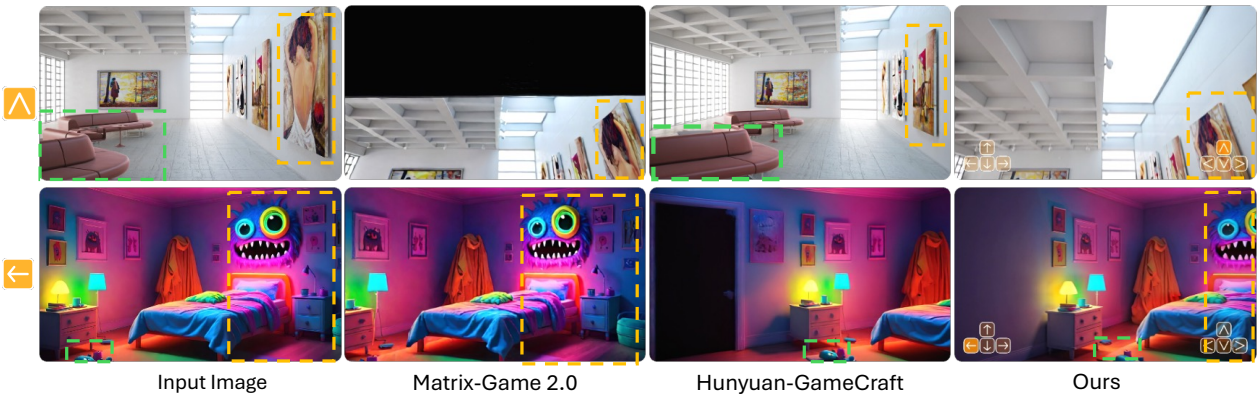



**Figure 8 Qualitative comparison of action control.** In the "gallery" example (top row), Hunyuan-GameCraft pans the camera left instead of rotating it upward (*i.e.* `Tile Up`). Matrix-Game-2.0 nominally follows the intended action, but it introduces a large black region at the top of the frame. In the "bedroom" example (bottom row), Hunyuan-GameCraft rotates the camera instead of translating left. Matrix-Game-2.0 drifts left instead of executing the commanded leftward rotation.

***Action Accuracy.*** All baselines support control via the same set of discrete keyboard actions (W/A/S/D translation and camera rotation), although their internal mappings from actions to camera speed may differ. To ensure a fair comparison, we apply a behavioral evaluation protocol: every model executes the same predefined action sequence, and we reconstruct its induced camera trajectory from the generated video using ViPE (Huang et al., 2025a). The reconstructed trajectory is aligned to the canonical ground-truth camera trajectory using a Sim(3) Umeyama alignment (Umeyama, 2002), which removes scale and coordinate-frame differences. We then evaluate translational and rotational Relative Pose Error (RPE-trans, RPE-rot) to measure how closely each model follows the intended camera trajectory, independent of internal speed tuning. We show the results in table 2. **RELIC** demonstrates the most faithful adherence to the target motion, resulting in the lowest overall RPE.

## 5.4 Qualitative Comparison

***Action accuracy.*** We compare the accuracy of action control in figure 8. **RELIC** adheres to the commanded actions most faithfully, producing motion that stays closest to the intended trajectory. For example, when applying `Tilt Up` ⋀, Matrix-Game-2.0 (He et al., 2025) fails to generate new content at the top image boundary, resulting in a black void, while Hunyuan-GameCraft (Li et al., 2025a) exhibits negligible vertical camera movement. Similarly, when applying `tuck Left` ←, Hunyuan-GameCraft behaves more like `Pan Left` <, and Matrix-Game-2.0 incorrectly remains static instead of a lateral movement. In contrast, **RELIC** accurately follows the trajectory, revealing the ceiling structure and shifting the viewing angle correctly without artifacts.

***Memory.*** Due to the difficulty in aligning all baselines with the same actions, we show a qualitative comparison on memory in figure 9. Previous baselines such as Hunyuan-GameCraft (Li et al., 2025a) and Matrix-Game-2.0 (He et al., 2025) fail to maintain object persistence in this scenario. For example, Hunyuan-GameCraft forgets the bench once the viewpoint moves away and returns, and Matrix-Game 2.0 quickly loses the context of the input image, whereas our model consistently regenerates the previously observed content (also see figure 1).

