STATE WORLD MODELS: EXECUTE THE RULES, GENERATE THE PIXELS

Anonymous authors
Paper under double-blind review

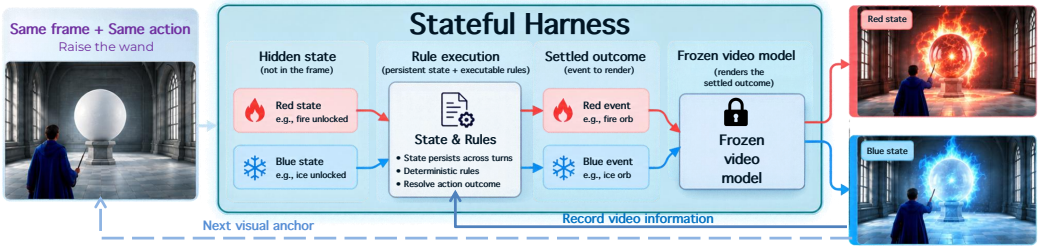


Figure 1: **State World Models with STATEFUL HARNESS.** The same initial frame and action can lead to different outcomes under different world states. The harness executes predefined rules to determine the event before a frozen video model renders it, illustrated by the red and blue elemental states. The generated clip provides the visual anchor for the next turn.

ABSTRACT

Existing world models can generate action-conditioned video from a single image. Many focus on camera movement, allowing users to explore the generated scene. However, navigation alone does not ensure that an action follows persistent rules when its outcome depends on state that is invisible in the current frame. We present **STATEFUL HARNESS**, a framework that separates world evolution from video rendering to support stateful and rule-governed interaction. It maintains an evolving world state under specified rules, renders each resulting event with a frozen video model, and uses the generated clip to update the world state for the next turn. At each turn, the harness resolves the action selected by the user and compiles the resulting event into an instruction, while a preview clip from an editable 3D scene provides spatial and camera guidance. To evaluate rule compliance, we build **WorldState-Bench**, a benchmark of 50 counterfactual twin pairs that share an initial frame and action but differ in hidden state and required outcome. Under rule-only prompting, the best of the 11 evaluated video backends produces the required endings for both cases in 32% of pairs. For MiniMax-H3, specifying the required endings raises this rate from 32% to 64%, while the full harness reaches 92% with the same frozen model. These results support separating rule execution from video rendering for world interactions with hidden state.

1 INTRODUCTION

Video generation has advanced rapidly, with recent models producing visually realistic videos with coherent motion (Wan et al., 2025, Wu et al., 2025). Building on this progress, interactive video world models condition generation on navigation signals, allowing users to move through and explore generated environments (Parker-Holder & Fruchter, 2025, Wang et al., 2026).

Navigation control makes a generated scene explorable, but does not determine the consequences of interacting with it (Parker-Holder & Fruchter, 2025, Wang et al., 2026). Some video world models support richer interactions by conditioning generation on player or agent actions (Valevski et al., 2025, Zhu et al., 2026a). However, a responsive video does not by itself ensure that an action has the outcome required by the world state and rules, or that its consequences persist across later interactions (Ding et al., 2026, Xu et al., 2026a). For example, the same action should open a locked chest only when the player holds a key, even if the initial views are identical (Figure 2).

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.

Such distinctions require *semantic state* to record variables such as inventory and progress, together with an addressable *spatial state* to preserve scene layout and camera paths across turns. While prior systems incorporate explicit state through programmable engines, learned game variables, or reconstructed NPC state (Huang et al., 2026, Lin et al., 2026, Deng et al., 2026), their evaluations focus on selected mechanics and do not establish whether a frozen video model can render outcomes prescribed by newly specified rules involving hidden state. Explicit state alone does not ensure that the consequence required by its rules appears in the generated video. The central challenge is therefore to make state transitions authoritative and render their consequences faithfully.

We present STATEFUL HARNESS, a framework that separates world evolution from video rendering to support stateful interaction under specified rules (Figure 1). It maintains an evolving world state under specified rules, renders each resulting event with a frozen video model, and uses the generated clip to update the world state for the next turn. Given a world description and an initial image, a coding agent defines the world state, available actions, and executable rules, while an editable 3D scene represents the spatial layout. The rules remain fixed as the world state evolves through user actions. At each turn, the user observes the current scene and selects an action, which the runtime resolves against the current state before rendering. The resolved event is compiled into a VisualScript instruction, and a preview clip from the updated 3D scene provides spatial and camera guidance. A frozen video model renders the event from these inputs and a visual anchor drawn from the initial image or preceding clip. A vision language model then interprets the generated clip and proposes updates to the semantic and spatial state based on the observed events. The runtime incorporates updates compatible with the fixed rules, and the resulting state determines the consequences of subsequent actions. Together, rule execution, video generation, and visual observation form a closed loop that yields *state world models*.

To evaluate whether an action yields a visible event consistent with hidden state, we construct WorldState-Bench with a Change-to-Comply protocol. Change assesses whether the action produces a discernible event, and Comply assesses whether that event satisfies an executable rule. Each of the 50 cases comprises a counterfactual pair with the same initial frame, action, and rule program but different hidden values that require distinct outcomes. We introduce PAIRFOLLOW and PAIRSTRICT as two metrics for joint evaluation of each counterfactual pair. PAIRFOLLOW requires both clips to show their required endings, whereas PAIRSTRICT additionally requires the forbidden ending to be absent from both clips. Among eleven unwrapped video backends, the highest PAIRFOLLOW score under the Rule-only condition is 32%. For MiniMax-H3, explicitly specifying the required ending increases PAIRFOLLOW from 32% to 64%, although PAIRSTRICT reaches only 4%. With MiniMax-H3, the complete STATEFUL HARNESS framework achieves 92% PAIRFOLLOW and 90% PAIRSTRICT on the same cases. Beyond the single-turn probe, the same closed loop carries state across longer episodes, as in the multi-turn duel of Figure 6, and extends training-free to worlds whose custom rules and state are written as a new world definition.

2 RELATED WORK

World Models. World models study action-conditioned predictions (Micheli et al., 2022, Hafner et al., 2025, Ha & Schmidhuber, 2018). The Genie series introduces latent actions and promptable environments (Bruce et al., 2024, Parker-Holder et al., 2024, Parker-Holder & Fruchter, 2025), while Oasis demonstrates interaction in an open world (Decart & Etched, 2024). Other systems expand scene range, streaming, and instruction following (He et al., 2025b, Wang et al., 2026, Robbyant Team et al., 2026, Yin et al., 2026, DreamX Team et al., 2026, Chen et al., 2026a), separate appearance from actions (Yu et al., 2025b), use game corpora (Che et al., 2025, Kanervisto et al., 2025, Guo et al., 2025, Yang et al., 2024a, Tang et al., 2025), or address memory and visual consistency (Xiao et al., 2025, Yu et al., 2025a, Li et al., 2025b, Hu et al., 2026, Garcin et al., 2026). Building on action-conditioned rendering, STATEFUL HARNESS uses executable rules to resolve each action against explicit semantic and spatial state before a frozen video model renders the resulting event.

Geometric Control of Video Generation. Camera-conditioned systems inject trajectories through learned pose or motion representations (Wang et al., 2024, Yang et al., 2024b, He et al., 2025a, Xu et al., 2024, Bahmani et al., 2025). Other methods reconstruct geometry for training-free control (Hou & Chen, 2024) or use boxes, source videos, and reference clips (Lin et al., 2023, Bai et al., 2025, Jiang et al., 2025). MoGe4D generates trajectories over recovered geometry (Zhang et al.,

2026b). The STATEFUL HARNESS uses a related interface to carry an editable scene across turns, with rule execution determining which spatial edits and semantic events are sent to the renderer.

Explicit State and Rules. Rules can be searched as generative objects or compiled from language (Salen & Zimmerman, 2003, Cook et al., 2013, Hsu et al., 2026). PWM couples an engine and a 3D oriented-box representation with conditioning at the pixel level (Huang et al., 2026). Code World Model compiles a code-maintained state into proxy video and fine-tunes its renderer (Chen et al., 2026b). Other work generates rule code without a video renderer (Lehrach et al., 2026), recovers state from generated frames (Deng et al., 2026), or conditions generation on selected state variables (Lin et al., 2026, Meng et al., 2026, Zhu et al., 2026a). Narrative and character state can also be generated together (Li et al., 2025a, Cheng et al., 2025). We combine a persistent editable scene, a semantic state ledger, and the standard reference inputs of a frozen renderer. This combination is minimal: new rule-governed worlds reduce to a small typed program, no engine or fine-tuning. Table 12 compares these interfaces and evaluation protocols.

