BOXWORLD: LET WORLD ENGINES CONTROL AND VIDEO GENERATORS RENDER

Anonymous authors
Paper under double-blind review

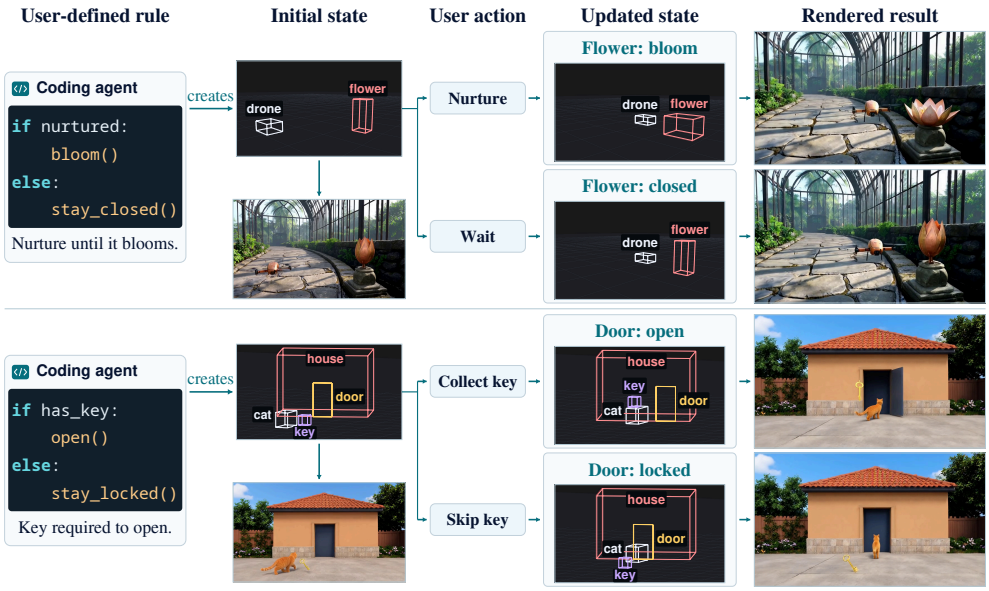


Figure 1: **BoxWorld separates interaction reasoning from generative rendering.** A coding agent creates a **Box World Engine** from user-defined rules. The engine updates object states from user actions, and the **Box-State Generative Renderer** renders them into video. Nurturing controls whether the flower blooms (top), while collecting a key controls whether the door opens (bottom). Videos are provided in the supplementary material.

ABSTRACT

We tackle the challenge of building real-time interactive video worlds with user-defined behaviors. Such worlds must track object states, apply interaction rules, and render the resulting consequences as users act. Yet existing end-to-end video world models typically learn interaction dynamics within the generator, making it difficult to specify object-level rules and reason about their consequences. In this work, we introduce `BoxWorld`, which separates interaction reasoning from generative rendering. Given user-specified rules, a coding agent creates a **Box World Engine** that determines interaction outcomes and exposes the resulting object states through a lightweight interface of layout boxes, persistent object attributes, and dynamic status variables. A **Box-State Generative Renderer** then renders this evolving state stream into video autoregressively, enabling online control over object motion, state switches, and conditional events. Rendering this state stream is non-trivial: the renderer must respond to object-level state updates in real time while remaining coherent with the generated history. To this end, we introduce a three-stage training strategy that learns object grounding from open-domain videos and state transitions from gameplay videos, both annotated automatically, and finally performs rollout distillation on self-generated histories. We further propose PairedStateBench, a paired benchmark for fine-grained interaction control. BoxWorld substantially outperforms end-to-end world models and remains competitive across multiple public benchmarks.

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

1 INTRODUCTION

Building models that can understand, simulate, and interact with the physical world has long been an ambitious goal of artificial intelligence (Ha & Schmidhuber, 2018). Recent progress in video generation (Team Wan et al., 2025; Google DeepMind, 2025; Team Seedance et al., 2026) has given this goal a concrete instantiation: *interactive video worlds*, where users can act in a generated world and experience it unfolding over time in response to their actions (Brooks et al., 2024; Bruce et al., 2024; Parker-Holder et al., 2024; Ball et al., 2025). We are interested in real-time interactive video worlds with user-defined behaviors, where users also specify how the world responds to their actions. For example, a user may specify that a flower blooms only when nurtured (Figure 1 (top)), or that a door opens only if the cat has collected the key (Figure 1 (bottom)). Realizing such behaviors requires the world to track object states, apply interaction rules, and render the resulting consequences as users act.

A prevailing approach learns interactive video worlds end to end, mapping visual history and user actions directly to future frames with a video generator that generalizes across diverse scenes (Bruce et al., 2024; Robbyant Team et al., 2026) (Figure 2(a)). However, the generator must then handle all three tasks implicitly, reasoning about whether rules are satisfied and what consequences should follow, which makes object-level rules and their consequences difficult to specify and control. A complementary direction instead conditions video generation on explicit states, ranging from trajectories (Li et al., 2025a) and layouts (Feng et al., 2025b) to physical priors (Zhang et al., 2025), 3D caches (Ren et al., 2025), and full simulators (Xie et al., 2025) (Figure 2(b)). These states provide finer control, but grow heavier and more domain-specific as control becomes finer, making them expensive to construct and difficult to obtain from videos at scale.

Taken together, these lines of work reveal a trade-off between generalization and fine-grained control in interactive video worlds. This motivates separating interaction reasoning from generative rendering: interaction rules are applied explicitly outside the generator, while the generator retains its broad visual priors and focuses on faithfully rendering the resulting object states. The two sides are connected by an explicit-but-minimal **Object-Centric State Interface**, which exposes only the variables needed for object-level interaction while remaining simple to specify, update, and apply across diverse scenes. Crucially, these state variables should be identifiable from videos with off-the-shelf perception models, so that learning the renderer is not restricted to predefined simulators or scarce action-paired interaction data.

Following this design, we present `BoxWorld`, a framework for real-time interactive video worlds with user-defined behaviors. Given user-specified rules, a coding agent creates a **Box World Engine**, which applies these rules to user actions, determines the resulting interaction outcomes, and updates object states accordingly. The engine exposes these states through the **Object-Centric State Interface**, composed of layout boxes, persistent object attributes, and dynamic status variables, leaving texture, lighting, style, and other visual details to generative rendering. A **Box-State Generative Renderer** then renders this evolving state stream into video autoregressively, enabling online control over object motion, state switches, and conditional events (Figure 2(c)).

However, rendering this state stream is non-trivial: the renderer must respond to object-level state updates in real time while remaining coherent with the generated history. First, states arrive online, so generation must be few-step and causal, whereas existing layout-to-video methods rely on multi-step bidirectional generation over the entire layout sequence (Li et al., 2025a; Lian et al., 2024; Tian et al., 2024; Feng et al., 2025b). Second, an update makes part of the generated history outdated, so the renderer must distinguish what to preserve from what to change. Third, the renderer must synthesize the new state of the affected object, such as a door swinging open. Existing video editing methods achieve similar localized changes, but rely on precise masks, costly inversion, or per-video optimization (Jiang et al., 2025; Mou et al., 2024; Ku et al., 2024; Gao et al., 2026a), making them unsuitable for online rendering from coarse boxes.

To address these challenges, we build the **Box-State Generative Renderer** on an autoregressive video generator (Yang et al., 2025a) and train it in three stages, each with a training scheme matched to its role (Sec. 3.3). The first stage, *Masked Layout Grounding*, learns frame-wise object grounding on open-domain videos (Tan et al., 2024; Chen et al., 2025b) with Diffusion Forcing (Chen et al., 2024), whose independent per-frame noise levels make grounding hold regardless of the noise level of the temporal context. The second stage, *State Dynamics Adaptation*, learns state transitions on

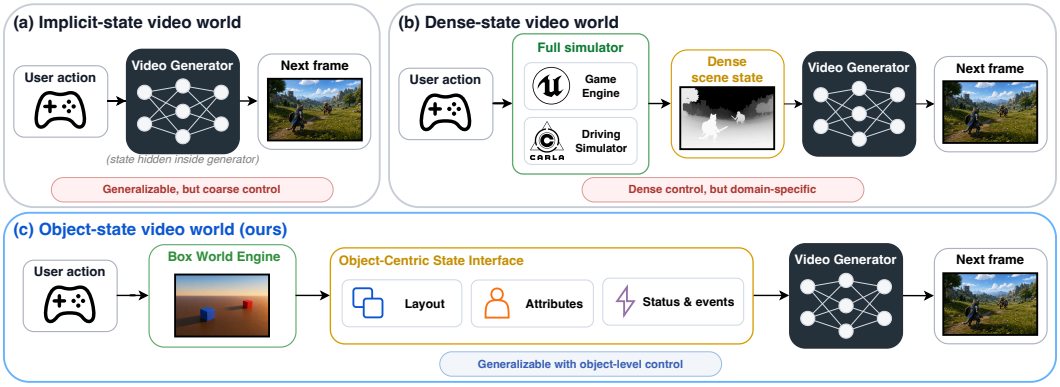


Figure 2: **Interfaces for interactive video worlds.** (a) Implicit-state approaches benefit from broad visual generalization, but provide only coarse control. (b) Dense-state approaches expose finer control through simulators, but such interfaces are often domain-specific and costly to construct. (c) Our `BoxWorld` exposes an explicit-but-minimal **Object-Centric State Interface** for object-level control, while retaining the generalization ability of video generators.

gameplay videos with teacher forcing, whose clean history depicts the preceding state, so that the renderer learns to change the affected object while continuing the rest of the scene. The third stage, *Rollout Distillation*, distills the renderer on its own rollouts with Self-Forcing-style distribution matching (Huang et al., 2025b), restoring few-step generation while adapting it to the self-generated histories encountered at inference. The state annotations are extracted automatically by off-the-shelf perception models (Liu et al., 2025; Zhang et al., 2024b), requiring no action-paired interaction logs (Sec. 3.4).

We further introduce PairedStateBench, which evaluates user-defined behaviors through paired branches with distinct intended outcomes, measuring whether generated videos realize specified consequences and suppress unspecified ones. `BoxWorld` substantially outperforms end-to-end world models on PairedStateBench, and remains competitive in short- and long-horizon video quality on VBench and VBench-Long (Huang et al., 2024), and layout-to-video generation on YouTube-VIS (Yang et al., 2021), while running in real time. In summary, our key contributions are as follows:

- We propose `BoxWorld`, a framework for real-time interactive video worlds with user-defined behaviors, in which a **Box World Engine** created by a coding agent determines interaction outcomes and a **Box-State Generative Renderer** realizes them through the **Object-Centric State Interface**.
- We introduce a three-stage training strategy with distinct roles: *Masked Layout Grounding* and *State Dynamics Adaptation* learn object grounding from open-domain videos and state transitions from gameplay videos, both annotated automatically, and *Rollout Distillation* restores few-step generation on the renderer's own rollouts.
- We propose PairedStateBench for fine-grained interaction control, on which `BoxWorld` substantially outperforms end-to-end world models while remaining competitive on public benchmarks for video quality and layout-to-video generation.

2 Related Work

Interactive Video World Models. Early controllable world models showed that playable dynamics can be learned directly from video (Menapace et al., 2021) or prompt-conditioned game loops (Menapace et al., 2024). This line then scaled to tokenized or latent world models (Wu et al., 2024; Bruce et al., 2024; Bar et al., 2025) and larger video corpora (Parker-Holder et al., 2024; Ball et al., 2025), and more recently to interactive simulators built from powerful video generators with action (Valevski et al., 2025; Che et al., 2025), camera (Huang et al., 2025a), and navigation control (Yu et al., 2025; He et al., 2026; Guo et al., 2025). Recent extensions further target open environments, character-centered interaction, and longer-horizon memory (Sun et al., 2025; He et al., 2025; Robbyant Team et al., 2026; Xiao et al., 2025; Zhu et al., 2025). In parallel, stronger backbones (HaCohen et al., 2024; Team Wan

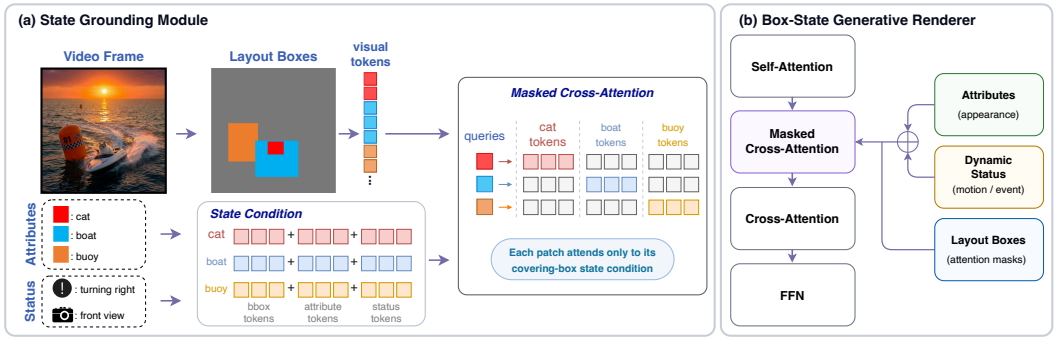


Figure 3: **Box-State Generative Renderer.** (a) The State Grounding Module encodes each object's box, attribute, and dynamic status into grounding tokens, and its masked cross-attention lets each visual token attend only to the objects whose boxes cover it. (b) The module is inserted between self-attention and cross-attention in each block of the renderer.

et al., 2025) together with causalization and distillation (Yin et al., 2025; Huang et al., 2025b; Yang et al., 2025a) are beginning to enable real-time interactive systems (Shin et al., 2026). These works show strong visual interaction, but scene dynamics and intervention logic still largely reside in the generator (Valevski et al., 2025; Huang et al., 2025a; Yang et al., 2025a), making object-level rules and their consequences difficult to specify and control.