***Comparison with Marble.*** We also qualitatively compare our method with Marble (World Labs, 2025b), a commercial system whose output is based on Gaussian-splatting reconstruction (Kerbl et al., 2023), in figure 10. Since Marble's final output is rendered frames from Gaussian Splatting, it inherently produces artifacts such as Gaussian floaters, which are visible in its results. In contrast, **RELIC** avoids these reconstruction artifacts and delivers clean, stable outputs.

# 6 Discussion and Conclusion

**Limitations.** Our system still exhibits several limitations. First, the generated videos demonstrate limited diversity and restricted scene dynamics, primarily due to training on datasets composed mostly of static scenes rendered from

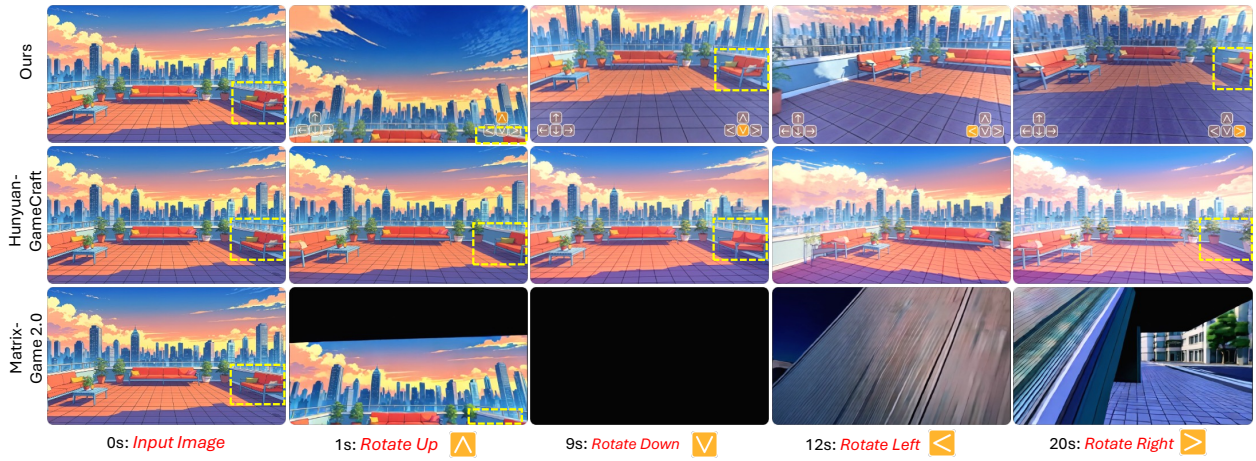



**Figure 9 Qualitative comparison of memory.** Previous methods, such as Hunyuan-GameCraft (Li et al., 2025a) and Matrix-Game-2.0 (He et al., 2025), forget the bench on the right-hand side in this case quickly.