World Modeling Evaluation. Most existing benchmarks measure visual quality, prompt adherence, navigation, subject actions, event editing, and perspective changes (Ying et al., 2026, Xu et al., 2026b,a, Zhao et al., 2026, Ding et al., 2026, Fang et al., 2026, Yang et al., 2026). Camera and geometry states can be tracked or remembered (Nam et al., 2026, Yu et al., 2025a, Li et al., 2025b). These measurements are valuable for navigability and continuity, but they do not directly test whether identical images and actions produce different rule-mandated outcomes when only hidden state changes. Concurrent work tests whether a shell-game arrangement is retained in the model's internal state across a swap chain (Shin et al., 2026); WorldState-Bench instead keeps the hidden state in an explicit rule program and tests whether the mandated outcome is executed in rendered video. Benchmarks inside executable games can compare behavior with engine state (Hafner, 2022, Bard et al., 2020, Light et al., 2023, Ouyang et al., 2026, La et al., 2026, Luo et al., 2026), but require an existing game or simulator. WorldState-Bench supplies an explicit rule program and tests its consequences in generated video.

3 METHOD: THE STATEFUL HARNESS

Framework Overview. A video model can depict an interaction, but a generated clip alone does not specify which changes should persist or whether those changes satisfy the rules of the world. Figure 2 shows how identical frames and actions require different outcomes when the hidden state differs. Inspired by the role of LLM harnesses, we place a stateful runtime around a frozen video model to separate rule execution, visual realization, and observation of the generated result.

Given a world description and an initial image, a coding agent constructs an executable game definition containing the state schema, available actions, rules, events, goals, and visual semantics. The harness initializes a semantic ledger and an editable 3D scene, while the game definition remains fixed throughout the episode. At each turn, the user observes the current scene and selects an action, which the runtime resolves against the current state and rules to determine the event and provisional state. The resolved event is compiled into a VisualScript instruction, and a preview clip from the updated 3D scene provides spatial and camera guidance. A frozen video model generates the next clip from these inputs and a visual anchor drawn from the initial image or preceding clip. A vision language model then interprets the generated events and proposes updates to the semantic and spatial state. The runtime checks these proposals against the fixed rules and incorporates compatible updates into the world state used to resolve the next action. The final frame also becomes the visual anchor for the next generation step. Together, rule execution, video generation, and visual observation form a closed loop that couples explicit world state with a frozen video model, yielding a *state world model*. Figure 3 illustrates world construction and the rule to render path.

We represent the evolving world state as $s_t = (z_t, u_t) \in \mathcal{S}_{\text{sem}} \times \mathcal{S}_{\text{sp}}$. The semantic component z_t records resources, entity attributes, progress, attitudes, and terminal status that may be invisible in the current frame. The spatial component u_t records an editable entity layout, scale, and camera path. A fixed rule program R determines which events an action permits and which state changes those events require. The harness updates the world state through a closed loop,

$$s_{t+1} = \mathcal{H}_R(s_t,\, a_t,\, o_{t+1}). \tag{1}$$

Here o_{t+1} denotes observations extracted by a vision language model from the generated clip, and $\mathcal{H}_R$ combines rule execution with compatible updates derived from these observations. The next



Figure 2: **Counterfactual Pairs Reveal the Role of Hidden State.** The two cases share the first frame I_0 and action a_t but differ in the hidden flag `has_key`. In the illustrated failure, an unwrapped video model opens the chest in both cases, violating the rule when no key is available. The harness maintains semantic and spatial state, executes the shared rule to determine whether the chest remains locked or opens, and renders each outcome with the same frozen video model.

state therefore incorporates both the consequences of the action and feedback from the generated video under the fixed rules R. The visual anchor f_t is initialized from the supplied image and subsequently updated from the final frame of each clip.

3.1 Stage 0: World Construction

World construction produces an executable game definition, an initial frame, a semantic ledger, and an editable spatial scene. The game definition fixes the rules for an episode, whereas the ledger and scene change as interactions unfold.

Executable Rules. From the world description, a coding agent writes the game definition in a small typed language. Each rule has a guard that reads the current state and action, and an effect that specifies ledger updates, spatial edits, and an event. Goal conditions inspect the ledger to determine whether an episode has ended. The agent validates and freezes the definition before interaction begins. In the counterfactual benchmark, both members of a pair share this definition.

Initial Frame and Spatial State. The benchmark supplies an audited first image shared by each counterfactual pair. For a new world, setup accepts a supplied image. A feed forward geometry model lifts the image into instance boxes and initializes u_0 as an untextured, editable 3D scene with a ground plane and labeled cuboids. We call this scene the *clay model*. Its explicit geometry supports object edits and procedural camera previews across turns.

Semantic State. The ledger is a serializable record of typed fields for resources, entity attributes, progress counters, attitudes, and terminal status. A twin specification declares the theme, twin identifier, and seeds. The initializer assigns different hidden values $z_A \neq z_B$ while preserving the same first frame and spatial layout. The rules therefore require different outcomes for the same visible action, even though rendering remains stochastic.

3.2 Stage 1: State Evolution and Video Generation

A turn begins when the user selects an action a_t after observing the current scene. In open interaction an intent agent can map a user instruction to a structured action. The runtime first settles the rule governed outcome, then compiles and renders it, and finally observes the generated clip for changes relevant to the next turn. This order gives fixed rules authority over action consequences while allowing generated events to inform the evolving world state.

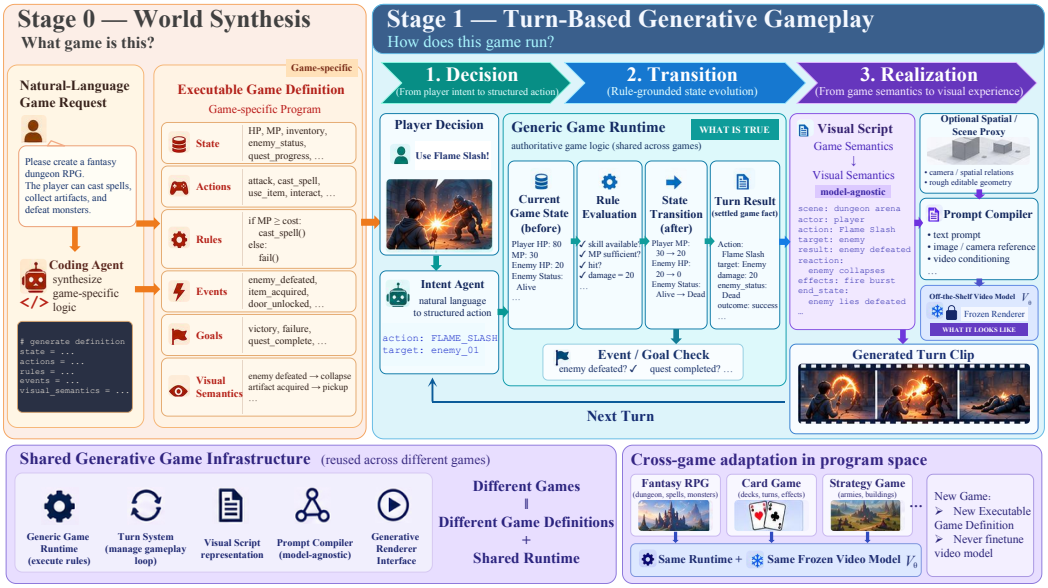


Figure 3: **The STATEFUL HARNESS.** In **Stage 0**, a coding agent constructs an executable game definition specifying the state, actions, rules, events, goals, and visual semantics. In **Stage 1**, the runtime evaluates the rules, updates the state, and determines the event and goal status for each action. The resolved event is compiled into an instruction for the frozen video model V_θ. Visual observations of the generated clip inform the state used in the next turn. Adaptation to a new game changes the game definition while retaining the shared runtime and video model weights.

Rule Settlement. The rule program evaluates its guards against s_t and a_t before video generation. It produces a provisional semantic state $\bar{z}_{t+1}$, a rule event e_t, and a spatial edit script ϵ_t. The edit script may add, move, or remove a box or change the camera path. The provisional spatial state is obtained by applying these edits to the clay model.

$$(\bar{z}_{t+1},\, \epsilon_t,\, e_t) \;=\; \tau_R(s_t,\, a_t), \qquad \bar{u}_{t+1} \;=\; \mathrm{apply}(u_t,\, \epsilon_t), \qquad \bar{s}_{t+1} \;=\; (\bar{z}_{t+1},\, \bar{u}_{t+1}). \tag{2}$$

An optional neural kernel ρ may propose dialogue or freeform narration for the settled event. Schema validation, value clamping, bounded retry, and fallback to the rule defined default constrain its proposals to the allowed set $\mathcal{A}_\tau(s, a)$. Thus the gated kernel $\mathcal{G} \circ \rho$ satisfies

$$\mathrm{Supp}\big((\mathcal{G} \circ \rho)(\cdot \mid s, a)\big) \;\subseteq\; \mathcal{A}_\tau(s, a). \tag{3}$$

The proposal may vary how an allowed event is expressed but cannot change the rule mandated transition in Eq. 2.

Spatial Guidance. The runtime applies ϵ_t to the clay model before rendering so that the preview represents the provisional spatial state. A procedural camera program renders this into a preview clip r_t, which is an editable representation of layout and cameras rather than the final appearance.