State-Conditioned Generation. Another line exposes explicit state and conditions image or video generation on it. Early grounded generation uses layouts and regions for object-level control (Li et al., 2023; Dou et al., 2024; Cheng et al., 2023), while ControlNet-style methods extend explicit conditioning to dense signals such as edges, depth, segmentation, and pose (Zhang et al., 2023; Zhao et al., 2023; Zavadski et al., 2025). Later methods extend this idea to planning (Lin et al., 2024), trajectories and tracklets (Yin et al., 2023; Ma et al., 2024; Li et al., 2025a; 2026; Zhang et al., 2026), camera and motion control (Wang et al., 2024; Feng et al., 2025a; Chen et al., 2023; Zhang et al., 2024a; Wang et al., 2023), and more compositional object guidance (Li et al., 2025b; Feng et al., 2025b; Mei et al., 2024; Zhou et al., 2025; Wang et al., 2025a). Recent physics-aware and simulator-grounded variants further improve consistency by injecting physical priors or planning (Zhang et al., 2025; Ji et al., 2025; Yang et al., 2025b; Wang et al., 2025b), or by conditioning generation on richer state such as simulators, depth, geometry, and 3D caches (Xie et al., 2025; Cai et al., 2024; Ren et al., 2025; Kong et al., 2025). Together, these methods show that explicit conditioning improves controllability, but they typically generate a complete clip offline, using multi-step bidirectional denoising given the full condition sequence in advance (Li et al., 2025a; Feng et al., 2025b; Wang et al., 2024), or depend on heavier state and supervision (Xie et al., 2025; Cai et al., 2024; Ren et al., 2025; Kong et al., 2025). In contrast, BoxWorld renders in real time, consuming object states as they arrive with few-step causal generation, and realizes state changes during the rollout while remaining coherent with the generated history.

3 METHOD

`BoxWorld` assigns interaction reasoning to a **Box World Engine** and generative rendering to a **Box-State Generative Renderer**, which communicate through the **Object-Centric State Interface**. We describe the engine and interface in Sec. 3.1, the renderer in Sec. 3.2, and how the renderer is trained in three stages (Sec. 3.3) on video-derived annotations (Sec. 3.4).

3.1 BOX WORLD ENGINE AND OBJECT-CENTRIC STATE INTERFACE

At time step t, the Object-Centric State Interface describes the scene as $\mathcal{S}_t = \{(\mathbf{b}_{t,i}, \mathbf{c}_i, \mathbf{d}_{t,i})\}_{i=1}^{N_t}$ over N_t active objects. For object i, the layout box $\mathbf{b}_{t,i}$ gives its 2D location in frame coordinates, the persistent attribute $\mathbf{c}_i$ describes its identity and appearance, and the dynamic status variable $\mathbf{d}_{t,i}$ describes its current status.

Given user-specified rules, a coding agent implements a Box World Engine from a shared engine template before interaction (Appendix B). During interaction, the engine maintains its internal scene

variables and updates them with each user action. It applies the rules to decide whether and when each object changes state, and projects the resulting object states into $\mathcal{S}_t$, which is published to a state buffer read asynchronously by the renderer.

Since $\mathcal{S}_t$ can be recovered from videos with off-the-shelf perception models (Sec. 3.4), the renderer can be trained without engines or action-paired interaction logs, and at inference time it can be driven by any engine that publishes the same interface (Sec. 4.2).

3.2 Box-State Generative Renderer

Frame-wise state grounding. The Box-State Generative Renderer generates the latent video autoregressively, modeling $p_\theta(\mathbf{x}_t \mid \mathbf{x}_{<t}, \mathcal{S}_t)$, where $\mathbf{x}_t$ is the t-th latent frame, $\mathbf{x}_{<t}$ is the generated history, and $\mathcal{S}_t$ is the current state. Since $\mathcal{S}_t$ changes over time as objects move and change status, each frame should be grounded only with its own state. We therefore inject states frame by frame, which matches this constraint and avoids costly cross-frame attention. Specifically, we insert into each backbone block a State Grounding Module implemented as frame-wise masked cross-attention (Figure 3). At frame t, the states of all objects are encoded into grounding tokens $\mathbf{E}_t \in \mathbb{R}^{(N_t L)\times D}$ with L tokens per object, and their boxes define a mask $\mathbf{M}_t \in \{0,1\}^{HW\times(N_t L)}$ that lets each visual token attend only to the objects covering it. We reshape the flattened visual tokens $\mathbf{h} \in \mathbb{R}^{B\times(THW)\times D}$ into per-frame tokens $\tilde{\mathbf{h}} \in \mathbb{R}^{(BT)\times(HW)\times D}$ and apply masked cross-attention to each frame:

$$\mathbf{Y}_t = \mathrm{Softmax}\left(\frac{(\tilde{\mathbf{h}}_t\mathbf{W}_Q)(\mathbf{E}_t\mathbf{W}_K)^\top}{\sqrt{D}} + \Phi(\mathbf{M}_t)\right)\mathbf{E}_t\mathbf{W}_V, \tag{1}$$

where $\Phi(\mathbf{M}_t)$ is 0 for allowed entries and $-\infty$ otherwise. The per-frame outputs $\mathbf{Y}_t$ are reshaped back into the video-token sequence $\bar{\mathbf{Y}}$ and injected through a gated residual:

$$\mathbf{h}' = \mathbf{h} + \lambda \tanh(\gamma)\bar{\mathbf{Y}}, \tag{2}$$

where λ is a control scale and γ is a learnable gate initialized to zero. Temporal coherence across frames is left to the causal self-attention of the backbone.

Asynchronous streaming rendering. At inference, the Box World Engine and the renderer run as asynchronous processes connected by the state buffer. The engine continuously receives user actions, updates its internal scene variables according to the rules, and publishes the resulting state $\mathcal{S}_t$ through the interface. Meanwhile, the renderer reads the latest state from the buffer whenever it generates new frames and renders them autoregressively from the generated history. A state update is therefore reflected in the next generated frames, while the engine keeps simulating the scene in parallel.

3.3 Three-Stage Training

We initialize the renderer from LongLive (Yang et al., 2025a), a few-step autoregressive video generator distilled from Wan2.1 (Team Wan et al., 2025), and train it in three stages: the first two learn object grounding and state transitions, respectively, and the third restores few-step generation by distilling on its own rollouts. In each stage, θ denotes the trainable parameters of that stage.

Stage 1: Masked Layout Grounding. This stage learns object grounding, binding each object to its layout box and persistent attribute, and trains on open-domain videos (Tan et al., 2024; Chen et al., 2025b) whose diversity supports grounding across scenes. We freeze the backbone and train only the State Grounding Module. To bind every frame to its own state, we follow Diffusion Forcing (Chen et al., 2024), which assigns independent noise levels to the latent frames, so the module learns grounding regardless of the noise level of its temporal context. The objective is

$$\mathcal{L}_{\mathrm{S1}}(\theta) = \mathbb{E}_{\mathbf{x}_0,t,\boldsymbol{\tau}}\left[w_{\mathrm{S1}}(\tau_t)\left\|\mathbf{v}_\theta\left(\mathbf{x}_{\tau_t,t}, \tau_t, \mathbf{x}_{\boldsymbol{\tau}_{<t},<t}, \mathcal{S}_t\right) - \mathbf{u}_{\tau_t,t}\right\|^2\right], \tag{3}$$

where each frame is noised independently with levels $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_T)$, so the history $\mathbf{x}_{\boldsymbol{\tau}_{<t},<t}$ is itself noisy at varying levels, and $\mathbf{v}_\theta$ predicts the target velocity $\mathbf{u}_{\tau_t,t}$. Since boxes mainly constrain coarse layout, which is determined early in denoising, the weighting w_{S1} (Duan et al., 2026) emphasizes high noise levels, and for efficiency the module is active only at these levels during inference.

Stage 2: State Dynamics Adaptation. This stage learns state transitions, which open-domain videos rarely contain, so we train on gameplay videos. We freeze the Stage 1 renderer and train LoRA adapters (Hu et al., 2022) in its linear layers, as LongLive (Yang et al., 2025a) shows that LoRA adaptation of a distilled streaming generator approaches the quality of full fine-tuning. To learn how a status change alters the video, we use teacher forcing, in which the renderer denoises the current frame given the clean ground-truth prefix. When a status changes, the prefix depicts the preceding status and the target depicts the new one, so the adapters learn to change the affected object while continuing the rest of the scene. The objective is

$$\mathcal{L}_{\mathrm{S2}}(\theta) = \mathbb{E}_{\mathbf{x}_0,t,\tau}\left[\left\|\mathbf{v}_\theta\left(\mathbf{x}_{\tau,t}, \tau, \mathbf{x}_{0,<t}, \mathcal{S}_t\right) - \mathbf{u}_{\tau,t}\right\|^2\right], \tag{4}$$

where the history $\mathbf{x}_{0,<t}$ is the clean ground-truth prefix and $\mathbf{x}_{\tau,t}$ is the noisy current frame.

Stage 3: Rollout Distillation. The denoising objectives of the first two stages shift the renderer away from its distilled few-step distribution, and teacher forcing trains it on clean prefixes, whereas inference conditions on self-generated ones. This stage therefore restores few-step generation by distilling the renderer on its own rollouts. We freeze the Stage 2 renderer and train an additional LoRA adapter θ with Self-Forcing-style distribution matching distillation (DMD) (Yin et al., 2024; Huang et al., 2025b), using the Stage 2 renderer as the teacher and state sequences from both open-domain and gameplay annotations. The student generates few-step autoregressive rollouts $\hat{\mathbf{x}}$ and is updated with

$$\nabla_\theta \mathcal{L}_{\mathrm{S3}} = \mathbb{E}_{\tau,\boldsymbol{\epsilon}}\left[w(\tau)\left(s_{\mathrm{fake}}(\hat{\mathbf{x}}_\tau, \tau, \mathcal{S}_{1:T}) - s_{\mathrm{real}}(\hat{\mathbf{x}}_\tau, \tau, \mathcal{S}_{1:T})\right)^\top \frac{\partial \hat{\mathbf{x}}}{\partial \theta}\right], \tag{5}$$

where $\hat{\mathbf{x}}_\tau$ is the rollout re-noised to level τ, s_{real} and s_{fake} are the scores of the teacher and of a critic initialized from it, and $w(\tau)$ is the normalizer following DMD. Sec. 4.4 evaluates its effect on long-horizon generation.

For efficient inference, we merge the LoRA adapters of LongLive and of Stages 2 and 3 into the backbone weights (Hu et al., 2022), so that the adapters add no runtime cost.

3.4 Video-to-Structure Annotation

We recover $\mathcal{S}_t$ from open-domain and gameplay videos with off-the-shelf perception models: LLaVA-NeXT-Video (Zhang et al., 2024b) for clip descriptions, Grounding DINO (Liu et al., 2025) for initial layout boxes, SAM 2 (Ravi et al., 2025) for tracking, and object-level captioning for persistent attributes. Open-domain videos cover diverse objects and scenes but contain few status changes, whereas gameplay videos contain frequent actions and triggered events. For both sources, we crop each tracked object across frames and describe its current status and action with emphasis on status changes, so that the per-frame descriptions form $\mathbf{d}_{t,i}$ and expose status transitions. Appendix A provides details and examples.

4 Experiments

4.1 Implementation Details

We train Stage 1 on filtered open-domain videos from VideoGen-1M (Tan et al., 2024) and 4DNeX (Chen et al., 2025b), Stage 2 on 4,000 short gameplay clips collected online, and Stage 3 on state sequences from both sources. The three stages run for 20K, 10K, and 400 iterations, with rank-16 LoRA adapters in Stage 2 and a rank-64 adapter in Stage 3. At inference, the renderer uses four denoising steps at 832×480, and the State Grounding Module is active only at noise levels above 0.8. The Box World Engines in our experiments are created by Kimi K2.5 (Kimi Team et al., 2026) from user-specified rules (Appendix C.1).

4.2 Fine-Grained Interaction Control

We evaluate fine-grained interaction control on PairedStateBench, whose cases range from directional motion and object handling to credential-dependent doors, charging, and multi-stage interactions. Each case specifies the intended state evolution, event conditions, and temporal windows, and paired

Table 1: **Results on PairedStateBench.** †Camera-control version, the only publicly released LingBot-v2 variant. Oracle variants additionally receive the interaction consequences as text.

Method	FPS↑	Trigger↑	Suppress↑	Paired-CF↑	State-Fid↑	OnTime↑	Task↑
Overall							
LingBot-v1	0.05	53.5	11.1	16.2	43.1	44.5	25.3
LingBot-v1 (Oracle)	0.05	45.5	25.4	16.2	42.3	27.7	18.4
LingBot-v2†	13.3	21.3	63.5	10.8	46.1	14.3	11.5
LingBot-v2† (Oracle)	13.3	19.0	69.8	16.2	57.1	17.5	11.5
LongLive	**20.7**	10.7	50.8	13.5	36.2	5.7	8.0
BoxWorld (Ours)	**20.7**	**92.2**	**90.5**	**97.3**	**91.6**	**91.2**	**83.9**
Easy							
LingBot-v1	0.05	75.0	–	60.0	38.3	75.0	47.8
LingBot-v1 (Oracle)	0.05	50.0	–	30.0	5.3	50.0	26.1
LingBot-v2†	13.3	25.0	–	20.0	6.9	25.0	30.4
LingBot-v2† (Oracle)	13.3	50.0	–	40.0	50.0	50.0	39.1
LongLive	**20.7**	0.0	–	30.0	0.0	0.0	26.1
BoxWorld (Ours)	**20.7**	**100.0**	–	**100.0**	**98.4**	**100.0**	**82.6**
Medium							
LingBot-v1	0.05	55.4	20.0	0.0	51.9	43.1	32.1
LingBot-v1 (Oracle)	0.05	50.4	20.0	0.0	53.4	25.4	25.0
LingBot-v2†	13.3	26.1	**100.0**	0.0	53.6	10.9	7.1
LingBot-v2† (Oracle)	13.3	15.2	**100.0**	0.0	59.9	12.0	3.6
LongLive	**20.7**	13.0	**100.0**	0.0	52.6	6.5	3.6
BoxWorld (Ours)	**20.7**	**85.9**	**100.0**	**100.0**	**90.2**	**85.9**	**89.3**
Hard							
LingBot-v1	0.05	47.8	8.3	0.0	36.7	40.6	5.6
LingBot-v1 (Oracle)	0.05	39.9	27.1	15.8	37.7	26.1	8.3
LingBot-v2†	13.3	15.9	52.1	10.5	44.6	15.9	2.8
LingBot-v2† (Oracle)	13.3	17.4	60.4	10.5	55.7	17.4	0.0
LongLive	**20.7**	10.1	35.4	10.5	27.5	5.8	0.0
BoxWorld (Ours)	**20.7**	**97.1**	**87.5**	**94.7**	**91.9**	**94.9**	**80.6**

cases share the initial scene but differ in the intended outcome. We compare against the end-to-end world models LingBot-v1 (Robbyant Team et al., 2026) and LingBot-v2 (Gao et al., 2026b), each with an oracle variant that additionally receives the interaction consequences as text, and against LongLive (Yang et al., 2025a), the backbone of our renderer. The metrics measure whether required events appear (Trigger), forbidden consequences remain absent (Suppress), both branches of a pair succeed (Paired-CF), state windows and event onsets match the specification (State-Fid, OnTime), and complete multi-stage tasks succeed (Task). Appendix C.2 provides the full benchmark details.