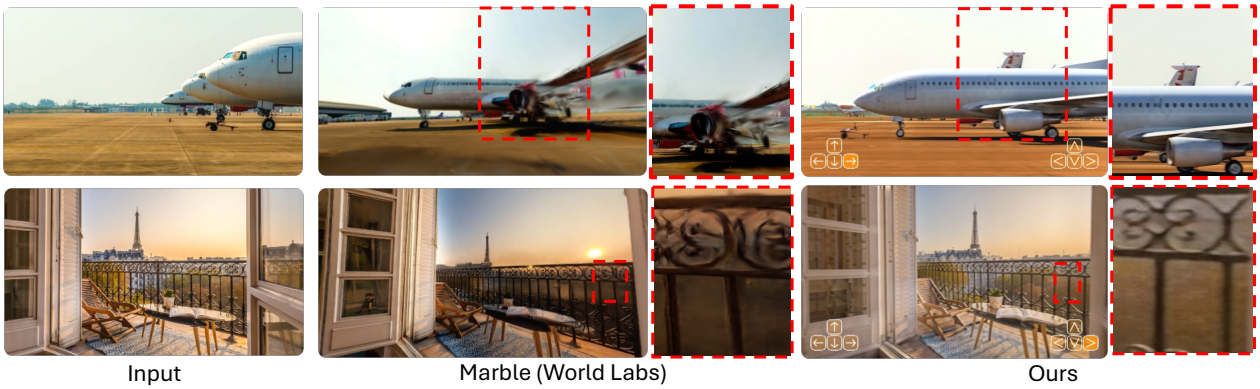



**Figure 10 Qualitative comparison with Marble from World Labs** (World Labs, 2025b). We compare with a commercial solution, Marble. Since the final product from Marble is a Gaussian splatting (Kerbl et al., 2023) rendered result, it inevitably introduces artifacts such as Gaussian floaters. Our method **RELIC** , instead, generates clean results.

the Unreal Engine. Additionally, our approach struggles to generate extremely long videos—on the scale of minutes. Moreover, the combination of large model size, KV cache requirements for long-horizon memory, and multiple iterative denoising steps significantly impacts inference latency under resource-constrained settings. Nonetheless, we believe these issues can be mitigated through targeted refinements to the pipeline and appropriate scaling of data and training.

**Conclusion.** In this work, we presented **RELIC**, an interactive video world model that enables real-time inference and long-horizon spatial memory for virtual scene exploration. By integrating a lightweight, spatial-aware memory mechanism with the scalable Self-Forcing distillation paradigm, **RELIC** enables consistent world generation from a single image, without relying on explicit geometric representations. Our method shows that integrating compressed historical latents with full-horizon supervision from a long-context teacher can successfully address challenges such as drifting and memory forgetting, which have long hindered video generation. The architectural innovations in **RELIC** lay a scalable and adaptable foundation for general-purpose world simulators, with potential applications in embodied AI and immersive virtual content creation.

# 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.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Philip J. Ball, Jakob Bauer, Frank Belletti, Bethanie Brownfield, Ariel Ephrat, Shlomi Fruchter, Agrim Gupta, Kristian Holsheimer, Aleksander Holynski, Jiri Hron, Christos Kaplanis, Marjorie Limont, Matt McGill, Yanko Oliveira, Jack Parker-Holder, Frank Perbet, Guy Scully, Jeremy Shar, Stephen Spencer, Omer Tov, Ruben Villegas, Emma Wang, Jessica Yung, Cip Baetu, Jordi Berbel, David Bridson, Jake Bruce, Gavin Buttimore, Sarah Chakera, Bilva Chandra, Paul Collins, Alex Cullum, Bogdan Damoc, Vibha Dasagi, Maxime Gazeau, Charles Gbadamosi, Woohyun Han, Ed Hirst, Ashyana Kachra, Lucie Kerley, Kristian Kjems, Eva Knoepfel, Vika Koriakin, Jessica Lo, Cong Lu, Zeb Mehring, Alex Moufarek, Henna Nandwani, Valeria Oliveira, Fabio Pardo, Jane Park, Andrew Pierson, Ben Poole, Helen Ran, Tim Salimans, Manuel Sanchez, Igor Saprykin, Amy Shen, Sailesh Sidhwani, Duncan Smith, Joe Stanton, Hamish Tomlinson, Dimple Vijaykumar, Luyu Wang, Piers Wingfield, Nat Wong, Keyang Xu, Christopher Yew, Nick Young, Vadim Zubov, Douglas Eck, Dumitru Erhan, Koray Kavukcuoglu, Demis Hassabis, Zoubin Gharamani, Raia Hadsell, Aäron van den Oord, Inbar Mosseri, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 3: A new frontier for world models, 2025. https://deepmind.google/discover/blog/genie-3-a-new-frontier-for-world-models/.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. V-jepa: Latent video prediction for visual representation learning, 2023.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators, 2024. https://openai.com/research/video-generation-models-as-world-simulators.

Haoxuan Che, Xuanhua He, Quande Liu, Cheng Jin, and Hao Chen. Gamegen-x: Interactive open-world game video generation. *arXiv preprint arXiv:2411.00769*, 2024.

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, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, et al. Skyreels-v2: Infinite-length film generative model. *arXiv preprint arXiv:2504.13074*, 2025a.