Instruction Compilation. The VisualScript compiler translates the settled state and event into a typed visual script with slots for actor, action, target, reaction, dialogue, effects, end state, and shots. A fixed compiler maps this record to the text instruction p_t, so identical inputs produce identical instructions. The text carries semantic facts that an oriented box stream cannot express, while the preview supplies layout and camera guidance. The instruction identifies the preview as a spatial reference to discourage the backend from reproducing its clay appearance.

Frozen Video Rendering. The video model receives the instruction as text, the preview clip as video reference, and the visual anchor f_t as image reference. It generates a clip v_{t+1} depicting the settled event, and the last frame becomes the anchor for the next turn.

$$(p_t,\, r_t) \;=\; \mathcal{C}(\bar{z}_{t+1},\, \bar{u}_{t+1},\, e_t), \qquad v_{t+1} \;=\; V_\theta(p_t,\, r_t,\, f_t), \qquad f_{t+1} \;=\; \mathrm{last}(v_{t+1}). \tag{4}$$

The model weights remain frozen, and the runtime uses the stock text, image, and video reference interfaces of a compatible backend. Normal operation makes one generation request per turn, with

bounded retries for transient errors. On a scene change, the anchor chain can be reset and appearance information is carried in the instruction instead.

Visual Observation and State Update. A vision language model examines the generated clip together with the settled event and provisional state. It identifies visible events and proposes changes to semantic or spatial fields that the video makes observable. The runtime admits a proposed change only when it is well formed and compatible with the fixed rule program. A visual detail may therefore become part of the next state, while an apparent event that contradicts the settled outcome is recorded as a rendering inconsistency and cannot overturn that outcome. We express this step as

$$o_{t+1} = O_\psi(v_{t+1}, \bar{s}_{t+1}, e_t), \qquad s_{t+1} = U_R(\bar{s}_{t+1}, o_{t+1}), \tag{5}$$

where O_ψ proposes observations and U_R validates and incorporates permitted updates. The next action is evaluated against s_{t+1}, closing the interaction loop without changing the rules R. The counterfactual evaluation in this paper tests rendered compliance.

4 BENCHMARK: WORLDSTATE-BENCH

Overview. Existing benchmarks evaluate visual quality, prompt adherence, navigation, subject actions, event editing, and camera or geometric consistency (Ying et al., 2026, Xu et al., 2026b,a, Zhao et al., 2026, Ding et al., 2026, Fang et al., 2026, Yang et al., 2026, Nam et al., 2026). These evaluations do not directly isolate whether a change in hidden state produces the distinct visible consequence prescribed by an executable rule. Benchmarks based on executable games assess behavior conditioned on state provided by existing engines (Hafner, 2022, Bard et al., 2020, Light et al., 2023, Ouyang et al., 2026, La et al., 2026, Luo et al., 2026). Worlds initialized from a description and an image lack this predefined engine state. To evaluate such worlds, WorldState-Bench uses counterfactual pairs with identical visible inputs, different hidden values, and expected outcomes determined by an executable rule before generation. Joint evaluation of each pair tests whether both required outcomes are rendered correctly under the corresponding hidden states.

We formalize this with the Change-to-Comply protocol. *Change* assesses whether an action produces a discernible event beyond camera motion. *Comply* assesses whether the event satisfies the rule for the specified hidden state. This separates the absence of an event from the generation of an event that contradicts the rule. The primary metrics quantify Comply, while clips with no discernible event or unjudgeable content are recorded as UNCLEAR and receive no credit.

Counterfactual Hidden-State Pairs. Each case contains two counterfactual instances, A and B, that share the same first frame and action script. They differ in one hidden value $z_A \neq z_B$ that is not observable in the shared frame. The same rule program requires different visible endings for the two instances. A correct ending in an isolated clip does not establish sensitivity to hidden state, whereas identical endings across a pair necessarily violate the required outcome for at least one instance. We therefore count a pair as compliant only when both clips show their required endings. A stricter criterion additionally requires the forbidden ending to be absent throughout both clips.

Case Construction. The benchmark comprises 50 cases drawn from 49 hidden variables and 21 scenario themes. Hidden values select which object, character, or elemental form carries the effect. Each hidden variable must be unobservable in the first frame and must select rule branches with visibly different outcomes. FLUX.2 (Black Forest Labs, 2025) generates the shared first frame at 1280×720, and each frame is audited for visual cues that reveal the hidden value. This audit forms part of benchmark construction. Authors specify each case program in a typed language that determines the initial states, required endings, and fixed choice judging questions. The harness executes the same rules used to derive the expected outcomes, so the evaluation measures whether the generated videos depict these outcomes.

We compile two instruction conditions from each case specification. *Rule-only* provides the original case instruction, hidden value, selected rule branch, and action. *Rule+end* additionally specifies the required terminal outcome in an explicit final state sentence. Table 1 reports results under these conditions, while the harness uses the VisualScript instruction described in §3.2.

Judging. We use the multimodal Qwen3.8-27B-VL (Yang et al., 2025) to answer fixed choice questions from sampled video frames. The questions identify the ending shown and determine whether the forbidden outcome appears at any point in the clip. At the clip level, ENDFOLLOW requires

Table 1: Main results on WorldState-Bench under Rule-only instructions (Track A) and Rule+end instructions (Track B). Bold indicates the best result and underline the best among the unwrapped baselines. The highlighted row reports STATEFUL HARNESS with text and first-frame reference inputs. Rows within a track share the probe, first frame, and judge; the matched comparison for the highlighted row is the same backbone under the Rule+end probe (Appendix Table 4), whose instruction wording the harness condition subsumes.

	A. Rule-only probe			B. Rule+end probe		
Backend	ENDFOLLOW↑	PAIRFOLLOW↑	TWINCONTRAST↑	STRICT↑	CNTABSENT↑	PAIRSTRICT↑
Image-to-video generators						
Wan2.2 A14B (Wan et al., 2025)	65 %	<u>32 %</u>	38 %	<u>54 %</u>	63 %	<u>26 %</u>
LongCat-Video 13.6B (LongCat Team et al., 2025)	47 %	8 %	24 %	46 %	62 %	8 %
HunyuanVideo 1.5 (Wu et al., 2025)	57 %	20 %	26 %	29 %	30 %	4 %
MiniMax-H3 (Chen et al., 2026a)	<u>66 %</u>	<u>32 %</u>	32 %	29 %	30 %	4 %
Interactive world models						
LingBot-World v2 (Robbyant Team et al., 2026)	48 %	16 %	38 %	44 %	76 %	14 %
Echo-WM Flash (Zhang et al., 2026a)	57 %	26 %	<u>42 %</u>	43 %	60 %	8 %
FantasyWorld (Dai et al., 2026)	34 %	8 %	30 %	35 %	81 %	8 %
AlayaWorld (AlayaWorld Team et al., 2026)	37 %	12 %	28 %	33 %	85 %	6 %
InSpatio-World (InSpatio Team et al., 2026)	19 %	2 %	14 %	17 %	92 %	4 %
Astra (Zhu et al., 2026b)	11 %	0 %	14 %	16 %	<u>96 %</u>	0 %
HY-WorldPlay (Sun et al., 2025)	9 %	0 %	4 %	7 %	94 %	0 %
STATEFUL HARNESS	**96 %**	**92 %**	**96 %**	**95 %**	**98 %**	**90 %**

the prescribed ending, CNTABSENT requires the absence of the forbidden outcome, and STRICT requires both conditions. At the pair level, PAIRFOLLOW and PAIRSTRICT require both clips to satisfy ENDFOLLOW and STRICT, respectively. TWINCONTRAST records whether the judged endings differ between the two clips. The judge receives the expected outcomes for the relevant rule branch, but neither the hidden variable nor the rule text.

5 EXPERIMENTS

5.1 EXPERIMENT SETUP

The rule compliance experiments use the 50 counterfactual pairs in WorldState-Bench, with 100 clips for individual evaluation and 50 pairs for joint evaluation (§4). Each backend generates one clip of a few seconds per action at its native resolution and frame rate. Clips contain 77 to 128 frames at 15 to 24 fps, with resolutions from 592×336 to 1280×720. The harness generates clips at 864×480 and 24 fps. We evaluate eleven existing backends without additional training or fine-tuning (Table 1). Four image-to-video generators receive instruction text and the initial frame, namely Wan2.2 (Wan et al., 2025), LongCat-Video (LongCat Team et al., 2025), HunyuanVideo 1.5 (Wu et al., 2025), and MiniMax-H3 (Chen et al., 2026a). Seven interactive world models receive inputs through their standard action interfaces, namely LingBot-World v2 (Robbyant Team et al., 2026), Echo-WM Flash (Zhang et al., 2026a), FantasyWorld (Dai et al., 2026), AlayaWorld (AlayaWorld Team et al., 2026), InSpatio-World (InSpatio Team et al., 2026), Astra (Zhu et al., 2026b), and HY-WorldPlay (Sun et al., 2025). The Qwen3.8-27B-VL judge evaluates frames sampled at 3 fps together with the final frame. Generation and judging use eight A800 80 GB GPUs.