Quantitative Results. As shown in Table 1, BoxWorld realizes both the occurrence and the suppression of specified consequences, reaching 92.2% Trigger and 90.5% Suppress, and satisfies both branches in **97.3%** of pairs, whereas no baseline exceeds **16.2%** Paired-CF. The end-to-end models tend to produce the likely event regardless of the branch condition (Figure 4), and supplying the resolved outcome as text does not close this gap: both oracle variants remain at 16.2% Paired-CF. LongLive reaches only 13.5%, indicating that the specified outcomes are realized through state grounding. BoxWorld also completes **83.9%** of the full multi-stage tasks, compared with at most **25.3%** for the baselines, and the Task gap widens with difficulty: on hard cases, BoxWorld retains 94.7% Paired-CF and **80.6%** Task, whereas the baselines reach at most 15.8% and **8.3%**, respectively. These results show that fine-grained interaction consequences, which end-to-end models struggle to infer from visual history, are straightforward for an explicit engine to determine, and that the Box-State Generative Renderer faithfully realizes them once they are expressed as object-level states.

Qualitative Results. Figure 4 compares branches that differ in whether the precondition is met, such as nurturing the flower or collecting the key. BoxWorld renders the event only once the precondition holds, whereas LingBot-v1 renders it before the precondition is met, even with oracle text. Figure 5 further shows rollouts from six engines, each rendered in two visual worlds. The same interaction,

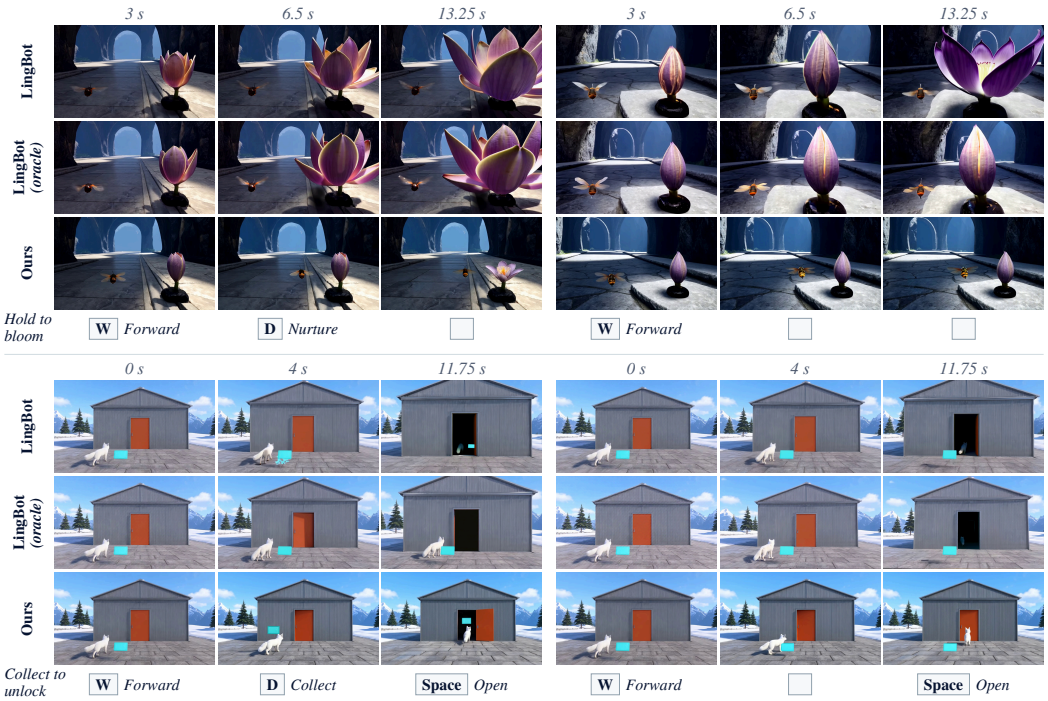


Figure 4: **Qualitative comparison on PairedStateBench.** Each case compares two branches from the same scene that differ in whether the precondition is met: nurturing the flower before it blooms (top) and collecting the key before opening the door (bottom). BoxWorld renders the event only once the precondition is met, whereas LingBot-v1 renders it without the precondition, and its oracle variant still opens the door without the key and blooms the flower before nurturing. More cases are in Appendix C.2 and the supplementary material.

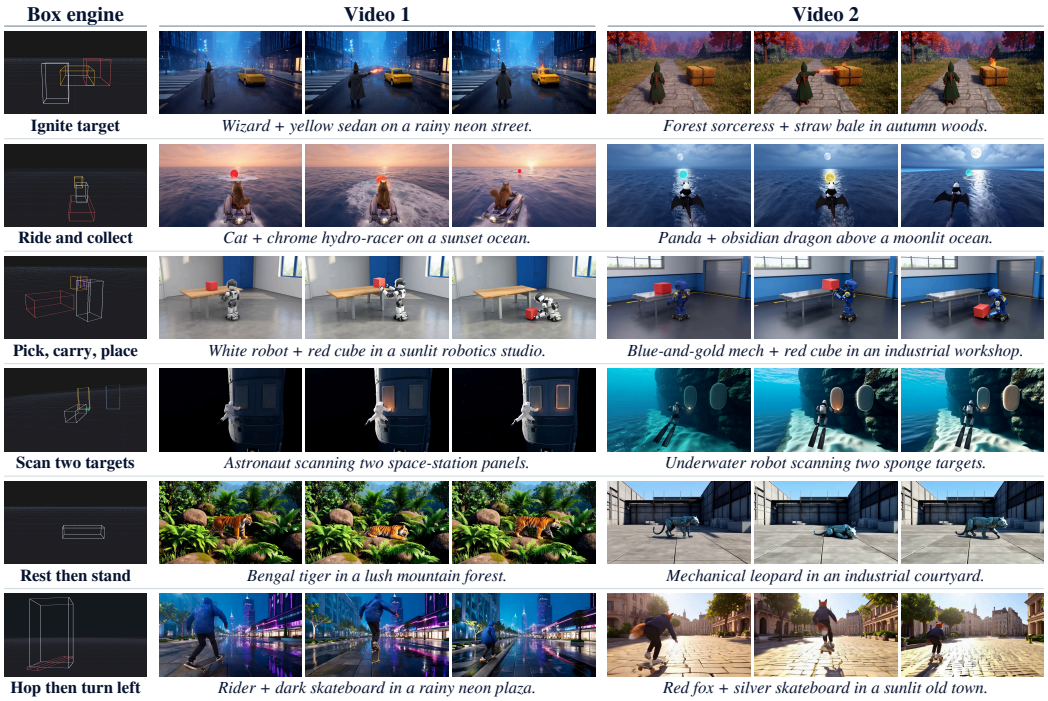


Figure 5: **Qualitative results of BoxWorld.** Each row shows a Box World Engine (left) and two videos rendered from the same state stream with different object attributes and scene descriptions. Videos are provided in the supplementary material.

such as picking up, carrying, and placing an object, is realized consistently across scenes and styles, indicating that one engine can drive diverse visual worlds through the Object-Centric State Interface.

4.3 Public Benchmarks

Video Quality. We evaluate how much of the pretrained visual prior the renderer retains on VBench and 30-second VBench-Long (Huang et al., 2024). As shown in Table 3 (Appendix C.3), BoxWorld achieves video quality comparable to LongLive at the same 20.7 FPS while improving the Semantic score from 76.47 to 79.54 on VBench and from 75.82 to 77.67 on VBench-Long, suggesting that object-level states strengthen compositional control.

Layout-to-Video Generation. Following BlobGEN-Vid (Feng et al., 2025b), we test on YouTube-VIS 2021, where ground-truth tracks provide layout boxes and per-track descriptions provide object attributes. As shown in Table 4 (Appendix C.4), BoxWorld is the only real-time method, yet it achieves the lowest FVD (62), the highest regional text and image alignment (rCLIP$_t$ 0.291, rCLIP$_i$ 0.837), and the highest regional cross-frame consistency (rCFC 0.965) among all compared methods. Its mIOU of 0.572 is second only to BlobGEN-Vid (0.598), whose blob representation offers finer spatial support than boxes.

4.4 Ablation Studies

Training Stages. As shown in Table 2, the Total score on VBench and VBench-Long increases with each stage, and Stage 3 brings the largest gain on VBench-Long, from 81.82 to 83.00, showing that rollout distillation benefits long-horizon generation.On PairedStateBench, Stage 2 raises Suppress from 60.3% to 85.7% and Task from 49.4% to 64.4%, indicating that gameplay adaptation teaches the renderer to change an object only when its status changes. Stage 3 further raises Paired-CF from 67.6% to 97.3% and Task to 83.9%, indicating that rollout distillation keeps state changes faithful over self-generated histories. Appendix C.5 reports all metrics and further compares Stage 2 adapter variants.

Table 2: **Ablations of training stages and the state interface.** Full tables are in Appendix C.

VBench		
Variant	Total$^{\uparrow}$	Long Total$^{\uparrow}$
Stage 1	83.30	81.72
Stage 2	83.57	81.82
Stage 3	**83.76**	**83.00**
YouTube-VIS		
Variant	mIOU$^{\uparrow}$	FVD$^{\downarrow}$
Stage 3	**0.572**	**62**
w/o spatial mask	0.291	79
w/o object attributes	0.323	68
PairedStateBench		
Variant	Suppress$^{\uparrow}$	Task$^{\uparrow}$
Stage 1	60.3	49.4
Stage 2	85.7	64.4
Stage 3	**90.5**	**83.9**
Variant	Trigger$^{\uparrow}$	OnTime$^{\uparrow}$
Stage 3	**92.2**	**91.2**
w/o dynamic status	42.5	16.1

State Interface. On YouTube-VIS, removing the spatial mask lets every visual token attend to all objects and reduces mIOU from 0.572 to 0.291, which supports routing each object's state to its own region, and replacing all object attributes with the generic word "object" reduces mIOU to 0.323, showing that attributes identify objects beyond their box geometry. On PairedStateBench, replacing dynamic status with static descriptions lowers Trigger from 92.2% to 42.5% and OnTime from 91.2% to 16.1%, since dynamic status specifies whether and when each object changes.

5 Conclusion

We presented `BoxWorld`, which separates interaction reasoning from generative rendering for real-time interactive video worlds. A Box World Engine, created by a coding agent from user-specified rules, determines interaction outcomes and exposes them through the Object-Centric State Interface of layout boxes, persistent attributes, and dynamic status variables. A Box-State Generative Renderer, trained in three stages on automatically annotated videos, then renders the resulting state stream into video in real time. On PairedStateBench, BoxWorld realizes fine-grained interaction consequences that end-to-end world models struggle to produce, while retaining video quality comparable to its backbone. These results suggest a practical path for controllable interactive video worlds: let a world engine decide what changes, and let a video generator render how those changes appear.

AI USE STATEMENT

The authors used large language models (LLMs) in three ways. First, as components of the method, vision-language models annotate the training videos (Sec. 3.4), and Kimi K2.5 serves as the coding agent that creates the Box World Engines from user-specified rules (Sec. 4.1). Second, for evaluation, Kimi K2.5 and Qwen3.8-27B judge the PairedStateBench videos, and their disagreements are resolved by manual review (Appendix C.2). Third, LLMs were used only for minor grammar checking and language polishing. Apart from their roles as the method and evaluation components described above, LLMs did not contribute to the research design, methodology, experiments, or analysis, and all technical work and insights are entirely our own. Theoretical proofs are not applicable to this work. We reviewed all AI-assisted outputs, including the generated engines, the evaluation judgments, and the edited text. We take full responsibility for the content of the paper.

ETHICS STATEMENT

This work follows the ICLR Code of Ethics. We use only publicly available data with proper attribution and respect their original licensing terms. Our computational experiments pose minimal societal risk, though we acknowledge that machine learning methods can be misused if applied inappropriately. All results are reported honestly without selective presentation. We considered potential biases in our experimental setup and took reasonable steps to ensure fair evaluation across different conditions. The authors have no financial or personal conflicts of interest related to this research.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, Sec. 4.1 and Appendices A–C provide the implementation details, annotation pipeline, engine specification, and evaluation protocols. The open-domain training videos and all evaluation benchmarks are publicly available, and we will release the code, model checkpoints, engine templates, and PairedStateBench to the research community.

REFERENCES

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. Google DeepMind blog, 2025. URL https://deepmind.google/blog/genie-3-a-new-frontier-for-world-models/.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation World Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 15791–15801. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.01472. URL https://openaccess.thecvf.com/content/CVPR2025/html/Bar_Navigation_World_Models_CVPR_2025_paper.html.