Junyi Chen, Haoyi Zhu, Xianglong He, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Yang Zhou, Zizun Li, Zhoujie Fu, Jiangmiao Pang, and Tong He. Deepverse: 4d autoregressive video generation as a world model. *arXiv preprint arXiv:2506.01103*, 2025b.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198, 2024b.

Hyung Won Chung, Noah Constant, Xavier García, Adam Roberts, Yi Tay, Sharan Narang, and Orhan Firat. Unimax: Fairer and more effective language sampling for large-scale multilingual pretraining. *ArXiv*, abs/2304.09151, 2023. https://api.semanticscholar.org/CorpusID:258187051.

Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Self-forcing++: Towards minute-scale high-quality video generation. *arXiv preprint arXiv:2510.02283*, 2025.

Karan Dalal, Daniel Koceja, Jiarui Xu, Yue Zhao, Shihao Han, Ka Chun Cheung, Jan Kautz, Yejin Choi, Yu Sun, and Xiaolong Wang. One-minute video generation with test-time training. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17702–17711, 2025.

Ruili Feng, Han Zhang, Zhantao Yang, Jie Xiao, Zhilei Shu, Zhiheng Liu, Andy Zheng, Yukun Huang, Yu Liu, and Hongyang Zhang. The matrix: Infinite-horizon world generation with real-time moving control. *arXiv preprint arXiv:2412.03568*, 2024.

Kaifeng Gao, Jiaxin Shi, Hanwang Zhang, Chunping Wang, Jun Xiao, and Long Chen. Ca2-vdm: Efficient autoregressive video diffusion model with causal generation and cache sharing. *arXiv preprint arXiv:2411.16375*, 2024.

Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, Liang Li, Xiaojie Li, et al. Seedance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113*, 2025.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

Jinwei Gu. Cosmos world foundation models for physical ai. In *Proceedings of the 3rd International Workshop on Rich Media With Generative AI*, pages 39–39, 2025.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3), 2018.

Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, et al. Matrix-game 2.0: An open-source, real-time, and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025.

Roberto Henschel, Levon Khachatryan, Hayk Poghosyan, Daniil Hayrapetyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Streamingt2v: Consistent, dynamic, and extendable long video generation from text. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2568–2577, 2025.

Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. A recurrent vision-and-language bert for navigation. *arXiv preprint arXiv:2011.13922*, 2020.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Jinyi Hu, Shengding Hu, Yuxuan Song, Yufei Huang, Mingxuan Wang, Hao Zhou, Zhiyuan Liu, Wei-Ying Ma, and Maosong Sun. Acdit: Interpolating autoregressive conditional modeling and diffusion transformer. *arXiv preprint arXiv:2412.07720*, 2024.

Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, Jiawei Ren, Kevin Xie, Joydeep Biswas, Laura Leal-Taixe, and Sanja Fidler. Vipe: Video pose engine for 3d geometric perception. In *NVIDIA Research Whitepapers*, 2025a.

Siqiao Huang, Jialong Wu, Qixing Zhou, Shangchen Miao, and Mingsheng Long. Vid2world: Crafting video diffusion models to interactive world models. *arXiv preprint arXiv:2505.14357*, 2025b.

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

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in neural information processing systems*, 2025d.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. Fifo-diffusion: Generating infinite videos from text without training. *Advances in Neural Information Processing Systems*, 37:89834–89868, 2024.

Akio Kodaira, Tingbo Hou, Ji Hou, Masayoshi Tomizuka, and Yue Zhao. Streamdit: Real-time streaming text-to-video generation. *arXiv preprint arXiv:2507.03745*, 2025.

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.

Jens U Kreber and Joerg Stueckler. Guiding diffusion-based articulated object generation by partial point cloud alignment and physical plausibility constraints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3206–3214, 2025.

Ryutaro Kurai, Takefumi Hiraki, Yuichi Hiroi, Yutaro Hirao, Monica Perusquía-Hernández, Hideaki Uchiyama, and Kiyoshi Kiyokawa. Magiccraft: Natural language-driven generation of dynamic and interactive 3d objects for commercial metaverse platforms. *arXiv preprint arXiv:2504.21332*, 2025.

Dynamics Lab. Magica 2, 2025. https://blog.dynamicslab.ai/.