5.2 RESULTS ON WORLDSTATE-BENCH

Under Rule-only prompting, MiniMax-H3 achieves 66% ENDFOLLOW and 32% PAIRFOLLOW, while Wan2.2 achieves 65% and 32%, respectively (Table 1). The lower joint accuracy indicates that correct outcomes in individual clips do not imply compliance across both hidden states. Astra, HY-WorldPlay, and InSpatio-World achieve less than 20% ENDFOLLOW despite high CNTABSENT scores. These results indicate that the absence of forbidden content can coincide with failure to generate the required event. Explicit outcome instructions under the Rule+end probe (Track B) improve ENDFOLLOW for several backends but do not establish reliable joint compliance. The highest unwrapped PAIRSTRICT score is 26% for Wan2.2, while MiniMax-H3 reaches only 4% PAIRSTRICT and 30% CNTABSENT. The reported harness condition resolves each action before generation and does not use compliance judgments to select among repeated video samples.

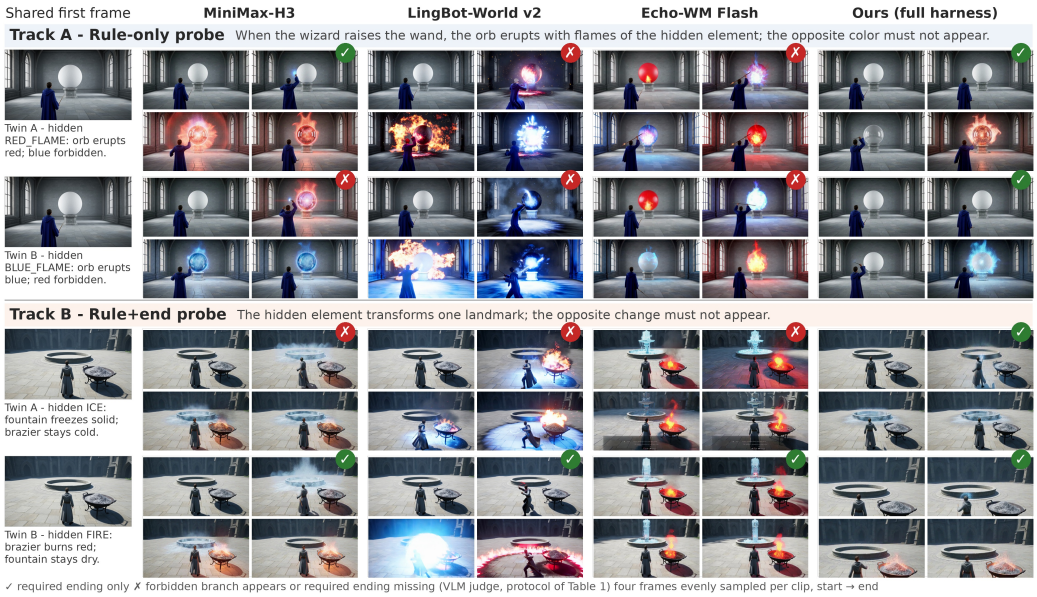


Figure 4: Rendered twin cases under the two probes. Each cell contains four evenly spaced frames from a 5 s clip. A ✓indicates that only the required ending appears, while a × indicates forbidden content or a missing required ending. The illustrated baselines reproduce the same ending across a pair or include forbidden content, whereas the harness generates the distinct required outcomes.

With the same frozen MiniMax-H3 weights, the harness reaches 92% PAIRFOLLOW, 90% PAIRSTRICT, and 96% TWINCONTRAST (Table 1). In the blind human comparison, STATEFUL HARNESS is preferred in 98% of pairwise comparisons against four widely-used unwrapped baselines, with no comparison won by a baseline (Figure 5, Appendix C).

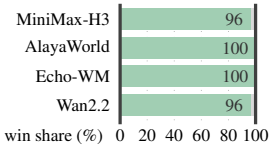


Figure 5: Human battles won by STATEFUL HARNESS

A component-chain decomposition on WorldState-Bench attributes the gain to outcome specification (+32 points) and to executed state with compiled references (+28 points), with both increments significant under paired exact McNemar tests over the 50 pairs.

5.3 QUALITATIVE RESULTS

Figure 4 shows counterfactual pairs in which unwrapped baselines include the forbidden outcome or reproduce an ending across both hidden states; Appendix Figures 13 and 14 add additional pairs. The harness generates the distinct outcomes specified by the rules. Figure 6 illustrates how settled state changes condition subsequent generation, using three real harness episodes read through their HUDs. The three episodes span 15, 29, and 11 turns respectively, showing that STATEFUL HARNESS sustains multi-turn interaction on the same frozen weights. The episodes are also three genres: town exploration, dating narrative, and turn-based duel, each with its own state transitions and rules. Each genre is a new executable game definition on the same runtime and video model, instantiated training-free. Appendix Figures 15, 16, and 17 unroll all turns of the three episodes.

5.4 ADDITIONAL EXPERIMENTS

External Benchmark. Table 2 compares the twenty models reported by Ying et al. (2026) and their class averages with two configurations using frozen MiniMax-H3 weights. These configurations use the official text protocol and the full harness with the spatial preview video, respectively. Video quality (78.0) and physical plausibility (68.8) are close to the corresponding class averages for text-driven models (77.4 and 66.9). Setting adherence increases from 71.1 under the official protocol to 87.2, and the interaction score of 77.7 is the highest reported in the comparison.

Channel Ablation. Table 8 compares reference channel configurations while holding the video model weights fixed. First-frame anchoring raises the full split from 74.2 to 76.1 and setting ad-

Table 2: WBench averages for the twenty models of Ying et al. (2026) plus our two entries (excluded from class averages). Text entries use the full split, camera/action the navigation split. Gray indicates class averages, and "–" denotes undefined values.

Metric group	Text-driven											Camera-controlled						Action-conditioned						Ours
	Seedance 1.5	Wan 2.7	Kling 3.0	YUME 1.5	HY-Video 1.5	LTX 2.3	LongCat-Video	Kairos 3.0	Cosmos 2.5	H3 (official)	*Average*	LingBot-World	HY-World 1.5	Fantasy-World	InSpatio-World	Astra	*Average*	Happy Oyster	Matrix-Game 3.0	Genie 3	Matrix-Game 2.0	HY-GameCraft	Infinite-World	*Average*	Ours
Video quality (6)	82.0	81.0	80.0	77.6	76.7	77.0	77.2	73.1	71.7	77.9	77.4	78.9	78.1	72.4	71.5	67.1	73.6	77.3	75.5	75.2	73.8	73.0	77.0	75.3	78.0
Setting (2)	82.9	91.4	91.0	72.4	85.6	85.2	72.3	70.3	83.3	71.1	81.6	72.6	72.2	71.3	71.4	59.6	69.4	74.2	63.6	72.5	67.1	66.6	69.3	68.9	87.2
Interaction (4)	67.9	71.7	72.8	48.2	54.6	49.0	44.8	41.4	43.2	68.9	54.8	–	–	–	–	–	–	–	–	–	–	–	–	–	77.7
Consistency (8)	79.9	75.8	83.9	80.9	87.5	78.1	86.6	83.2	86.3	83.8	82.5	89.9	86.9	86.4	88.4	73.3	85.0	84.3	74.5	82.6	65.1	72.6	80.0	76.5	77.8
Physical (2)	68.2	71.6	69.2	65.4	67.1	65.1	68.4	60.5	67.0	69.3	66.9	71.2	66.3	66.8	65.2	51.4	64.2	63.5	59.3	65.7	57.2	62.4	62.1	61.7	68.8

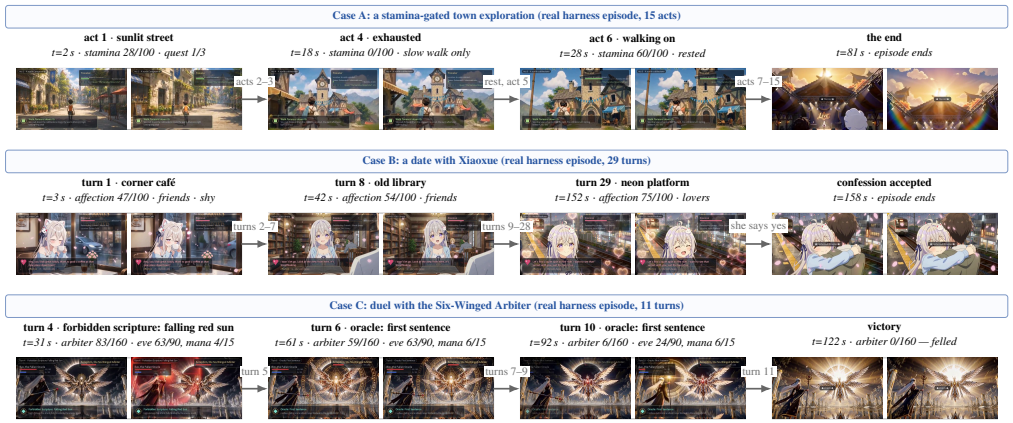


Figure 6: Turn sequences on the same frozen weights. All three rows are real harness episodes, with two adjacent frames per chunk at native aspect, each chunk headed by the video's own banner and HUD state (top: stamina and quest progress; middle: affection, relationship, mood; bottom: arbiter HP, eve HP, mana); elided turns are named on the arrows.