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. OpenAI technical report, 2024. URL https://openai.com/index/video-generation-models-as-world-simulators/.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C.Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative Interactive Environments. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623. PMLR, 2024. URL https://proceedings.mlr.press/v235/bruce24a.html.

Shengqu Cai, Duygu Ceylan, Matheus Gadelha, Chun-Hao Paul Huang, Tuanfeng Yang Wang, and Gordon Wetzstein. Generative Rendering: Controllable 4D-Guided Video Generation with 2D Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 17-21, 2024*, pp. 7611–7620. IEEE, 2024. doi: 10.1109/CVPR52733.2024.00727. URL https://doi.org/10.1109/CVPR52733.2024.00727.

Haoxuan Che, Xuanhua He, Quande Liu, Cheng Jin, and Hao Chen. GameGen-X: Interactive Open-world Game Video Generation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, pp. 37546–37593. OpenReview.net, 2025. URL https://openreview.net/forum?id=8VG8tpPZhe.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion Forcing: Next-token Prediction Meets Full-Sequence Diffusion. In *Advances in Neural Information Processing Systems*, volume 37, pp. 24081–24125. Curran Associates, Inc., 2024. doi: 10.52202/079017-0759. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/2aee1c4159e48407d68fe16ae8e6e49e-Paper-Conference.pdf.

Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, Weiming Xiong, Wei Wang, Nuo Pang, Kang Kang, Zhiheng Xu, Yuzhe Jin, Yupeng Liang, Yubing Song, Peng Zhao, Boyuan Xu, Di Qiu, Debang Li, Zhengcong Fei, Yang Li, and Yahui Zhou. SkyReels-V2: Infinite-length Film Generative Model. *arXiv preprint arXiv:2504.13074*, 2025a. doi: 10.48550/arXiv.2504.13074. URL https://arxiv.org/abs/2504.13074.

Weifeng Chen, Yatai Ji, Jie Wu, Hefeng Wu, Pan Xie, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-A-Video: Controllable Text-to-Video Diffusion Models with Motion Prior and Reward Feedback Learning. *arXiv preprint arXiv:2305.13840*, 2023. doi: 10.48550/arXiv.2305.13840. URL https://arxiv.org/abs/2305.13840.

Zhaoxi Chen, Tianqi Liu, Long Zhuo, Jiawei Ren, Zeng Tao, He Zhu, Fangzhou Hong, Liang Pan, and Ziwei Liu. 4DNeX: Feed-Forward 4D Generative Modeling Made Easy. *arXiv preprint arXiv:2508.13154*, 2025b. doi: 10.48550/arXiv.2508.13154. URL https://arxiv.org/abs/2508.13154.

Jiaxin Cheng, Xiao Liang, Xingjian Shi, Tong He, Tianjun Xiao, and Mu Li. LayoutDiffuse: Adapting Foundational Diffusion Models for Layout-to-Image Generation. *arXiv preprint arXiv:2302.08908*, 2023. doi: 10.48550/arXiv.2302.08908. URL https://arxiv.org/abs/2302.08908.

Haoge Deng, Ting Pan, Haiwen Diao, Zhengxiong Luo, Yufeng Cui, Huchuan Lu, Shiguang Shan, Yonggang Qi, and Xinlong Wang. Autoregressive Video Generation without Vector Quantization. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, pp. 44730–44745. OpenReview.net, 2025. URL https://openreview.net/forum?id=JE9tCwe3lp.

Huanzhang Dou, Ruixiang Li, Wei Su, and Xi Li. GVDIFF: Grounded Text-to-Video Generation with Diffusion Models. *arXiv preprint arXiv:2407.01921*, 2024. doi: 10.48550/arXiv.2407.01921. URL https://arxiv.org/abs/2407.01921.

Zhongjie Duan, Hong Zhang, and Yingda Chen. Diffusion Templates: A Unified Plugin Framework for Controllable Diffusion. *arXiv preprint arXiv:2604.24351*, 2026. doi: 10.48550/arXiv.2604.24351. URL https://arxiv.org/abs/2604.24351.

Wanquan Feng, Jiawei Liu, Pengqi Tu, Tianhao Qi, Mingzhen Sun, Tianxiang Ma, Songtao Zhao, Siyu Zhou, and Qian He. I2VControl-Camera: Precise Video Camera Control with Adjustable Motion Strength. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, pp. 49560–49572. OpenReview.net, 2025a. URL https://openreview.net/forum?id=AcAD4VEgCX.

Weixi Feng, Chao Liu, Sifei Liu, William Yang Wang, Arash Vahdat, and Weili Nie. BlobGEN-Vid: Compositional Text-to-Video Generation with Blob Video Representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 12989–12998. Computer Vision Foundation / IEEE, 2025b. doi: 10.1109/CVPR52734.2025.01212. URL https://openaccess.thecvf.com/content/CVPR2025/html/Feng_BlobGEN-Vid_Compositional_Text-to-Video_Generation_with_Blob_Video_Representations_CVPR_2025_paper.html.

Chenjian Gao, Lihe Ding, Xin Cai, Zhanpeng Huang, Zibin Wang, and Tianfan Xue. Controllable First-Frame-Guided Video Editing via Mask-Aware LoRA Fine-Tuning. In *International Conference on Learning Representations*, pp. 61741–61765, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/64e2449d74f84e5b1a5c96ba7b3d308e-Abstract-Conference.html.

Zelin Gao, Qiuyu Wang, Jiapeng Zhu, Jingye Chen, Zichen Liu, Qingyan Bai, Jiahao Wang, Yufeng Yuan, Hanlin Wang, Yichong Lu, Ka Leong Cheng, Haojie Zhang, Jian Gao, Tianrui Feng, Yuzheng Liu, Yao Yao, Yinghao Xu, Xing Zhu, Yujun Shen, and Hao Ouyang. Infinite Worlds with Versatile Interactions. *arXiv preprint arXiv:2607.07534*, 2026b. doi: 10.48550/arXiv.2607.07534. URL https://arxiv.org/abs/2607.07534.

Google DeepMind. Veo 3 Model Card. Google DeepMind, 2025. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Veo-3-Model-Card.pdf. First published May 23, 2025; revised January 13, 2026.

Junliang Guo, Yang Ye, Tianyu He, Haoyu Wu, Yushu Jiang, Tim Pearce, and Jiang Bian. MineWorld: a Real-Time and Open-Source Interactive World Model on Minecraft. *arXiv preprint arXiv:2504.08388*, 2025. doi: 10.48550/arXiv.2504.08388. URL https://arxiv.org/abs/2504.08388.

David Ha and Jürgen Schmidhuber. World Models. *arXiv preprint arXiv:1803.10122*, 2018. doi: 10.48550/arXiv.1803.10122. URL https://arxiv.org/abs/1803.10122.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. LTX-Video: Realtime Video Latent Diffusion. *arXiv preprint arXiv:2501.00103*, 2024. doi: 10.48550/arXiv.2501.00103. URL https://arxiv.org/abs/2501.00103.

Haoran He, Yang Zhang, Liang Lin, Zhongwen Xu, and Ling Pan. Pre-Trained Video Generative Models as World Simulators. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, volume 40, pp. 4645–4653. AAAI Press, 2026. doi: 10.1609/AAAI.V40I6.42465. URL https://doi.org/10.1609/AAAI.V40I6.42465.

Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, Baixin Xu, Hao-Xiang Guo, Kaixiong Gong, Size Wu, Wei Li, Xuchen Song, Yang Liu, Yangguang Li, and Yahui Zhou. Matrix-game 2.0: An open-source real-time and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025. doi: 10.48550/arXiv.2508.13009. URL https://arxiv.org/abs/2508.13009.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

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*, 2025a. doi: 10.48550/arXiv.2505.14357. URL https://arxiv.org/abs/2505.14357.

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*, volume 38, Main Conference, pp. 167283–167308. Curran Associates, Inc., 2025b. doi: 10.52202/085713-5576. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/f4823f831af67a3ef15e41a85434422a-Paper-Conference.pdf.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 17-21, 2024*, pp. 21807–21818. IEEE, 2024. doi: 10.1109/CVPR52733.2024.02060. URL https://doi.org/10.1109/CVPR52733.2024.02060.

Sihui Ji, Xi Chen, Xin Tao, Pengfei Wan, and Hengshuang Zhao. PhysMaster: Mastering Physical Representation for Video Generation via Reinforcement Learning. *arXiv preprint arXiv:2510.13809*, 2025. doi: 10.48550/arXiv.2510.13809. URL https://arxiv.org/abs/2510.13809.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. VACE: All-in-One Video Creation and Editing. In *IEEE/CVF International Conference on Computer Vision*, pp. 17191–17202, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Jiang_VACE_All-in-One_Video_Creation_and_Editing_ICCV_2025_paper.html.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal Flow Matching for Efficient Video Generative Modeling. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, pp. 23378–23402. OpenReview.net, 2025. URL https://openreview.net/forum?id=66NzcRQuOq.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, S. H. Cai, Yuan Cao, Ziwei Chai, Y. Charles, H. S. Che, Cheng Chen, Guanduo Chen, Huarong Chen, Jia Chen, Jianlong Chen, Jun Chen, Kefan Chen, Liang Chen, Ruijue Chen, Xinhao Chen, Yanru Chen, Yanxu Chen, Yicun Chen, Yimin Chen, Yingjiang Chen, Yuankun Chen, Yujie Chen, Yutian Chen, Zhirong Chen, Ziwei Chen, Dazhi Cheng, Yean Cheng, Minghan Chu, Jialei Cui, Jiaqi Deng, Muxi Diao, Hao Ding, Mengfan Dong, Mengnan Dong, Yuxin Dong, Yuhao Dong, Angang Du, Chenzhuang Du, Dikang Du, Lingxiao Du, Yulun Du, Yu Fan, Shengjun Fang, Qiulin Feng, Yichen Feng, Garimugai Fu, Kelin Fu, Hongcheng Gao, Tong Gao, Yuyao Ge, Shangyi Geng, Chengyang Gong, Xiaochen Gong, Zhuoma Gongque, Qizheng Gu, Xinran Gu, Yicheng Gu, Longyu Guan, Shuhao Guan, Yuanying Guo, Xiaoru Hao, Dailan He, Tianhong He, Weiran He, Wenyang He, Yibo He, Yunjia He, Chao Hong, Hao Hu, Jiaxi Hu, Yangyang Hu, Zhenxing Hu, Ke Huang, Ruiyuan Huang, Weixiao Huang, Zhiqi Huang, Chaobo Jia, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yu Jing, Guokun Lai, Aidi Li, C. Li, Cheng Li, Fang Li, Guanghe Li, Guanyu Li, Haitao Li, Haoyang Li, Jia Li, Jingwei Li, Junxiong Li, Lincan Li, Mo Li, Weihong Li, Wentao Li, Xinhang Li, Xinhao Li, Yang Li, Yanhao Li, Yiwei Li, Yuxiao Li, Zhaowei Li, Zhaoxi Li, Zheming Li, Weilong Liao, Jiawei Lin, Xiaohan Lin, Yibo Lin, Zhishan Lin, Zichao Lin, Cheng Liu, Chenyu Liu, Hongzhang Liu, Liang Liu, Shaowei Liu, Shudong Liu, Shuran Liu, Tianwei Liu, Tianyu Liu, Weizhou Liu, Xiangyan Liu, Yangyang Liu, Yanming Liu, Yibo Liu, Yuanxin Liu, Zhengying Liu, Zhongnuo Liu, Enzhe Lu, Haoyu Lu, Zhiyuan Lu, G. Luo, Junyu Luo, Tongxu Luo, Yashuo Luo, Long Ma, Shaoguang Mao, Yuan Mei, Xin Men, Fanqing Meng, Zhiyong Meng, Yibo Miao, Minqing Ni, Kun Ouyang, Siyuan Pan, Bo Pang, Yuchao Qian, Ruoyu Qin, Zeyu Qin, Jiezhong Qiu, Bowen Qu, Zeyu Shang, Youbo Shao, Tianxiao Shen, Zhennan Shen, Juanfeng Shi, Lidong Shi, Shengyuan Shi, Feifan Song, Pengwei Song, Tianhui Song, Xiaoxi Song, Hongjin Su, Jianlin Su, Zhaochen Su, Lin Sui, Jinsong Sun, Junyao Sun, Tongyu Sun, Flood Sung, Yunpeng Tai, Chuning Tang, Heyi Tang, Xiaojuan Tang, Zhengyang Tang, Jiawen Tao, Shiyuan Teng, Chaoran Tian, Pengfei Tian, Bowen Wang, Chensi Wang, Chuang Wang, Congcong Wang, Dingkun Wang, Dinglu Wang, Dongliang

Wang, Feng Wang, Hailong Wang, Haiming Wang, Hao Wang, Hengzhi Wang, Huaqing Wang, Hui Wang, Jiahao Wang, Jinhong Wang, Jiuzheng Wang, Kaixin Wang, Linian Wang, Qibin Wang, Shengjie Wang, Shuyi Wang, Si Wang, Wei Wang, Xiaochen Wang, Xinyuan Wang, Yao Wang, Yejie Wang, Yipu Wang, Yiqin Wang, Yucheng Wang, Yuzhi Wang, Zhaoji Wang, Zhaowei Wang, Zhengtao Wang, Zhexu Wang, Zifan Wang, Zihan Wang, Zizhe Wang, Chu Wei, Ming Wei, Chuan Wen, Zichen Wen, Chengjie Wu, Haoning Wu, Junyan Wu, Rucong Wu, Wenhao Wu, Yuefeng Wu, Yuhao Wu, Yuxin Wu, Zijian Wu, Chenjun Xiao, Jin Xie, Xiaotong Xie, Yuchong Xie, Bowei Xing, Boyu Xu, Jianfan Xu, Jing Xu, Jinjing Xu, L. H. Xu, Lin Xu, Suting Xu, Weixin Xu, Xinbo Xu, Xinran Xu, Yangchuan Xu, Yichang Xu, Yuemeng Xu, Zelai Xu, Ziyao Xu, Junjie Yan, Yuzi Yan, Guangyao Yang, Hao Yang, Junwei Yang, Kai Yang, Ningyuan Yang, Xiaofei Yang, Xinlong Yang, Xinyu Yang, Ying Yang, Yi Yang, Yi Yang, Zhen Yang, Zhilin Yang, Zonghan Yang, Haotian Yao, Dan Ye, Haoran Ye, Wenjie Ye, Zhuorui Ye, Peng Yebo, Bohong Yin, Chengzhen Yu, Longhui Yu, Tao Yu, Tianxiang Yu, Enming Yuan, Mengjie Yuan, Xiaokun Yuan, Yang Yue, Weihao Zeng, Dunyuan Zha, Haobing Zhan, Dehao Zhang, Hao Zhang, Jin Zhang, Puqi Zhang, Qiao Zhang, Rui Zhang, Xiaobin Zhang, Xiaoyun Zhang, Y. Zhang, Yadong Zhang, Yangkun Zhang, Yichi Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yushun Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Chenguang Zhao, Feifan Zhao, Jinxiang Zhao, Shuai Zhao, Xiangyu Zhao, Xuanle Zhao, Yikai Zhao, Zijia Zhao, Huabin Zheng, Ruihan Zheng, Shaojie Zheng, Tengyang Zheng, Junfeng Zhong, Longguang Zhong, Weiming Zhong, M. Zhou, Runjie Zhou, Xinyu Zhou, Zaida Zhou, Jinguo Zhu, Liya Zhu, Xinhao Zhu, Yuxuan Zhu, Zhen Zhu, Jingze Zhuang, Weiyu Zhuang, Ying Zou, and Xinxing Zu. Kimi K2.5: Visual Agentic Intelligence. *arXiv preprint arXiv:2602.02276*, 2026. doi: 10.48550/arXiv.2602.02276. URL https://arxiv.org/abs/2602.02276.