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

Shenggui Li, Fuzhao Xue, Chaitanya Baranwal, Yongbin Li, and Yang You. Sequence parallelism: Long sequence training from system perspective. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2391–2404, 2023.

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

Shanchuan Lin, Ceyuan Yang, Hao He, Jianwen Jiang, Yuxi Ren, Xin Xia, Yang Zhao, Xuefeng Xiao, and Lu Jiang. Autoregressive adversarial post-training for real-time interactive video generation. *arXiv preprint arXiv:2506.09350*, 2025.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. *arXiv preprint arXiv:2509.25161*, 2025a.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. *arXiv preprint arXiv:2509.25161*, 2025b.

Ruijie Lu, Yu Liu, Jiaxiang Tang, Junfeng Ni, Yuxiang Wang, Diwen Wan, Gang Zeng, Yixin Chen, and Siyuan Huang. Dreamart: Generating interactable articulated objects from a single image. *arXiv preprint arXiv:2507.05763*, 2025.

Rundong Luo, Haoran Geng, Congyue Deng, Puhao Li, Zan Wang, Baoxiong Jia, Leonidas Guibas, and Siyuan Huang. Physpart: Physically plausible part completion for interactable objects. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12386–12393. IEEE, 2025.

Baorui Ma, Huachen Gao, Haoge Deng, Zhengxiong Luo, Tiejun Huang, Lulu Tang, and Xinlong Wang. You see it, you got it: Learning 3d creation on pose-free videos at scale. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2016–2029, 2025.

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

OpenAI. Gpt-5.1, 2024. https://www.openai.com. Large language model.

Jack Parker-Holder, Philip Ball, Jake Bruce, Vibhavari Dasagi, Kristian Holsheimer, Christos Kaplanis, Alexandre Moufarek, Guy Scully, Jeremy Shar, Jimmy Shi, Stephen Spencer, Jessica Yung, Michael Dennis, Sultan Kenjeyev, Shangbang Long, Vlad Mnih, Harris Chan, Maxime Gazeau, Bonnie Li, Fabio Pardo, Luyu Wang, Lei Zhang, Frederic Besse, Tim Harley, Anna Mitenkova, Jane Wang, Jeff Clune, Demis Hassabis, Raia Hadsell, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 2: A large-scale foundation world model, 2024. https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model/.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023.

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

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 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 Computer Vision and Pattern Recognition Conference*, pages 6121–6132, 2025.

Pascal Roth, Jonas Frey, Cesar Cadena, and Marco Hutter. Learned perceptive forward dynamics model for safe and platform-aware robotic navigation. *arXiv preprint arXiv:2504.19322*, 2025.

David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models. *arXiv preprint arXiv:2402.09470*, 2024.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. *ArXiv*, abs/2407.08608, 2024. https://api.semanticscholar.org/CorpusID:271098045.

Joonghyuk Shin, Zhengqi Li, Richard Zhang, Jun-Yan Zhu, Jaesik Park, Eli Schechtman, and Xun Huang. Motionstream: Real-time video generation with interactive motion controls. *arXiv preprint arXiv:2511.01266*, 2025.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

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

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models, 2023.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Mingzhen Sun, Weining Wang, Gen Li, Jiawei Liu, Jiahui Sun, Wanquan Feng, Shanshan Lao, SiYu Zhou, Qian He, and Jing Liu. Ar-diffusion: Asynchronous video generation with auto-regressive diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7364–7373, 2025.

HunyuanWorld Team, Zhenwei Wang, Yuhao Liu, Junta Wu, Zixiao Gu, Haoyuan Wang, Xuhui Zuo, Tianyu Huang, Wenhuan Li, Sheng Zhang, et al. Hunyuanworld 1.0: Generating immersive, explorable, and interactive 3d worlds from words or pixels. *arXiv preprint arXiv:2507.21809*, 2025.

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

Sifan Tu, Xin Zhou, Dingkang Liang, Xingyu Jiang, Yumeng Zhang, Xiaofan Li, and Xiang Bai. The role of world models in shaping autonomous driving: A comprehensive survey. *arXiv preprint arXiv:2502.10498*, 2025.

Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 13(4):376–380, 2002.