herence from 71.1 to 89.6, and adding the control video reaches 79.0 on navigation with interaction rising from 69.0 to 77.7, consistent with the preview's camera guidance; consistency dips from 83.8 to 75.3 and recovers to 77.8. Supplying the same control video under the raw text protocol yields 67.1 on the full split, improving camera trajectory scores; this transfer of the preview's geometry and appearance into the output motivates instructions that specify how the reference should be interpreted. Thirty generated clips illustrate this effect: clay appearance arises when the instruction does not identify the reference as a layout preview (§3.2). On our benchmark, the same frozen weights under the full harness improve every metric in Table 1 over the raw baseline of its track by 30 to 86 points: EndFollow from 66 to 96, PairFollow from 32 to 92, TwinContrast from 32 to 96, and PairStrict from 4 to 90.

6 Conclusion

Navigation control enables exploration of a generated scene. Counterfactual pairs test whether the same action produces the distinct outcomes required by different hidden states under shared rules. On WorldState-Bench, unwrapped image-to-video backends often fail this test even when the ending is specified in the prompt. The Stateful Harness separates rule execution from video rendering, with explicit semantic and spatial state determining the event depicted by a frozen video model. Visual observations of the generated clip inform the state used in subsequent interactions. With MiniMax-H3 weights held fixed, PairFollow increases from 32% under Rule-only instructions and 64% with explicit ending instructions to 92% with the full harness. These results support explicit state and executable rules for controlling stateful rendered interactions. The same recipe already sustains multi-turn episodes of 11 to 29 turns and extends to spatial triggers, where the rule program, not the renderer, absorbs the added bookkeeping.

AI USAGE STATEMENT

We used generative AI tools for manuscript writing assistance and for auxiliary tooling scripts, but not for research ideation, experiment design, or producing any reported result. AI models are also used as system components, for judging generated videos and for constructing game definitions. All AI-assisted output was reviewed by authors, who take full responsibility for the final content of the work.

REFERENCES

AlayaWorld Team, Kaipeng Zhang, Chuanhao Li, Yifan Zhan, Yongtao Ge, Yuanyang Yin, Jiaming Tan, Kang He, Liaoyuan Fan, Ruicong Liu, et al. Alayaworld: Long-horizon and playable video world generation. *arXiv preprint arXiv:2607.06291*, 2026.

Sherwin Bahmani, Ivan Skorokhodov, Aliaksandr Siarohin, Willi Menapace, Guocheng Qian, Michael Vasilkovsky, Hsin-Ying Lee, Chaoyang Wang, Jiaxu Zou, Andrea Tagliasacchi, et al. Vd3d: Taming large video diffusion transformers for 3d camera control. In *International Conference on Learning Representations*, volume 2025, pp. 66712–66737, 2025.

Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14834–14844. IEEE, 2025.

Nolan Bard, Jakob N. Foerster, Sarath Chandar, et al. The Hanabi challenge: A new frontier for AI research. *Artificial Intelligence*, 280:103216, 2020. doi: 10.1016/j.artint.2019.103216.

Black Forest Labs. FLUX.2. https://blackforestlabs.com, 2025.

Blender Online Community. Blender – a 3D modelling and rendering package. https://www.blender.org, 2025.

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

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, et al. SAM 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

Haoxuan Che, Xuanhua He, Quande Liu, Cheng Jin, and Hao Chen. Gamegen-x: Interactive open-world game video generation. In *International Conference on Learning Representations*, volume 2025, pp. 37546–37593, 2025.

Danze Chen, Zeqing Wang, Ziyue Lin, Xingyi Yang, and Yeying Jin. H3-World: Turning language understanding into world control. *arXiv preprint arXiv:2609.01560*, 2026a. URL https://arxiv.org/abs/2609.01560v1. Version 1.

Yiwen Chen, Guosheng Lin, and Chi Zhang. Code world model: Coding agent as world brain. *arXiv preprint arXiv:2608.25927*, 2026b. URL https://arxiv.org/abs/2608.25927.

Junhao Cheng, Yuying Ge, Yixiao Ge, Jing Liao, and Ying Shan. Animegamer: Infinite anime life simulation with next game state prediction. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10875–10885. IEEE, 2025.

Michael Cook, Simon Colton, Azalea Raad, and Jeremy Gow. Mechanic miner: Reflection-driven game mechanic discovery and level design. In *Applications of Evolutionary Computation (EvoApplications)*, volume 7835 of *Lecture Notes in Computer Science*, pp. 284–293, 2013. doi: 10.1007/978-3-642-37192-9_29.

Yixiang Dai, Fan Jiang, Chiyu Wang, Mu Xu, and Yonggang Qi. Fantasyworld: Geometry-consistent world modeling via unified video and 3d prediction. In *International Conference on Learning Representations*, volume 2026, pp. 103603–103622, 2026.

Decart and Etched. Oasis: A universe in a transformer. Technical report and demonstration, 2024. URL https://oasis-model.github.io/.

Zhiyang Deng, Boran Zhang, Danze Chen, and Yeying Jin. WorldMind: Decoupled game world model for state-aware NPC behavior. *arXiv preprint arXiv:2608.21439*, 2026. URL https://arxiv.org/abs/2608.21439.

Kaixin Ding, Xi Chen, Minghong Cai, et al. PlayWorld: Benchmarking world models with agent players over long-horizon objectives. *arXiv preprint arXiv:2608.13552*, 2026. URL https://arxiv.org/abs/2608.13552v2. Version 2. Tables 2–3.

DreamX Team, Yancheng Bai, Rui Chen, Xiangxiang Chu, Rujing Dang, Hao Dou, Bingjie Gao, Qiwen Gu, Siyu Hong, Jiachen Lei, Geng Li, Jifan Li, Ruimin Lin, Qingfeng Shi, Bingze Song, Lei Sun, Jing Tang, Ruitian Tian, Jun Wang, Jiahong Wu, Pengfei Zhang, Shen Zhang, and Jiashu Zhu. DreamX-World 1.0: A general-purpose interactive world model. *arXiv preprint arXiv:2606.16993*, 2026. URL https://arxiv.org/abs/2606.16993.

Jianjie Fang, Yingshan Lei, Qin Wan, Ziyou Wang, Yuchao Huang, Yongyan Xu, Baining Zhao, Weichen Zhang, Chen Gao, Xinlei Chen, et al. iworld-bench: A benchmark for interactive world models with a unified action generation framework. *arXiv preprint arXiv:2605.03941*, 2026.

Samuel Garcin, Thomas Walker, Steven McDonagh, Tim Pearce, Hakan Bilen, Tianyu He, Kaixin Wang, and Jiang Bian. Beyond pixel histories: World models with persistent 3D state. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2603.03482.

Junliang Guo, Yang Ye, Tianyu He, et al. MineWorld: A real-time and open-source interactive world model on Minecraft. *arXiv preprint arXiv:2504.08388*, 2025. URL https://arxiv.org/abs/2504.08388.

David Ha and Jürgen Schmidhuber. World models. *CoRR*, abs/1803.10122, 2018. URL https://arxiv.org/abs/1803.10122.

Danijar Hafner. Benchmarking the spectrum of agent capabilities. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2109.06780.

Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models. *arXiv preprint arXiv:2509.24527*, 2025. URL https://arxiv.org/abs/2509.24527.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling camera control for text-to-video generation. In *International Conference on Learning Representations*, 2025a. URL https://arxiv.org/abs/2404.02101.

Xianglong He, Chunli Peng, Zexiang Liu, et al. Matrix-game 2.0: An open-source real-time and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025b. URL https://arxiv.org/abs/2508.13009.

Chen Hou and Zhibo Chen. Training-free camera control for video generation. *arXiv preprint arXiv:2406.10126*, 2024.

Ting-Chen Hsu, Lianye Zhang, Jiangxu Lin, Zhaoyi Yu, Fei Qin, and Zihao Chen. If: Cargo: Llm-based semantic compilation for al-native rule programming games. *arXiv preprint arXiv:2608.12195*, 2026.

Anthony Hu, Václav Volhejn, Adrien Ramanana Rahary, et al. Multiplayer interactive world models with representation autoencoders. *arXiv preprint arXiv:2607.05352*, 2026. URL https://arxiv.org/abs/2607.05352.