Hanyang Kong, Xingyi Yang, Xiaoxu Zheng, and Xinchao Wang. WorldWarp: Propagating 3D Geometry with Asynchronous Video Diffusion. *arXiv preprint arXiv:2512.19678*, 2025. doi: 10.48550/arXiv.2512.19678. URL https://arxiv.org/abs/2512.19678.

Max Ku, Cong Wei, Weiming Ren, Harry Yang, and Wenhu Chen. AnyV2V: A Tuning-Free Framework For Any Video-to-Video Editing Tasks. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=RFrJCkw2oa.

Pengxiang Li, Kai Chen, Zhili Liu, Ruiyuan Gao, Lanqing Hong, Dit-Yan Yeung, Huchuan Lu, and Xu Jia. TrackDiffusion: Tracklet-Conditioned Video Generation via Diffusion Models. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2025, Tucson, AZ, USA, February 28 - March 4, 2025*, pp. 3539–3548. IEEE, 2025a. doi: 10.1109/WACV61041.2025.00349. URL https://doi.org/10.1109/WACV61041.2025.00349.

Quanhao Li, Zhen Xing, Rui Wang, Haidong Cao, Qi Dai, Daoguo Dong, and Zuxuan Wu. FlashMotion: Few-Step Controllable Video Generation with Trajectory Guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8986–8996, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Li_FlashMotion_Few-Step_Controllable_Video_Generation_with_Trajectory_Guidance_CVPR_2026_paper.html.

Yaowei Li, Xintao Wang, Zhaoyang Zhang, Zhouxia Wang, Ziyang Yuan, Liangbin Xie, Ying Shan, and Yuexian Zou. Image Conductor: Precision Control for Interactive Video Synthesis. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, volume 39, pp. 5031–5038. AAAI Press, 2025b. doi: 10.1609/AAAI.V39I5.32533. URL https://doi.org/10.1609/AAAI.V39I5.32533.

Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. GLIGEN: Open-Set Grounded Text-to-Image Generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 18-22, 2023*, pp. 22511–22521. IEEE, 2023. doi: 10.1109/CVPR52729.2023.02156. URL https://doi.org/10.1109/CVPR52729.2023.02156.

Long Lian, Baifeng Shi, Adam Yala, Trevor Darrell, and Boyi Li. LLM-grounded Video Diffusion Models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, pp. 50207–50227. OpenReview.net, 2024. URL https://openreview.net/forum?id=exKHibougU.

Han Lin, Abhay Zala, Jaemin Cho, and Mohit Bansal. VideoDirectorGPT: Consistent Multi-Scene Video Generation via LLM-Guided Planning. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=sKNIjS2brr.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with Grounded Pre-training for Open-Set Object Detection. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XLVII*, volume 15105 of *Lecture Notes in Computer Science*, pp. 38–55. Springer, 2025. doi: 10.1007/978-3-031-72970-6_3. URL https://doi.org/10.1007/978-3-031-72970-6_3.

Wan-Duo Kurt Ma, J. P. Lewis, and W. Bastiaan Kleijn. TrailBlazer: Trajectory Control for Diffusion-Based Video Generation. In *SIGGRAPH Asia 2024 Conference Papers, SA 2024, Tokyo, Japan, December 3-6, 2024*, pp. 97:1–97:11. ACM, 2024. doi: 10.1145/3680528.3687652. URL https://doi.org/10.1145/3680528.3687652.

Jianbiao Mei, Tao Hu, Xuemeng Yang, Licheng Wen, Yu Yang, Tiantian Wei, Yukai Ma, Min Dou, Botian Shi, and Yong Liu. DreamForge: Motion-Aware Autoregressive Video Generation for Multi-View Driving Scenes. *arXiv preprint arXiv:2409.04003*, 2024. doi: 10.48550/arXiv.2409.04003. URL https://arxiv.org/abs/2409.04003.

Willi Menapace, Stéphane Lathuilière, Sergey Tulyakov, Aliaksandr Siarohin, and Elisa Ricci. Playable Video Generation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 10056–10065. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00993. URL https://openaccess.thecvf.com/content/CVPR2021/html/Menapace_Playable_Video_Generation_CVPR_2021_paper.html.

Willi Menapace, Aliaksandr Siarohin, Stéphane Lathuilière, Panos Achlioptas, Vladislav Golyanik, Sergey Tulyakov, and Elisa Ricci. Promptable Game Models: Text-guided Game Simulation via Masked Diffusion Models. *ACM Transactions on Graphics*, 43(2):17:1–17:16, 2024. doi: 10.1145/3635705. URL https://doi.org/10.1145/3635705.

Chong Mou, Mingdeng Cao, Xintao Wang, Zhaoyang Zhang, Ying Shan, and Jian Zhang. ReVideo: Remake a Video with Motion and Content Control. In *Advances in Neural Information Processing Systems*, volume 37, pp. 18481–18505. Curran Associates, Inc., 2024. doi: 10.52202/079017-0586. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/20e6b4dd2b1f82bc599c593882f67f75-Paper-Conference.pdf.

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. Google DeepMind blog, 2024. URL https://deepmind.google/blog/genie-2-a-large-scale-foundation-world-model/.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026. URL https://qwen.ai/blog?id=qwen3.8.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment Anything in Images and Videos. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, pp. 28085–28128. OpenReview.net, 2025. URL https://openreview.net/forum?id=Ha6RTeWMd0.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 6121–6132. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.00574. URL https://openaccess.thecvf.com/content/CVPR2025/html/Ren_GEN3C_3D-Informed_World-Consistent_Video_Generation_with_Precise_Camera_Control_CVPR_2025_paper.html.

Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, Yihang Chen, Jie Liu, Yansong Cheng, Yao Yao, Jiayi Zhu, Yihao Meng, Kecheng Zheng, Qingyan Bai, Jingye Chen, Zehong Shen, Yue Yu, Xing Zhu, Yujun Shen, and Hao Ouyang. Advancing Open-source World Models. *arXiv preprint arXiv:2601.20540*, 2026. doi: 10.48550/arXiv.2601.20540. URL https://arxiv.org/abs/2601.20540.

Sand.ai, Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, W. Q. Zhang, Weifeng Luo, Xiaoyang Kang, Yuchen Sun, Yue Cao, Yunpeng Huang, Yutong Lin, Yuxin Fang, Zewei Tao, Zheng Zhang, Zhongshu Wang, Zixun Liu, Dai Shi, Guoli Su, Hanwen Sun, Hong Pan, Jie Wang, Jiexin Sheng, Min Cui, Min Hu, Ming Yan, Shucheng Yin, Siran Zhang, Tingting Liu, Xianping Yin, Xiaoyu Yang, Xin Song, Xuan Hu, Yankai Zhang, and Yuqiao Li. MAGI-1: Autoregressive Video Generation at Scale. *arXiv preprint arXiv:2505.13211*, 2025. doi: 10.48550/arXiv.2505.13211. URL https://arxiv.org/abs/2505.13211.

Joonghyuk Shin, Zhengqi Li, Richard Zhang, Jun-Yan Zhu, Jaesik Park, Eli Shechtman, and Xun Huang. MotionStream: Real-Time Video Generation with Interactive Motion Controls. In *International Conference on Learning Representations*, pp. 7381–7406, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/0cece806cd3d1dfad4a893f016ad3d7d-Abstract-Conference.html.

Wenqiang Sun, Fangyun Wei, Jinjing Zhao, Xi Chen, Zilong Chen, Hongyang Zhang, Jun Zhang, and Yan Lu. From Virtual Games to Real-World Play. *arXiv preprint arXiv:2506.18901*, 2025. doi: 10.48550/arXiv.2506.18901. URL https://arxiv.org/abs/2506.18901.

Zhiyu Tan, Xiaomeng Yang, Luozheng Qin, and Hao Li. VidGen-1M: A Large-Scale Dataset for Text-to-video Generation. *arXiv preprint arXiv:2408.02629*, 2024. doi: 10.48550/arXiv.2408.02629. URL https://arxiv.org/abs/2408.02629.

Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, Mojie Chi, Xuyan Chi, Jian Cong, Qinpeng Cui, Fei Ding, Qide Dong, Yujiao Du, Haojie Duanmu, Junliang Fan, Jiarui Fang, Jing Fang, Zetao Fang, Chengjian Feng, Yu Gao, Diandian Gu, Dong Guo, Hanzhong Guo, Qiushan Guo, Boyang Hao, Hongxiang Hao, Haoxun He, Jiaao He, Qian He, Tuyen Hoang, Heng Hu, Ruoqing Hu, Yuxiang Hu, Jiancheng Huang, Weilin Huang, Zhaoyang Huang, Zhongyi Huang, Jishuo Jin, Ming Jing, Ashley Kim, Shanshan Lao, Yichong Leng, Bingchuan Li, Gen Li, Haifeng Li, Huixia Li, Jiashi Li, Ming Li, Xiaojie Li, Xingxing Li, Yameng Li, Yiying Li, Yu Li, Yueyan Li, Chao Liang, Han Liang, Jianzhong Liang, Ying Liang, Wang Liao, J. H. Lien, Shanchuan Lin, Xi Lin, Feng Ling, Yue Ling, Fangfang Liu, Jiawei Liu, Jihao Liu, Jingtuo Liu, Shu Liu, Sichao Liu, Wei Liu, Xue Liu, Zuxi Liu, Ruijie Lu, Lecheng Lyu, Jingting Ma, Tianxiang Ma, Xiaonan Nie, Jingzhe Ning, Junjie Pan, Xitong Pan, Ronggui Peng, Xueqiong Qu, Yuxi Ren, Yuchen Shen, Guang Shi, Lei Shi, Yinglong Song, Fan Sun, Li Sun, Renfei Sun, Wenjing Tang, Boyang Tao, Zirui Tao, Dongliang Wang, Feng Wang, Hulin Wang, Ke Wang, Qingyi Wang, Rui Wang, Shuai Wang, Shulei Wang, Weichen Wang, Xuanda Wang, Yanhui Wang, Yue Wang, Yuping Wang, Yuxuan Wang, Zijie Wang, Ziyu Wang, Guoqiang Wei, Meng Wei, Di Wu, Guohong Wu, Hanjie Wu, Huachao Wu, Jian Wu, Jie Wu, Ruolan Wu, Shaojin Wu, Xiaohu Wu, Xinglong Wu, Yonghui Wu, Ruiqi Xia, Xin Xia, Xuefeng Xiao, Shuang Xu, Bangbang Yang, Jiaqi Yang, Runkai Yang, Tao Yang, Yihang Yang, Zhixian Yang, Ziyan Yang, Fulong Ye, Bingqian Yi, Xing Yin, Yongbin You, Linxiao Yuan, Weihong Zeng, Xuejiao Zeng, Yan Zeng, Siyu Zhai, Zhonghua Zhai, Bowen Zhang, Chenlin Zhang, Heng Zhang, Jun Zhang, Manlin Zhang, Peiyuan Zhang, Shuo Zhang, Xiaohe Zhang, Xiaoying Zhang, Xinyan Zhang, Xinyi Zhang, Yichi Zhang, Zixiang Zhang, Haiyu Zhao, Huating Zhao, Liming Zhao, Yian Zhao, Guangcong Zheng, Jianbin Zheng, Xiaozheng Zheng, Zerong Zheng, Kuan Zhu, and Feilong Zuo. Seedance 2.0: Advancing Video Generation for World

Complexity. *arXiv preprint arXiv:2604.14148*, 2026. doi: 10.48550/arXiv.2604.14148. URL https://arxiv.org/abs/2604.14148.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and Advanced Large-Scale Video Generative Models. *arXiv preprint arXiv:2503.20314*, 2025. doi: 10.48550/arXiv.2503.20314. URL https://arxiv.org/abs/2503.20314.