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, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

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

Jing Wang, Fengzhuo Zhang, Xiaoli Li, Vincent YF Tan, Tianyu Pang, Chao Du, Aixin Sun, and Zhuoran Yang. Error analyses of auto-regressive video diffusion models: A unified framework. *arXiv preprint arXiv:2503.10704*, 2025b.

Thaddäus Wiedemer, Yuxuan Li, Paul Vicol, Shixiang Shane Gu, Nick Matarese, Kevin Swersky, Been Kim, Priyank Jaini, and Robert Geirhos. Video models are zero-shot learners and reasoners. *arXiv preprint arXiv:2509.20328*, 2025.

World Labs. Rtfm: A real-time frame model, 2025a. https://www.worldlabs.ai/blog/rtfm.

World Labs. Marble, 2025b. https://marble.worldlabs.ai/. Product site.

World Labs. Generating worlds, 2025c. https://www.worldlabs.ai/blog/generating-worlds. Product site.

Haoyu Wu, Diankun Wu, Tianyu He, Junliang Guo, Yang Ye, Yueqi Duan, and Jiang Bian. Geometry forcing: Marrying video diffusion and 3d representation for consistent world modeling. *arXiv preprint arXiv:2507.07982*, 2025a.

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*, 2025b.

Hongchi Xia, Entong Su, Marius Memmel, Arhan Jain, Raymond Yu, Numfor Mbiziwo-Tiapo, Ali Farhadi, Abhishek Gupta, Shenlong Wang, and Wei-Chiu Ma. Drawer: Digital reconstruction and articulation with environment realism. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21771–21782, 2025.

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

Desai Xie, Zhan Xu, Yicong Hong, Hao Tan, Difan Liu, Feng Liu, Arie Kaufman, and Yang Zhou. Progressive autoregressive video diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6322–6332, 2025.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, Song Han, and Yukang Chen. Longlive: Real-time interactive long video generation. *arXiv preprint arXiv:2509.22622*, 2025.

Deheng Ye, Fangyun Zhou, Jiacheng Lv, Jianqi Ma, Jun Zhang, Junyan Lv, Junyou Li, Minwen Deng, Mingyu Yang, Qiang Fu, et al. Yan: Foundational interactive video generation. *arXiv preprint arXiv:2508.08601*, 2025.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6613–6623, 2024b.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22963–22974, 2025.

Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T Freeman, and Jiajun Wu. Wonderworld: Interactive 3d scene generation from a single image. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5916–5926, 2025a.

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

Jiwen Yu, Yiran Qin, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Gamefactory: Creating new games with generative interactive videos. *arXiv preprint arXiv:2501.08325*, 2025c.

Liping Yuan, Jiawei Wang, Haomiao Sun, Yuchen Zhang, and Yuan Lin. Tarsier2: Advancing large vision-language models from detailed video description to comprehensive video understanding. *arXiv preprint arXiv:2501.07888*, 2025.

Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. Arf: Artistic radiance fields. In *European Conference on Computer Vision*, pages 717–733. Springer, 2022.

Lvmin Zhang and Maneesh Agrawala. Packing input frame context in next-frame prediction models for video generation. *arXiv preprint arXiv:2504.12626*, 2025.

Qihang Zhang, Shuangfei Zhai, Miguel Angel Bautista Martin, Kevin Miao, Alexander Toshev, Joshua Susskind, and Jiatao Gu. World-consistent video diffusion with explicit 3d modeling. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21685–21695, 2025a.

Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William T Freeman, and Hao Tan. Test-time training done right. *arXiv preprint arXiv:2505.23884*, 2025b.

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

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023.

Deyu Zhou, Quan Sun, Yuang Peng, Kun Yan, Runpei Dong, Duomin Wang, Zheng Ge, Nan Duan, and Xiangyu Zhang. Taming teacher forcing for masked autoregressive video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7374–7384, 2025a.

Xin Zhou, Dingkang Liang, Sifan Tu, Xiwu Chen, Yikang Ding, Dingyuan Zhang, Feiyang Tan, Hengshuang Zhao, and Xiang Bai. Hermes: A unified self-driving world model for simultaneous 3d scene understanding and generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025b.