Zheng-Hui Huang, Guixu Lin, Jiacheng Lin, Yi-Chuan Huang, Ruihan Yu, Muyao Niu, Siqi Yang, Yu-Lun Liu, Yung-Yu Chuang, Kaipeng Zhang, and Zhixiang Wang. Programmable world model. *arXiv preprint arXiv:2609.10540*, 2026. URL https://arxiv.org/abs/2609.10540v1. Version 1.

InSpatio Team, Donghui Shen, Guofeng Zhang, Haomin Liu, Haoyu Ji, Hujun Bao, Hongjia Zhai, Jialin Liu, Jing Guo, Nan Wang, et al. Inspatio-world: A real-time 4d world simulator via spatiotemporal autoregressive modeling. *arXiv preprint arXiv:2604.07209*, 2026.

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

Anssi Kanervisto, Dave Bignell, Linda Yilin Wen, et al. World and human action models towards gameplay ideation. *Nature*, 638:656–663, 2025. doi: 10.1038/s41586-025-08600-3. URL https://www.nature.com/articles/s41586-025-08600-3.

Brian La, Sejoon Chang, Ben Kim, Junyoung Bae, Aamish Ahmad Beg, Sei Chang, Gonzalo Gonzalez-Pumariega, and Kanav Goyal. Gameenginebench: Evaluating coding agents on real c++ runtime environments. *arXiv preprint arXiv:2607.03525*, 2026. URL https://arxiv.org/abs/2607.03525.

Wolfgang Lehrach, Daniel Hennes, Miguel Lazaro-Gredilla, Xinghua Lou, Carter Wendelken, Zun Li, Antoine Dedieu, Marc Lanctot, Atil Iscen, John Schultz, et al. Code world models for general game playing. In *International Conference on Learning Representations*, volume 2026, pp. 133870–133927, 2026.

Jialu Li, Yuanzhen Li, Neal Wadhwa, Yael Pritch, David E Jacobs, Michael Rubinstein, Mohit Bansal, and Nataniel Ruiz. Unbounded: A generative infinite game of character life simulation. In *International Conference on Learning Representations*, volume 2025, pp. 87465–87491, 2025a.

Runjia Li, Philip Torr, Andrea Vedaldi, and Tomas Jakab. Vmem: Consistent interactive video scene generation with surfel-indexed view memory. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25690–25699. IEEE, 2025b.

Jonathan Light, Min Cai, Sheng Shen, and Ziniu Hu. Avalonbench: Evaluating llms playing the game of avalon. *arXiv preprint arXiv:2310.05036*, 2023. URL https://arxiv.org/abs/2310.05036.

Han Lin, Abhay Zala, Jaemin Cho, and Mohit Bansal. VideoDirectorGPT: Consistent multi-scene video generation via LLM-guided planning. In *Conference on Language Modeling (COLM)*, 2023. URL https://arxiv.org/abs/2309.15091.

Zijun Lin, Zeqing Wang, Cheston Tan, Bihan Wen, and Yeying Jin. StatePlay: State-aware game world models for mechanics-consistent generation. *arXiv preprint arXiv:2607.26754*, 2026. URL https://arxiv.org/abs/2607.26754.

LongCat Team, Xunliang Cai, Qilong Huang, Zhuoliang Kang, Hongyu Li, Shijun Liang, Liya Ma, Siyu Ren, Xiaoming Wei, Rixu Xie, and Tong Zhang. Longcat-video technical report, 2025. URL https://arxiv.org/abs/2510.22200.

Tongxu Luo, Rongsheng Wang, Jiaxi Bi, et al. Gamecraft-bench: Can agents build playable games end-to-end in a real game engine? In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2606.17861.

Zian Meng, Zhen Li, Chuanhao Li, Qiang Li, and Kaipeng Zhang. Marionette: Predicting world states, rendering geometry, painting appearance. *arXiv preprint arXiv:2608.14530*, 2026. URL https://arxiv.org/abs/2608.14530.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2209.00588.

Jisu Nam, Yicong Hong, Chun-Hao Paul Huang, et al. Worldcam: Interactive autoregressive 3d gaming worlds with camera pose as a unifying geometric representation. *arXiv preprint arXiv:2603.16871*, 2026. URL https://arxiv.org/abs/2603.16871.

Mingyu Ouyang, Siyuan Hu, Kevin Qinghong Lin, Hwee Tou Ng, and Mike Zheng Shou. Gameworld: Towards standardized and verifiable evaluation of multimodal game agents. *arXiv preprint arXiv:2604.07429*, 2026. URL https://arxiv.org/abs/2604.07429.

Jack Parker-Holder and Shlomi Fruchter. Genie 3: A new frontier for world models. Google DeepMind technical announcement, 2025. URL https://deepmind.google/blog/genie-3-a-new-frontier-for-world-models/.

Jack Parker-Holder, Philip Ball, Jake Bruce, et al. Genie 2: A large-scale foundation world model. Google DeepMind technical announcement, 2024. URL https://deepmind.google/blog/genie-2-a-large-scale-foundation-world-model/.

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. URL https://arxiv.org/abs/2601.20540.

Katie Salen and Eric Zimmerman. *Rules of Play: Game Design Fundamentals*. The MIT Press, 2003. ISBN 9780262240451.

Joonghyuk Shin, Yicong Hong, Jaesik Park, and Xun Huang. Can video world models track unobserved world states? *arXiv preprint arXiv:2608.30692*, 2026.

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

Junshu Tang, Jiacheng Liu, Jiaqi Li, Longhuang Wu, Haoyu Yang, Penghao Zhao, Siruis Gong, Xiang Yuan, Shuai Shao, Linfeng Zhang, and Qinglin Lu. Hunyuan-GameCraft-2: Instruction-following interactive game world model. *arXiv preprint arXiv:2511.23429*, 2025. URL https://arxiv.org/abs/2511.23429.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2408.14837.

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

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

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*, pp. 1–11, 2024.

Zile Wang, Zexiang Liu, Jiaxing Li, et al. Matrix-Game 3.0: Real-time and streaming interactive world model with long-horizon memory. *arXiv preprint arXiv:2604.08995*, 2026. URL https://arxiv.org/abs/2604.08995.

Bing Wu, Chang Zou, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Jack Peng, Jianbing Wu, Jiangfeng Xiong, Jie Jiang, et al. Hunyuanvideo 1.5 technical report. *arXiv preprint arXiv:2511.18870*, 2025.

Zeqi Xiao, Yushi Lan, Yifan Zhou, et al. WorldMem: Long-term consistent world simulation with memory. *arXiv preprint arXiv:2504.12369*, 2025. URL https://arxiv.org/abs/2504.12369.

Dejia Xu, Weili Nie, Chao Liu, Sifei Liu, Jan Kautz, Zhangyang Wang, and Arash Vahdat. Camco: Camera-controllable 3d-consistent image-to-video generation. *arXiv preprint arXiv:2406.02509*, 2024. URL https://arxiv.org/abs/2406.02509.

Ting-Bing Xu, Jiacheng Sui, Zhe Gao, Kewei Shi, Wenjin Yang, Zhicheng Liu, Zhaoxu Sun, Mingchao Sun, Hongyu Pan, Fan Jiang, Mu Xu, Qi Fan, Yang Gao, Yong Li, and Baoquan Chen. WorldRoamBench: An open-world benchmark for long-horizon stability of interactive world models. *arXiv preprint arXiv:2606.31672*, 2026a. URL https://arxiv.org/abs/2606.31672. Version 5. Version 2 appeared under the title WorldOdysseyBench.

Xiaojie Xu, Zhengyuan Lin, Kang He, Yukang Feng, Xiaofeng Mao, Yuanyang Yin, Yongtao Ge, and Kaipeng Zhang. WorldMark: A unified benchmark suite for interactive video world models. *arXiv preprint arXiv:2604.21686*, 2026b. URL https://arxiv.org/abs/2604.21686v2. Version 2.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Mingyu Yang, Junyou Li, Zhongbin Fang, Sheng Chen, Yangbin Yu, Qiang Fu, Wei Yang, and Deheng Ye. Playable Game Generation, 2024a. URL https://arxiv.org/abs/2412.00887v1.

Shiyuan Yang, Liang Hou, Haibin Huang, Chongyang Ma, Pengfei Wan, Di Zhang, Xiaodong Chen, and Jing Liao. Direct-a-video: Customized video generation with user-directed camera movement and object motion. In *ACM SIGGRAPH 2024 Conference Papers*, 2024b. URL https://arxiv.org/abs/2402.03162.

Yuxue Yang, Shuyao Shang, Jiahe Wang, et al. Worldexam: Benchmarking world models from apparent appearance to inherent reactivity. *arXiv preprint arXiv:2608.02603*, 2026. URL https://arxiv.org/abs/2608.02603.

Yuanyang Yin, Gongxuan Wang, Yifan Zhan, Chuanhao Li, Kaipeng Zhang, and Feng Zhao. Alaya-EVOKE: From linear-scaling supervision to endless world. *arXiv preprint arXiv:2608.13546*, 2026. URL https://arxiv.org/abs/2608.13546v2. Version 2.