Ye Tian, Ling Yang, Haotian Yang, Yuan Gao, Yufan Deng, Jingmin Chen, Xintao Wang, Zhaochen Yu, Xin Tao, Pengfei Wan, Di Zhang, and Bin Cui. VideoTetris: Towards Compositional Text-to-Video Generation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 29489–29513. Curran Associates, Inc., 2024. doi: 10.52202/079017-0928. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/345208bdbbb6104616311dfc1d093fe7-Paper-Conference.pdf.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion Models Are Real-Time Game Engines. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, pp. 73754–73776. OpenReview.net, 2025. URL https://openreview.net/forum?id=P8pqeEkn1H.

Angtian Wang, Haibin Huang, Jacob Zhiyuan Fang, Yiding Yang, and Chongyang Ma. ATI: Any Trajectory Instruction for Controllable Video Generation. *arXiv preprint arXiv:2505.22944*, 2025a. doi: 10.48550/arXiv.2505.22944. URL https://arxiv.org/abs/2505.22944.

Chen Wang, Chuhao Chen, Yiming Huang, Zhiyang Dou, Yuan Liu, Jiatao Gu, and Lingjie Liu. PhysCtrl: Generative Physics for Controllable and Physics-Grounded Video Generation. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 167907–167932. Curran Associates, Inc., 2025b. doi: 10.52202/085713-5598. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/f53fd88a4340063ecd258c0ae9948b40-Paper-Conference.pdf.

Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. VideoComposer: Compositional Video Synthesis with Motion Controllability. In *Advances in Neural Information Processing Systems*, volume 36, pp. 7594–7611. Curran Associates, Inc., 2023. doi: 10.52202/075280-0334. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/180f6184a3458fa19c28c5483bc61877-Paper-Conference.pdf.

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. MotionCtrl: A Unified and Flexible Motion Controller for Video Generation. In *ACM SIGGRAPH 2024 Conference Papers, SIGGRAPH 2024, Denver, CO, USA, 28 July 2024- 1 August 2024*, pp. 114:1–114:11. ACM, 2024. doi: 10.1145/3641519.3657518. URL https://doi.org/10.1145/3641519.3657518.

Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. iVideoGPT: Interactive VideoGPTs are Scalable World Models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 68082–68119. Curran Associates, Inc., 2024. doi: 10.52202/079017-2173. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/7dbb5bfab324e3b86af9bd0df15498dd-Paper-Conference.pdf.

Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. WorldMem: Long-term Consistent World Simulation with Memory. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 49632–49652. Curran Associates, Inc., 2025. doi: 10.52202/

085713-1659. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/470629a47e2d65ce0606c40055df5d26-Paper-Conference.pdf.

Ziyang Xie, Zhizheng Liu, Zhenghao Peng, Wayne Wu, and Bolei Zhou. Vid2Sim: Realistic and Interactive Simulation from Video for Urban Navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 1581–1591. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.00155. URL https://openaccess.thecvf.com/content/CVPR2025/html/Xie_Vid2Sim_Realistic_and_Interactive_Simulation_from_Video_for_Urban_Navigation_CVPR_2025_paper.html.

Linjie Yang, Yuchen Fan, Yang Fu, and Ning Xu. The 3rd Large-scale Video Object Segmentation Challenge - Video Instance Segmentation Track, June 2021. URL https://youtube-vos.org/dataset/vis/.

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*, 2025a. doi: 10.48550/arXiv.2509.22622. URL https://arxiv.org/abs/2509.22622.

Xindi Yang, Baolu Li, Yiming Zhang, Zhenfei Yin, Lei Bai, Liqian Ma, Zhiyong Wang, Jianfei Cai, Tien-Tsin Wong, Huchuan Lu, and Xu Jia. VLIPP: Towards Physically Plausible Video Generation with Vision and Language Informed Physical Prior. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-23, 2025*, pp. 12360–12370. IEEE, 2025b. doi: 10.1109/ICCV51701.2025.01149. URL https://doi.org/10.1109/ICCV51701.2025.01149.

Shengming Yin, Chenfei Wu, Jian Liang, Jie Shi, Houqiang Li, Gong Ming, and Nan Duan. DragNUWA: Fine-grained Control in Video Generation by Integrating Text, Image, and Trajectory. *arXiv preprint arXiv:2308.08089*, 2023. doi: 10.48550/arXiv.2308.08089. URL https://arxiv.org/abs/2308.08089.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step Diffusion with Distribution Matching Distillation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6613–6623, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yin_One-step_Diffusion_with_Distribution_Matching_Distillation_CVPR_2024_paper.html.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo Durand, Eli Shechtman, and Xun Huang. From Slow Bidirectional to Fast Autoregressive Video Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 22963–22974. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.02138. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yin_From_Slow_Bidirectional_to_Fast_Autoregressive_Video_Diffusion_Models_CVPR_2025_paper.html.

Jiwen Yu, Yiran Qin, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. GameFactory: Creating New Games with Generative Interactive Videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11590–11599, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Yu_GameFactory_Creating_New_Games_with_Generative_Interactive_Videos_ICCV_2025_paper.html.

Denis Zavadski, Johann-Friedrich Feiden, and Carsten Rother. ControlNet-XS: Rethinking the Control of Text-to-Image Diffusion Models as Feedback-Control Systems. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXXVIII*, volume 15146 of *Lecture Notes in Computer Science*, pp. 343–362. Springer, 2025. doi: 10.1007/978-3-031-73223-2_20. URL https://doi.org/10.1007/978-3-031-73223-2_20.

Ke Zhang, Cihan Xiao, Jiacong Xu, Yiqun Mei, and Vishal M. Patel. Think Before You Diffuse: Infusing Physical Rules into Video Diffusion. *arXiv preprint arXiv:2505.21653*, 2025. doi: 10.48550/arXiv.2505.21653. URL https://arxiv.org/abs/2505.21653.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 2-6, 2023*, pp. 3813–3824. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00355. URL https://doi.org/10.1109/ICCV51070.2023.00355.

Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, Xiaopeng Zhang, Wangmeng Zuo, and Qi Tian. ControlVideo: Training-free Controllable Text-to-video Generation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, pp. 54441–54461. OpenReview.net, 2024a. URL https://openreview.net/forum?id=5a79AqFr0c.

Yuanhan Zhang, Bo Li, Haotian Liu, Yong Jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. LLaVA-NeXT: A Strong Zero-shot Video Understanding Model. LLaVA project blog, 2024b. URL https://llava-vl.github.io/blog/2024-04-30-llava-next-video/.

Zewei Zhang, Jia Jun Cheng Xian, Kaiwen Liu, Ming Liang, Hang Chu, Jun Chen, and Renjie Liao. TrajLoom: Dense Future Trajectory Generation from Video. *arXiv preprint arXiv:2603.22606*, 2026. doi: 10.48550/arXiv.2603.22606. URL https://arxiv.org/abs/2603.22606.

Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K. Wong. Uni-ControlNet: All-in-One Control to Text-to-Image Diffusion Models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 11127–11150. Curran Associates, Inc., 2023. doi: 10.52202/075280-0491. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/2468f84a13ff8bb6767a67518fb596eb-Paper-Conference.pdf.

Haitao Zhou, Chuang Wang, Rui Nie, Jinlin Liu, Dongdong Yu, Qian Yu, and Changhu Wang. TrackGo: A Flexible and Efficient Method for Controllable Video Generation. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, volume 39, pp. 10743–10751. AAAI Press, 2025. doi: 10.1609/AAAI.V39I10.33167. URL https://doi.org/10.1609/AAAI.V39I10.33167.

Haoyi Zhu, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Chunhua Shen, Jiangmiao Pang, and Tong He. AETHER: Geometric-Aware Unified World Modeling. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-23, 2025*, pp. 8535–8546. IEEE, 2025. doi: 10.1109/ICCV51701.2025.00799. URL https://doi.org/10.1109/ICCV51701.2025.00799.

A VIDEO-TO-STRUCTURE ANNOTATION

A.1 DATA PREPARATION

This section describes how we collect, filter, and annotate the training videos.

A.1.1 COLLECTION AND FILTERING

Collection: We collect videos from two sources:

- **Online Datasets:** We rely on two large-scale datasets: VideoGen-1M (Tan et al., 2024) and 4DNeX (Chen et al., 2025b), which capture diverse objects, environments, and interactions.
- **Gameplay Data:** We collect diverse gameplay videos from online sources, which provide structured object behaviors and interactions.

Filtering: Once videos are collected, we apply a filtering process to ensure only high-quality, relevant videos are included in the dataset:

- **Resolution and Quality Filtering:** We only select videos with a minimum resolution of 720p, 16 FPS, and 81 frames.
- **Content Relevance:** We keep only videos in which objects are identifiable and in motion.

A.1.2 DATA ANNOTATION

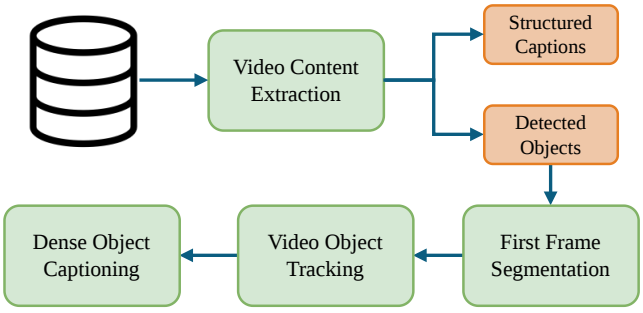


Figure 6: Video-to-Structure Annotation Pipeline.

Our data labeling process follows a three-stage approach: **Video Content Extraction**, **Object Detection and Tracking**, and **Object Captioning**. For gameplay data, we implement an additional dedicated stage: **Structural Video Description with Element Breakdown**. Figures 7 and 8 show two representative examples of our annotation results, drawn from the online dataset and the gameplay dataset, respectively.

Video Content Extraction. In the initial stage of labeling, we extract high-level content from each video using LLaVA-NeXT-Video (Zhang et al., 2024b). This model processes the entire video and performs two main tasks:

- **Object Identification:** The model identifies the main objects present in each video to provide an initial list of visual entities appearing in the scene, such as humans, vehicles, animals, and inanimate objects.
- **Full Video Description and Background Generation:** The model then generates a high-level description of the video, capturing not only the main objects but also the overall context of the scene. This includes background descriptions that characterize the environment, lighting, and any static elements in the scene that remain unchanged throughout the video.

Object Detection and Tracking. Once objects have been identified by the video captioning model, we proceed to detect and track the position of these objects across video frames.

- **Initial Object Detection:** Grounding DINO (Liu et al., 2025) detects the identified objects in the first frame and produces their initial bounding boxes.
- **Object Tracking:** After the initial detection, we use SAM 2 (Ravi et al., 2025) to track these objects across subsequent frames. SAM 2 allows us to maintain temporal consistency by tracking the same object across frames, even if it undergoes motion, partial occlusion, or appearance changes due to lighting or viewpoint shifts.

Object Captioning. In the final labeling step, we crop each tracked object and caption it frame by frame with an image captioning model.

- **Object Descriptions:** Each caption describes the object's identity and appearance (e.g., "a red car"), which forms its persistent attribute.
- **Temporal Captioning:** Each caption also describes the object's current status and action, with emphasis on status changes, which forms its dynamic status. For example, the red car might be described as "a red car driving down the road" in one frame and "a red car stopping at a traffic light" in another.

Structural Video Description with Element Breakdown. For gameplay datasets, we enhance the standard video description by including additional elements that are specific to game environments and interactions. This breakdown captures both the visual aspects and the dynamic game mechanics. Specifically, we expand on the following key elements:

- **Spatial Layout:** Descriptions of how objects are spatially arranged in the frame, such as their relative positioning and proximity to other objects or the camera. For example, "The racing car is positioned in the center of the track, ahead of the player's car".
- **Action and Interaction Descriptions:** These provide detailed information about how objects interact with each other. We annotate the relationship between the primary player-controlled entity and the adversaries or objectives. This captures the behavioral logic of a game, where the player's actions directly influence the target's state and vice-versa. For example, "The racing car aggressively closes the gap with the lead vehicle".
- **Background Descriptions:** Beyond individual object interactions, broader scene descriptions are generated, focusing on the environment or setting where the objects exist. For example, "An urban street with tall skyscrapers, neon signs, and a crowd of pedestrians walking."

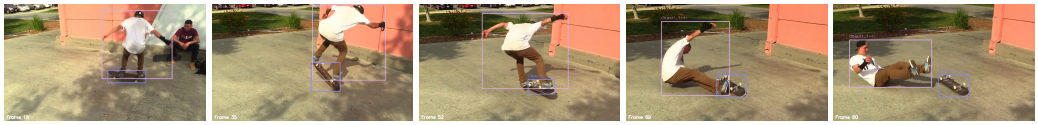
A.2 Example

Figures 7 and 8 illustrate how raw videos are converted into the object-centric state used by BoxWorld. Figure 7 shows an open-domain clip from VideoGen-1M, where the annotation pipeline extracts a global video description, tracks salient objects across frames, and captions each object frame by frame. The captions describe both the persistent appearance of each object and its changing status, such as the skateboarder jumping, losing balance, and sitting on the pavement after the fall. Figure 8 shows a gameplay clip, whose global description is further structured into the main object, camera, environment, targets, and interactions.

B Box World Engine Specification

A Box World Engine is an online state producer for the Object-Centric State Interface. It runs as a lightweight state machine that maintains entities, global variables, and authored update rules. At each step, the engine receives a user action, scripted policy output, or external control signal; evaluates the subset of rules that applies to the current state; updates object boxes, attributes, and dynamic statuses; and publishes the resulting state $\mathcal{S}_t$ to a state buffer. The renderer reads the latest state from the buffer asynchronously, while the engine continues stepping independently. This design keeps interaction logic editable and inspectable without coupling it to the renderer's denoising schedule or internal latent representation.

Global caption "The video features a person in a white t-shirt and brown pants attempting a skateboard trick. He is positioned on a concrete surface with a pink wall in their back. The individual starts by balancing on the skateboard, then jumps into the air while flipping the board. As they land, they lose balance and fall to the ground, ending up sitting on the pavement with the skateboard beside them."