Kaining Ying, Hengrui Hu, Siyu Ren, Jiamu Li, Fengjiao Chen, Ziwen Wang, Xuezhi Cao, Xunliang Cai, and Henghui Ding. WBench: A comprehensive multi-turn benchmark for interactive video world model evaluation. *arXiv preprint arXiv:2605.25874*, 2026. URL https://arxiv.org/abs/2605.25874.

Jiwen Yu, Jianhong Bai, Yiran Qin, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Context as memory: Scene-consistent interactive long video generation with memory retrieval. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025a.

Jiwen Yu, Yiran Qin, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Gamefactory: Creating new games with generative interactive videos. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11590–11599. IEEE, 2025b.

Z.ai. GLM-5.3. https://github.com/zai-org/GLM-5, 2026.

Songchun Zhang, Yaowei Li, Junhao Zhuang, Weiyang Jin, Haoyu Wang, Xin Lu, Yilang Sun, Shiyi Zhang, Haoran Li, Xiaoxiao Ma, Yuming Li, Yijun Liu, Yaofeng Su, Yanwen Ma, Haoyu Wu, Zihan Su, Yue Ma, Lvmin Zhang, Haoyang Huang, Zeyue Xue, Anyi Rao, and Nan Duan. Echowm: Open and enterable omnimodal world models. *arXiv preprint arXiv:2608.23189*, 2026a. URL https://arxiv.org/abs/2608.23189.

Yanran Zhang, Ziyi Wang, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Geometry-aware single-image 4d synthesis via dense trajectory generation. In *European Conference on Computer Vision*, pp. 95–114. Springer, 2026b.

Yuke Zhao, Wangbo Zhao, Weijie Wang, Zeyu Zhang, Dakai An, Akide Liu, Yinghao Yu, Jiasheng Tang, Fan Wang, Wei Wang, and Bohan Zhuang. WorldOlympiad: Can your world model survive a triathlon? *arXiv preprint arXiv:2606.11129*, 2026. URL https://arxiv.org/abs/2606.11129.

Shangwen Zhu, Qianyu Peng, Zhao Pu, Zhilei Shu, Xiangrui Ke, Zhaohu Xing, Zizhao Tong, Zeqing Wang, Xinyu Cui, Zian Zheng, Huangji Wang, Jian Zhao, Yeying Jin, Fan Cheng, and Ruili Feng. Incantation: Natural language as the action interface for multi-entity video world models. *arXiv preprint arXiv:2605.18601*, 2026a. URL https://arxiv.org/abs/2605.18601. Version 2, revised July 12, 2026.

Yixuan Zhu, Jiaqi Feng, Wenzhao Zheng, Yuan Gao, Xin Tao, Pengfei Wan, Jiwen Lu, and Jie Zhou. Astra: General interactive world model with autoregressive denoising. In *International Conference on Learning Representations*, volume 2026, pp. 79167–79184, 2026b.

Appendix

A WorldState-Bench: Case Construction Details 16
B Judging Protocol and Metric Definitions 18
C Human Audit Platform and Protocols 18
D Additional Results 19
E Additional Qualitative Examples 25
F Method Details: Formal Framework, Turn Lifecycle, and Domain Adaptation 31
G Positioning against Concurrent Explicit-State Systems 35
H Reproducibility Details 35
I Limitations and Future Work 35

A WorldState-Bench: Case Construction Details

Case Anatomy. Each case comprises five components. These components are a theme and scene description, a first frame generated by FLUX.2 at 1280×720, a hidden variable declaration $\texttt{name} \in \{v_A, v_B\}$, an action script, and rule text specifying distinct endings. Twins A/B share the theme, frame, action, and rules. They differ in the hidden value. The suite spans 21 themes (magic academy, dungeon, open-world exploration, dating, visual novel, museum, survival, escape room, sailing, farming, ranch, camping, shopkeeping, card games, board games, detective, turn-based combat, stealth, healing, cooking, and fishing) and 49 distinct hidden variables.

Hidden-Variable Families. The hidden values fall into three response types: *mechanism selection*, which object or mechanism responds (42 cases, e.g., a gate mode that raises the stone door versus a brazier mode that lights tall flames); *character response*, whether a character or creature responds instead of an object (6 cases, e.g., sheep gathering at a trough, an NPC sitting down); and *form selection*, which elemental form or scene side carries the effect (2 cases, e.g., red versus blue flames, the left versus the right brazier). All are invisible in the frame by construction and by audit.

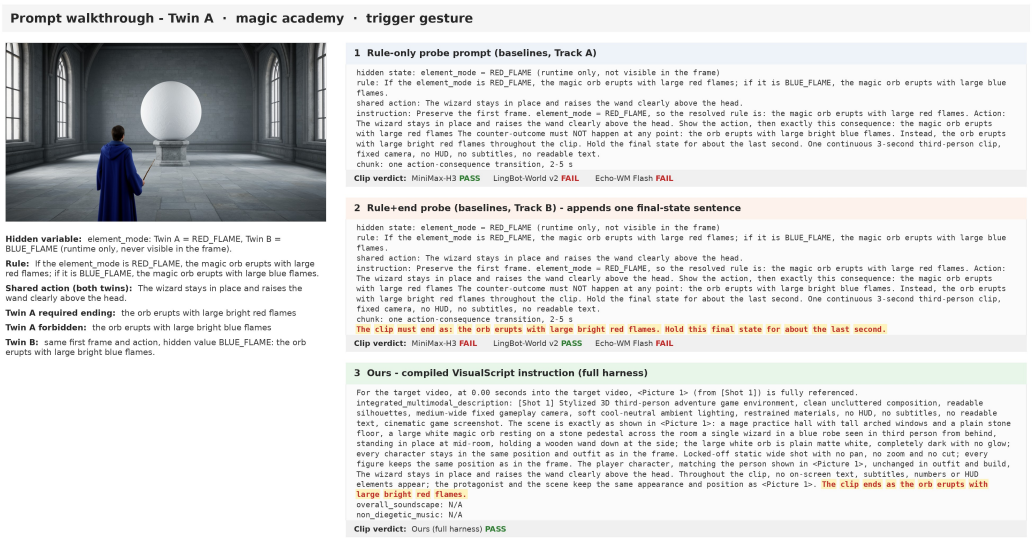
Trigger Types. Cases also differ in how the rule trigger is expressed, with *cast* covering verbal or ritual actions (20 clips), *gesture* covering physical actions (42), *mechanism* covering object interactions (26), and *zone* covering entry into or use of a spatial region (12). Zone-triggered cases are the most demanding (§5.2) because compliance additionally requires the render to make the zone's state visible. This motivates the spatial channel.

Instruction Probes. Every case is evaluated under two probes compiled from the same card. *Rule-only* provides the original "Inputs to the video model" text from the case specification, including the scene, shared action script, rule with both branches, and the consequence of the selected branch. *Rule+end* appends the required terminal state using the visual vocabulary defined for the case and explicitly asks the model to maintain this final state. The difference between the two conditions measures the effect of adding an explicit terminal outcome instruction. Table 5 reports this difference for each backend.

First-Frame Audit. Each first frame undergoes eight rounds of VLM auditing across six dimensions to check for visible evidence of the hidden value. The dimensions are object presence, color attribution, text, spatial cues, count cues, and state cues. If the audit detects evidence of the hidden value, the frame is regenerated using a revised prompt. The audit prompts, model settings, and the regeneration log are part of the reproduction artifacts listed in Appendix H.

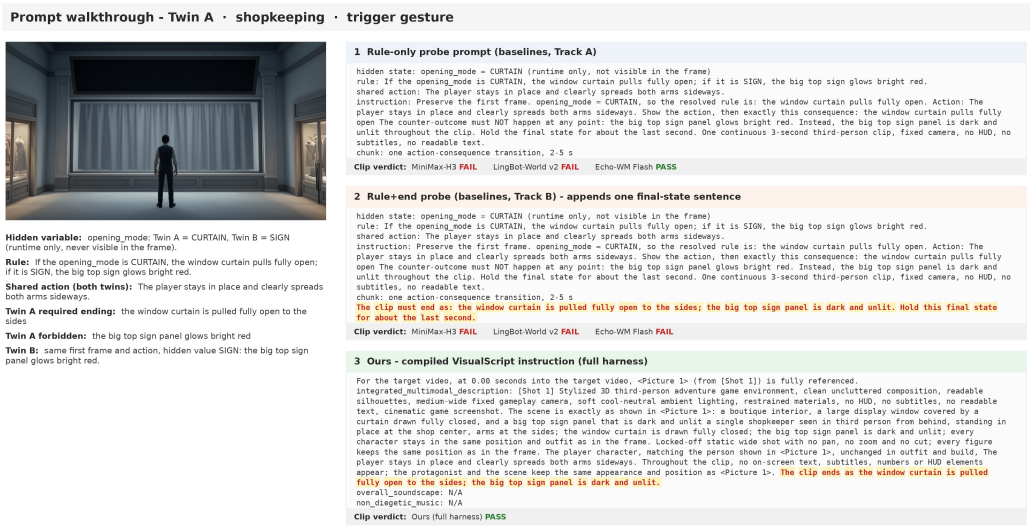
Rule Text. Rules are written in plain English with fixed sentence frames per family, e.g. "If the flame is RED, the ward shatters and the door opens. If the flame is BLUE, the ward holds and the door stays sealed." Both rule branches specify their endings so that the DSL can derive expected outcomes directly.