Object Descriptions:
Frame 18:
object 1: "The skateboard in the scene is a sleek, black deck with white wheels and trucks. The grip tape on top of the board appears to be worn from frequent use, suggesting it's well-loved by its owner. The underside of the board shows visible signs of wear as well, indicating that this isn't just any ordinary skateboard but one that has been used extensively for skateboarding stunts or tricks."
object 2: "The person in the scene is a young man skillfully riding his skateboard. He's wearing casual attire, including a white t-shirt and brown pants, which contrasts with the vibrant pink wall behind him. His black baseball cap adds to his urban style as he maneuvers through what appears to be an outdoor setting. The motion of skateboarding suggests energy and agility, capturing a dynamic moment that encapsulates youth culture and street sports."

Frame 35:
object 1: "The skateboard in the scene is a classic design with four wheels, two at each end. It features white grip tape on its top surface and appears to be made of wood or composite material. The trucks are visible beneath the deck, connecting it to the wheels, while the underside shows the bottom layer where the bearings reside."
object 2: "The person in the scene is a young man performing an action on his skateboard. He's wearing casual attire, including a white t-shirt and brown pants, with black sneakers that match his cap. His posture suggests he might be executing a trick or maneuver as he moves along the sidewalk next to a pink wall."

Frame 52:
object 1: "The skateboard in the scene is a classic design with four wheels, two at each end. It features a smooth deck and trucks that connect to the wheels for movement. The grip tape on top of the board provides traction when riding."
object 2: "The person in the scene is a young skateboarder wearing casual clothing, a white t-shirt paired with brown pants rolled up to the knees for ease of movement. The skateboarder has one arm extended behind for balance during the maneuver."

Frame 69:
object 1: "The skateboard in the scene is a classic design with four wheels, two at each end. It features white trucks and yellow wheels that contrast against its black deck. The grip tape on top of the board appears to be worn out, suggesting it has been used extensively for skateboarding activities."
object 2: "The person in the scene is a young man losing balance on his skateboard after the trick. He is dressed casually, wearing a white t-shirt and brown pants paired with black sneakers, with his arms extended as he tries to stay upright."

Frame 80:
object 1: "The skateboard in the scene is a classic design with four yellow wheels attached to its underside. The deck of the board features an intricate pattern, possibly graffiti or artwork, adding character and individuality to it. The trucks are silver-colored metal pieces that connect the wheels to the deck, providing stability for the rider's feet as they glide across surfaces."
object 2: "The person in the scene is a young man in casual attire, wearing a white t-shirt, brown pants, black sneakers, and a backward cap. He is sitting on the pavement after the fall, with the skateboard beside him."

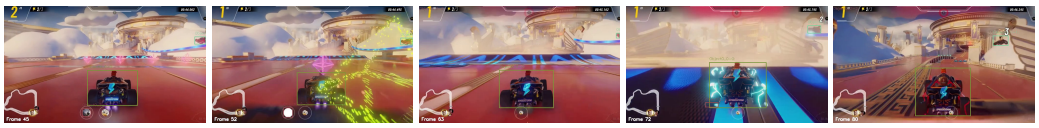
Figure 7: An example of our data annotation results from VideoGen-1M (Tan et al., 2024).

Engine components. Each engine instance is specified by four components. **Entities** define the active objects, including an identifier, semantic role, initial box or world-coordinate extent, persistent caption, and optional metadata. **Update rules** define how entities and optional global variables change in response to control signals and current state, including movement, activation, state transitions, visibility changes, object spawning, or other event-driven updates. **Camera and projection** parameters are used internally to map engine coordinates to image coordinates and to produce frame-level boxes. **State publishing** converts the updated engine state into the interface consumed by BoxWorld: a time-indexed stream of object entries $(\mathbf{b}_{t,i}, \mathbf{c}_i, \mathbf{d}_{t,i})$, where boxes are normalized frame coordinates, attributes describe persistent identity and appearance, and dynamic statuses describe the current status of each object.

Authoring modes. In our experiments, a coding agent generates each engine from user-specified rules and the shared engine template; the engine can also be provided directly by the user. An author specifies the entities, the control inputs they can receive, the rules that update object state, and the projection used to expose boxes in image coordinates. Changing the interaction behavior then amounts to editing the update rules or the labels attached to entities, while the renderer continues to consume the same Object-Centric State Interface.

Reuse and authoring cost. The interface and renderer are shared across worlds, while each engine supplies its own transition rules. An existing 2D or 3D engine can retain its native mechanics and physics, project its active entities into image-space boxes, and export the associated state descriptions.

Global caption (Gameplay Data): "Main Object: The main object in the video is a racing car with blue and purple glowing accents, speeding through a grand, ornate track.; The car is moving forward on the track, accelerating and navigating through the course.; Main Camera: The camera follows the car's motion, capturing the dynamic and fast-paced nature of the race.; Landscape Environment: The background environment is a grand, ancient-style arena with golden pillars and a large circular emblem on the ceiling. The track is adorned with blue and pink glowing elements, and there are clouds in the sky.; Time/Weather/Lighting: The video takes place in a time of day that appears to be either dawn or dusk, given the soft lighting and the presence of clouds in the sky. The weather conditions is clear with no visible rain or fog. The lighting is bright enough to clearly see the details of the race track and the surroundings, including the golden structures and the blue neon lights on the track.; Targets: The secondary target in the video is a red car that appears on the right side of the track, slightly ahead of the main car.; The secondary target, a black car with blue accents and a lightning bolt symbol on the back, is moving in the same direction as the primary car. It maintains a consistent distance behind the primary car throughout the video.; Interaction: The car navigates through the track, passing under a sign that reads "LAP 2" and continues to race towards the finish line.; The video features a racing game where the player's car is seen speeding through a grand, ornate track. The environment is richly detailed with classical architecture, including columns and statues, set against a bright, sunny sky. The track is adorned with various visual effects such as blue and pink glowing lines and circular markers indicating the path. The player's car navigates through the track, overtaking other opponents, and collects power-ups that enhance its speed and performance. The car's speed is indicated by a blue and purple exhaust effect, and the player maintains a high position in the race."



Object Descriptions:
Frame 45:
object 1: "The car in the scene is a high-performance vehicle, likely designed for speed and agility. It features sleek aerodynamic lines with vibrant colors that suggest it's built to stand out on the road or track. The rear of the car prominently displays an emblem resembling a lightning bolt, which could be indicative of its powerful engine capabilities. Additionally, there are visible exhaust pipes at the back, hinting at the performance-oriented nature of this machine."
object 2: "The car in the scene is a vibrant red, with its glossy finish reflecting light. The vehicle's design features sleek lines and curves that give it an aerodynamic appearance. It has tinted windows which add to its stylish look, while also providing privacy for passengers inside."

Frame 52:
object 1: "The car in the scene is a futuristic vehicle with an aerodynamic design, featuring sleek lines and angular shapes. It has large black tires that contrast against its predominantly dark color scheme. The body of the car appears to be made from a metallic material, giving it a reflective quality. On top of the car, there's a glowing blue light source emanating bright particles or sparks around it, suggesting some form of energy emission or propulsion system."
object 2: "The car in the scene is a red sports car seen from behind, with a glossy finish and sleek, aerodynamic lines. It stays slightly ahead on the right side of the track."

Frame 63:
object 1: "The car in the scene is a vibrant, high-performance vehicle with an aerodynamic design. It features striking blue and orange accents that give it a dynamic appearance. The hood of the car has a prominent lightning bolt graphic painted on it, adding to its aggressive look."
object 2: "The car in the scene is a striking blend of vibrant hues, with its body painted in a rich shade of red that transitions into a deep pink towards the front. The vehicle's design features sleek lines and curves, giving it an aerodynamic appearance. Its wheels are not visible from this angle, but one can imagine them to be well-maintained and complementing the overall aesthetic of the car."

Frame 72:
object 1: "The car in the scene is a sleek, futuristic vehicle with an aerodynamic design. It features vibrant blue and black colors that give it a high-tech appearance. The body of the car has intricate details such as glowing lines running along its side, adding to its advanced look. On top of the car, there's a small structure or attachment resembling a lightning bolt, which stands out against the dark background."
object 2: "The car in the scene is a red sports car with a glossy finish. Its sharp angles and aerodynamic lines suggest speed as it continues along the track."

Frame 80:
object 1: "The car in the scene is a sleek, high-performance vehicle with an aerodynamic design. It features vibrant colors and intricate detailing on its bodywork, including what appears to be glowing blue accents that add to its futuristic aesthetic. The rear of the car has prominent wings and exhaust pipes, suggesting it's built for speed and power."
object 2: "The car in the scene is a red sports car with a glossy body and rounded edges, still racing along the track ahead of the camera."

Figure 8: An example of our data annotation results from gameplay videos.

In our experiments, implementing a new rule system from the shared engine template requires 205–714 lines of code (median 351). A new visual or scene variant typically requires a 10–20-line JSON configuration, after which the engine produces the trajectories, event timing, and counterfactual branches programmatically.

C Additional Experimental Details and Results

C.1 Implementation Details

Stage 1 uses filtered open-domain videos from VideoGen-1M (Tan et al., 2024) and 4DNeX (Chen et al., 2025b), which provide diverse objects, environments, and mostly smooth object motion. Stage 2 uses 4,000 short gameplay clips collected online, where object roles, interactions, and event consequences are more explicit. We keep clips with sufficient resolution, frame rate, duration, visible

Algorithm 1 Generic Online Box World Engine

Initialize entities $\mathcal{E}$, global state $\mathbf{g}$, and camera/projection parameters C
Initialize authored update rules $\mathcal{R}$ and output state buffer $\mathcal{B}$
while the interactive session is active **do**
    Receive user action, scripted policy output, or external control signal u_t
    Evaluate applicable rules $\mathcal{R}_t \subseteq \mathcal{R}$ from $(\mathcal{E}, \mathbf{g}, u_t)$
    Update entity boxes, attributes, dynamic statuses, and global state using $\mathcal{R}_t$
    Add, remove, activate, or deactivate entities if specified by the rules
    Project active entities to frame coordinates using C
    Publish $\mathcal{S}_t = \{(\mathbf{b}_{t,i}, \mathbf{c}_i, \mathbf{d}_{t,i})\}_{i=1}^{N_t}$ to $\mathcal{B}$
    The renderer asynchronously reads the latest state from $\mathcal{B}$ for video synthesis
end while

moving objects, and reliable detection and tracking results, and discard clips with severe blur, unstable views, or ambiguous object identities (Appendix A).

We train the State Grounding Module for 20K iterations with batch size 32, using AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay $= 0.01$) and linear warmup to a peak learning rate of 10^{-4}. Stage 2 trains rank-16 LoRA adapters for 10K iterations with the same batch size and optimizer, using linear warmup to 3×10^{-5} followed by cosine decay to 10% of the peak learning rate. Stage 3 trains a separate rank-64 LoRA adapter for 400 DMD steps with all preceding components frozen. We use FlexAttention, bfloat16 mixed precision, and gradient checkpointing. At inference, the State Grounding Module is active only at noise levels above 0.8. The Box World Engines are created by Kimi K2.5 (Kimi Team et al., 2026) from the shared engine template in Appendix B.

C.2 PairedStateBench Protocol

Benchmark design. PairedStateBench tests whether a generated video realizes the consequences that user-defined rules assign to user actions. Each case pairs an engine-exported state trajectory with a specification of the events that must occur, the consequences that must not occur, and the state each object should hold over time. The two branches of a pair share the same scene and initial layout but receive different actions, so a model passes a pair only when it produces the intended outcome in both branches. The benchmark contains 87 cases from 16 scene families, such as object handling, credential-dependent doors, charging, gravity control, and pressure-plate doors, and includes 37 branch pairs. The easy, medium, and hard subsets contain 23, 28, and 36 cases, respectively.

Model inputs. BoxWorld uses the global prompt and the per-object box and dynamic-status sequence exported by the engine. LongLive uses the same global prompt without the object-level state sequence. LingBot-v1 uses the first frame of the case, the global prompt, and its native keyboard and camera inputs converted from the engine's navigation actions. LingBot-v2 uses the first frame, the global prompt, and camera conditions, as its public release exposes only camera-path conditioning. The oracle variants additionally receive a text description of the resolved state trajectory, which specifies the intended interaction consequences.

Metrics.

- *Trigger*: the fraction of required events that appear in the video.
- *Suppress*: the fraction of forbidden consequences that remain absent. Easy cases specify single actions without forbidden consequences, so Suppress does not apply to them.
- *Paired-CF*: the fraction of pairs in which both branches realize their target outcomes.
- *OnTime*: the fraction of required events whose located onset f and engine boundary $f^\star$ satisfy
$$|\lfloor f/12 \rfloor - \lfloor f^\star/12 \rfloor| \leq 1, \tag{6}$$
using decoded frame indices at 16 FPS.
- *State-Fid*: whether each object holds its specified state within each state window. State-Fid excludes frames within the OnTime tolerance of each state boundary, so that it measures

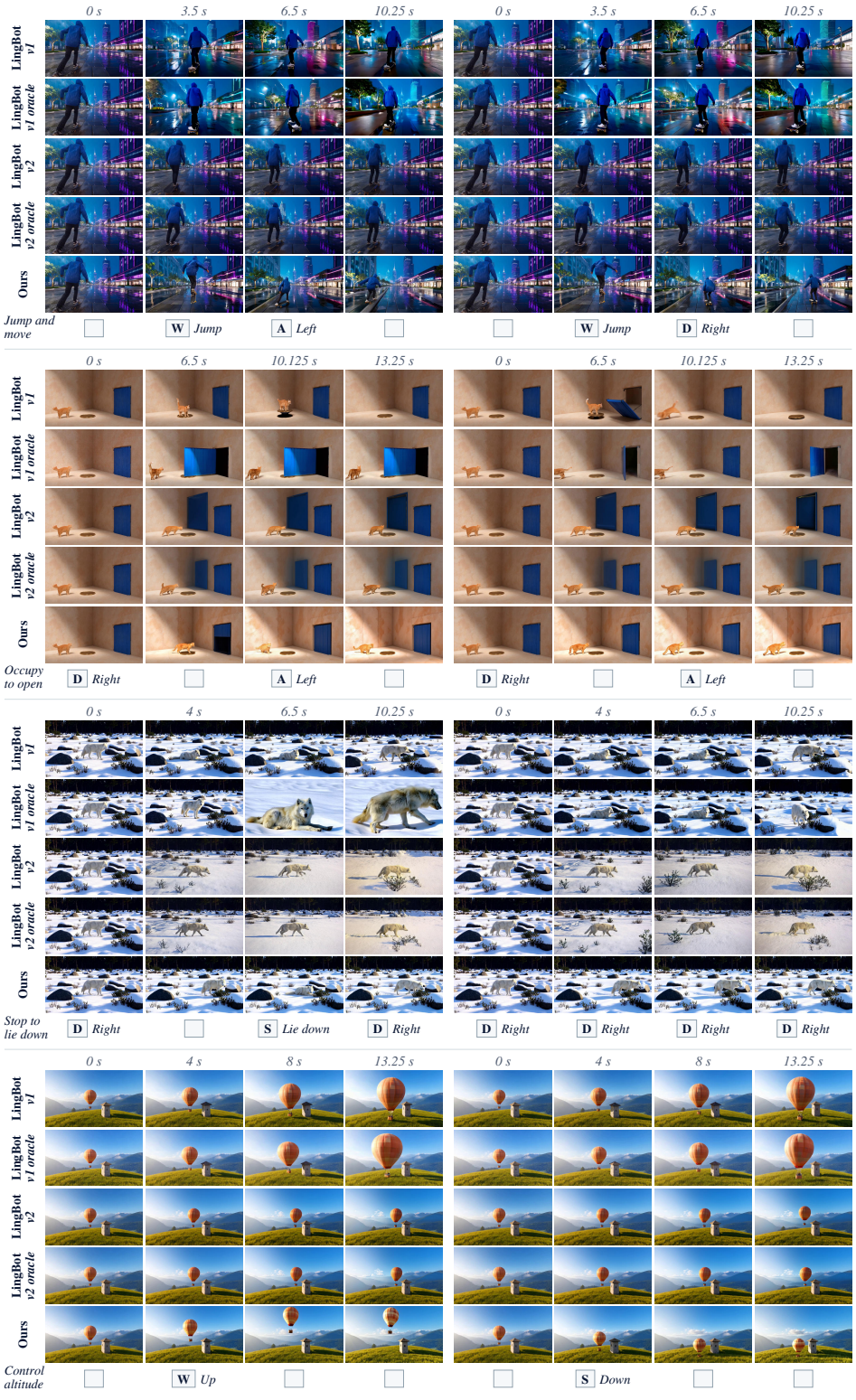


Figure 9: Additional qualitative comparisons on PairedStateBench with LingBot-v1, LingBot-v2, and their oracle variants: jumping and moving, occupying a pressure plate to open a door, stopping to lie down, and controlling altitude. Each case compares two branches with different user actions. Videos are provided in the supplementary material.



Figure 10: Additional qualitative VBench results comparing LongLive with the three stages of BoxWorld.

whether each state holds during its stable phase, while OnTime measures when the state switches.

- *Task*: whether all action stages, forbidden-state constraints, event ordering, and final outcomes hold in the same video.

Single-video metrics are averaged within each video and then over videos.

Evaluation. Kimi K2.5 (Kimi Team et al., 2026) and Qwen3.8-27B (Qwen Team, 2026) inspect chronological contact sheets of each decoded video and judge event outcomes, state agreement, and onset frames, and judgments on which the two models disagree are resolved by manual review. The two models disagree on 39 of the 522 evaluated videos (7.5%), mainly where camera following or perspective scaling makes the direction or height of motion ambiguous. These judgments are compared with the engine reference to compute each metric. We will release the benchmark cases and evaluation code. Figure 9 shows additional qualitative comparisons.

C.3 Video Quality

Table 3: Video quality on VBench and 30-second VBench-Long. FPS measures generation throughput; baseline scores follow their reported evaluation settings.

		VBench			VBench-Long (30 s)		
Model	FPS$^{\uparrow}$	Total$^{\uparrow}$	Quality$^{\uparrow}$	Semantic$^{\uparrow}$	Total$^{\uparrow}$	Quality$^{\uparrow}$	Semantic$^{\uparrow}$
LTX-Video (HaCohen et al., 2024)	8.98	80.00	82.30	70.79	–	–	–
Wan2.1 (Team Wan et al., 2025)	0.78	84.26	85.30	80.09	–	–	–
SkyReels-V2 (Chen et al., 2025a)	0.49	82.67	84.70	74.53	75.29	80.77	53.37
MAGI-1 (Sand.ai et al., 2025)	0.19	79.18	82.04	67.74	–	–	–
NOVA (Deng et al., 2025)	0.88	80.12	80.39	79.05	–	–	–
Pyramid Flow (Jin et al., 2025)	6.7	81.72	84.74	69.62	–	–	–
CausVid (Yin et al., 2025)	17.0	81.20	84.05	69.80	–	–	–
Self Forcing (Huang et al., 2025b)	17.0	84.31	85.07	81.28	81.59	83.82	72.70
LongLive (Yang et al., 2025a)	20.7	84.87	86.97	76.47	83.52	85.44	75.82
BoxWorld (Stage 1)	20.7	83.30	84.46	78.62	81.72	82.81	77.36
BoxWorld (Stage 2)	20.7	83.57	84.22	80.97	81.82	82.85	77.71
BoxWorld (Stage 3)	20.7	83.76	84.82	79.54	83.00	84.34	77.67

We follow the standard VBench and VBench-Long protocols, generating videos at 832×480 with four denoising steps. VBench videos have 81 frames, and VBench-Long videos extend to 30 seconds with the rolling KV cache of LongLive (Yang et al., 2025a). For each prompt, the object states are obtained by applying the annotation pipeline of Sec. 3.4 to a video generated by LongLive from the same prompt. Table 3 reports the VBench and VBench-Long results discussed in Sec. 4.3, together with additional non-real-time and real-time baselines. Compared with LongLive, Stage 1 improves the Semantic score from 76.47 to 78.62 while maintaining competitive Total and Quality scores, indicating that the State Grounding Module injects object-region conditions at high noise levels without substantially degrading visual quality. Stage 2 further improves the Semantic score to 80.97, and Stage 3 raises Total from 83.57 to 83.76 and Quality from 84.22 to 84.82. Figure 10 shows qualitative examples comparing LongLive with the three stages.

C.4 Layout-to-Video Generation

We follow the YouTube-VIS 2021 setting and evaluation protocol of BlobGEN-Vid (Feng et al., 2025b), where ground-truth tracks provide layout boxes and per-track descriptions provide object attributes. Table 4 reports the full comparison, and Table 6 reports the grounding ablations under the same protocol.

C.5 Additional Ablations

Quantitative adaptation ablations. Table 5 evaluates the adaptation settings on VBench. Removing the spatial mask from Stage 3 reduces Total from 83.76 to 80.33 and Semantic from 79.54 to

Table 4: Layout-to-video generation on YouTube-VIS 2021. Baseline scores are quoted from prior work, and all metrics follow the protocol of BlobGEN-Vid (Feng et al., 2025b).

Method	Real-time	FVD$^{\downarrow}$	mIOU$^{\uparrow}$	rCLIP$_t^{\uparrow}$	rCLIP$_i^{\uparrow}$	rCFC$^{\uparrow}$
TrackDiffusion (Li et al., 2025a)	No	464	0.492	0.273	0.804	0.940
LVD (Lian et al., 2024)	No	558	0.281	0.261	0.803	0.861
VideoTetris (Tian et al., 2024)	No	590	0.166	0.267	0.799	0.423
BlobGEN-Vid (Feng et al., 2025b)	No	317	0.598	0.289	0.836	0.958
BoxWorld (Stage 3)	Yes	62	0.572	0.291	0.837	0.965

Table 5: Quantitative VBench ablations. The mask ablation is applied to Stage 3, and the adapter variants are compared with Stage 2; all variants use the four-step streaming sampler.

Variant	Quality$^{\uparrow}$	Semantic$^{\uparrow}$	Total$^{\uparrow}$
Stage 3	**84.82**	79.54	**83.76**
w/o spatial mask	81.78	74.51	80.33
Stage 2	84.22	**80.97**	83.57
w/o dynamics LoRA (Stage 1)	84.46	78.62	83.30
High-noise-only LoRA	83.84	77.44	82.56
Open-domain video LoRA	82.68	79.16	81.98

Table 6: Grounding ablations on YouTube-VIS. Both ablations are applied to Stage 3; the attribute ablation replaces every per-box caption with "object," retaining the global prompt, boxes, and original evaluation targets. All rows use four-step causal autoregressive generation.

Variant	FVD$^{\downarrow}$	mIOU$^{\uparrow}$	rCLIP$_t^{\uparrow}$	rCLIP$_i^{\uparrow}$	rCFC$^{\uparrow}$
Stage 3	62	0.572	0.291	0.837	0.965
w/o spatial mask	79	0.291	0.264	0.783	0.955
w/o object attributes	68	0.323	0.273	0.785	0.963

Table 7: Training stages on PairedStateBench. All variants use the same engine-exported state sequences.

Variant	Trigger$^{\uparrow}$	Suppress$^{\uparrow}$	Paired-CF$^{\uparrow}$	State-Fid$^{\uparrow}$	OnTime$^{\uparrow}$	Task$^{\uparrow}$
Stage 1	87.3	60.3	64.9	83.4	83.8	49.4
Stage 2	82.0	85.7	67.6	83.6	81.5	64.4
Stage 3	**92.2**	**90.5**	**97.3**	**91.6**	**91.2**	**83.9**

74.51. Among the adapter variants, the full Stage 2 model obtains the highest Semantic and Total scores, 80.97 and 83.57, while removing the Stage 2 adapters reduces Semantic from 80.97 to 78.62. Applying the adapters only at noise levels above 0.8 and training them on open-domain videos instead of gameplay videos yield Total scores of 82.56 and 81.98, respectively.

Training stages on PairedStateBench. Table 7 reports all PairedStateBench metrics for the three stages. Stage 1 already triggers most required events (87.3% Trigger) but often produces forbidden consequences (60.3% Suppress). Stage 2 raises Suppress to 85.7% and Task to 64.4%, while Trigger and OnTime decrease slightly to 82.0% and 81.5%. Stage 3 improves every metric, raising Paired-CF from 67.6% to 97.3% and Task to 83.9%.

Spatial masks and object attributes. Table 6 reports the full grounding ablations on YouTube-VIS. Both ablations are applied to Stage 3. Removing the spatial mask reduces mIOU from 0.572 to 0.291 and worsens FVD from 62 to 79. Replacing every object caption with the generic word "object" reduces mIOU to 0.323 and rCLIP$_t$ from 0.291 to 0.273, while rCFC remains similar (0.965 and 0.963).

Dynamic status. On PairedStateBench, replacing dynamic status with static descriptions lowers Trigger from 92.2% to 42.5% and OnTime from 91.2% to 16.1%. Without dynamic status, the renderer receives no signal of whether or when each object changes, so the target events are often missing or mistimed.

D LIMITATIONS

By separating interaction reasoning from generative rendering, `BoxWorld` enables users to define object-level interaction rules and render their consequences in real time across diverse scenes and visual styles. Its current limitations mainly arise from the interface, the engine, and the training annotations. Although the Object-Centric State Interface, together with structured global prompts that describe the main objects, camera, environment, and interactions, covers most cases in our experiments, it may be insufficient for highly complex cases involving fine-grained geometry, articulated motion, or contact-rich interactions. Moreover, since the engine determines every interaction outcome, the rendered consequences are only as correct as the engine, and each new rule system requires its own engine implementation. The renderer is further trained on automatically annotated videos, where occasional annotation errors, such as objects missed by tracking, may lead to missing consequences or identity swaps. Future work can extend the interface with richer physical state and improve state annotation while keeping the interface lightweight and authorable.

E EXISTING ASSETS AND LICENSES

BoxWorld uses existing datasets, models, and evaluation tools cited in the main paper. We summarize the relevant public license or usage terms below.

Datasets and evaluation assets.

- VideoGen-1M (Tan et al., 2024) and 4DNeX (Chen et al., 2025b): public Hugging Face dataset pages list Apache-2.0 licenses.
- YouTube-VIS 2021 (Yang et al., 2021): YouTube-VIS annotations are licensed under CC BY 4.0 and the dataset is released for non-commercial research use.
- Online gameplay clips: used for research-only annotation and adaptation; raw clips are not redistributed because individual videos and games may have different platform licenses or publisher terms.
- VBench (Huang et al., 2024): the public evaluation repository is released under Apache-2.0.

Models and tools.

- Wan2.1 (Team Wan et al., 2025) and LongLive (Yang et al., 2025a): released under Apache-2.0.

- LingBot-World-v1 (Robbyant Team et al., 2026) and LingBot-World-v2 (Gao et al., 2026b): used only as evaluation baselines; released under Apache-2.0 and CC BY-NC-SA 4.0, respectively.
- Kimi K2.5 (Kimi Team et al., 2026): used to create Box World Engines and to evaluate PairedStateBench videos; released under a modified MIT license.
- Qwen3.8-27B (Qwen Team, 2026): used to evaluate PairedStateBench videos; released under Apache-2.0.
- LLaVA-NeXT-Video (Zhang et al., 2024b): used only as an off-the-shelf annotation tool; we do not redistribute its weights, and checkpoints may inherit different base-model licenses.
- Grounding DINO (Liu et al., 2025), SAM 2 (Ravi et al., 2025), and T5 (Raffel et al., 2020): used as annotation or text-encoding tools; Grounding DINO and SAM 2 are released under Apache-2.0, with SAM 2 including BSD-3-Clause terms for optional third-party components.