Figures 7 and 8 present complete examples for the orb pair under Rule-only instructions and the sign pair under Rule+end instructions, respectively. Each example includes the audited first frame, case specification, two baseline prompts, compiled VisualScript instruction, and the rule program with its settled state.



Rule Program and Settled State. For twin A, the hidden value is `element_mode = RED_FLAME`, while twin B uses `BLUE_FLAME`. Neither value is visible in the first frame. The RED_FLAME branch sets `orb.flame := red` and forbids blue flames, while the BLUE_FLAME branch sets `orb.flame := blue`. The ledger for twin A changes from `orb.flame = none` to `orb.flame = red` through the corresponding guarded update.

Figure 7: Prompt example for the orb pair, twin A. The panels show the Rule-only prompt, the Rule+end prompt with its appended outcome instruction highlighted, and the compiled VisualScript instruction. The annotations below the panels describe the rule program and settled state.



Rule Program and Settled State. For twin A, the hidden value is `opening_mode = CURTAIN`, while twin B uses `SIGN`. Neither value is visible in the first frame. The CURTAIN branch requires the curtain to open fully and forbids the upper sign from glowing red, while the SIGN branch requires the sign to glow red. For twin A, the guarded update opens the curtain and preserves the unlit sign.

Figure 8: Prompt example for the sign pair, twin A, using the layout of Figure 7. Highlighted clauses identify the appended Rule+end sentence and the ending specified by the compiled instruction. The annotations below the panels show the corresponding rule program and settled state.

B JUDGING PROTOCOL AND METRIC DEFINITIONS

Fixed-Choice QA. For each clip the judge answers two scored questions with fixed choices: (Q1) which ending occurs at the end of the clip, with mandated, forbidden, and unclear as the choices. (Q2) asks whether the forbidden outcome appears anywhere in the clip, with yes and no as the choices. The case program specifies both outcomes. Initial-state and action-onset questions are also recorded but excluded from the primary metrics. Answers outside the choice set or missing answers score as incorrect. Fixed answer choices constrain the output format and make verdicts reproducible; the accuracy of the judge's visual interpretation is audited against blinded human raters in Appendix C.

Judge Inputs. The judge receives sampled frames, the shared scene description, the action, and the branch-specific expected outcomes. The hidden value, rule program, and expected outcome of the other instance are excluded from the judge inputs. Reproducing a verdict requires the same clips, sampled frames, questions, judge version, and inference settings. The case program fixes the expected outcomes before video generation.

Independent Validation. The blind human audits in Appendix C assess clip correctness and pairwise preference independently of system identity. That appendix reports agreement between the human ratings and VLM judgments. The reported metrics are therefore produced by a fixed, versioned judge protocol and are validated against blinded human ratings on the audited subset.

B.1 METRIC DEFINITIONS

All metrics are percentages. Clip-level metrics cover 100 units. An up arrow indicates that higher is better, and a down arrow indicates that lower is better.

- ENDFOLLOW $\uparrow$ requires the final frames to depict the ending specified for that instance.
- CNTABSENT $\uparrow$ requires the forbidden outcome, defined as the required ending for the other instance, to be absent throughout the clip.
- STRICT $\uparrow$ (primary): ENDFOLLOW $\wedge$ CNTABSENT. The mandated outcome is the only salient ending shown (no contradictory outcome, no both-branch montage).
- UNCLEAR $\downarrow$: output unjudgeable (degenerate render, no discernible event). This is counted separately and never credited.

Pair-level metrics (over 50 pairs):

- PAIRFOLLOW (twin-joint): both twins individually ENDFOLLOW-correct.
- PAIRSTRICT (twin-strict, headline): both twins individually STRICT-correct (twin-joint scoring, §4).
- TWINCONTRAST $\uparrow$ indicates that the judged endings differ between instances A and B. This indicates that the pair is distinguishable and helps diagnose copy-paste behavior.

Probe-contrast metric: ΔSTRICT = STRICT(Rule+end) $-$ STRICT(Rule-only). This measures the effect of dictating the ending in addition to the rule instruction. Derived diagnostics (analysis only, Appendix D): the failure triad {direction error (wrong branch taken), leakage-only (right direction but forbidden content also shown), double error}. Branch gap $= |\text{strict}_A - \text{strict}_B|$.

C HUMAN AUDIT PLATFORM AND PROTOCOLS

We assess agreement between the VLM judge and human raters through audits conducted on a web platform (Appendix B). Each auditor receives a numeric identifier and a fixed assignment of evaluation items. The interface displays rendered clips and rule text while concealing system identities, prompts, and instruction conditions. The positions of systems in pairwise comparisons are randomized. The platform operates in two modes, mirroring the two question types of the benchmark.

Clip Verification. Seventeen cases, comprising nine Rule-only cases and eight Rule+end cases, are evaluated for both hidden states using the full harness, the frozen MiniMax-H3 baseline, and Wan2.2. This produces 102 clips for verification. For each clip, auditors answer the same two fixed choice questions used by the VLM judge. Q1 assesses whether the required ending occurs, and Q2 assesses whether the forbidden outcome appears anywhere in the clip. Both questions include an explicit "unclear" option in the human interface. Each clip receives three independent blinded ratings. Figure 9 shows the clip-verification interface.

Battle Preference. Twenty-five further case pairs are held out for pairwise comparison. Each pairwise comparison evaluates the full harness against one of four baselines under Rule-only instructions, namely frozen MiniMax-H3, AlayaWorld, Echo-WM, and Wan2.2. The four clips, covering hidden states A and B for each system, play simultaneously in a grid at 2× speed. Each comparison receives two independent ratings through a forced choice between the left and right systems, without a tie option. Figure 10 shows the battle grid.

Judge Validation. Under the same blind evaluation protocol, the VLM judge agrees with the human majority on 91.2 % of the 102 verification clips for Q1 and 86.3 % for Q2. Both rates approach the human reference agreement of 90.2 % and 80.7 %, respectively. The corresponding agreement coefficient is $\kappa = 0.72/0.72$. System scores show strong rank correlation with human scores on STRICT (Spearman $\rho = 0.94$) and CNTABSENT ($\rho = 1.00$).

Battle Results. Table 3 reports the human win rate of our full harness per opponent under forced choice. The harness is preferred in 98 % of comparisons, with rates ranging from 96 % to 100 % across baselines. No comparison favors a baseline under the aggregate decision rule, and the remaining comparisons have split auditor votes. At the individual vote level, 99.0 % of votes favor the harness, and agreement between auditor pairs ranges from 95 % to 100 %.

Table 3: Human preference for the full harness in pairwise comparisons with each baseline. Each comparison receives two ratings using randomized positions and forced choice without a tie option. The remaining comparisons contain split auditor votes, and none favors a baseline under the aggregate decision rule.

Opponent (Rule-only)	Ours win	Split
MiniMax-H3 baseline	96 %	4 %
AlayaWorld	100 %	0 %
Echo-WM Flash	100 %	0 %
Wan2.2 A14B	96 %	4 %
All opponents	98 %	2 %

D ADDITIONAL RESULTS

The Full Both-Probe Grid. Table 4 reports every metric under both instruction probes for all eleven backends, followed by the deployed harness condition. The harness is reported as a single condition: its compiler derives the rule branch and the required ending from the case program, so the compiled instruction is the same regardless of probe wording. Table 1 selects one probe per track from this grid. The grid also yields ΔSTRICT, the difference between Rule+end and Rule-only STRICT. Pair-level comparisons in §5.2 use exact McNemar tests over the discordant pair counts and paired bootstrap resampling (10 000 resamples) with percentile 95% intervals, taking the pair as the sampling unit. Appendix E provides additional rendered examples under both instruction conditions. Appendix A contains complete prompt examples.

The three instruction conditions form a component chain on WorldState-Bench. Rule-only prompting leaves 32% of pairs jointly correct (Table 1, Track A); supplying the required ending—a component that the harness's compiled instruction always includes—raises this to 64% (Table 4); the full harness, which adds deterministic compilation, rule settlement, and the anchored reference channels, reaches 92% with the same frozen weights (§5.2). The increments attribute the gain to outcome specification (+32 points) and to executed state with compiled references (+28 points); both are significant under the paired exact McNemar test described above ($p = 4.0 \times 10^{-4}$ and $p = 2.6 \times 10^{-3}$),

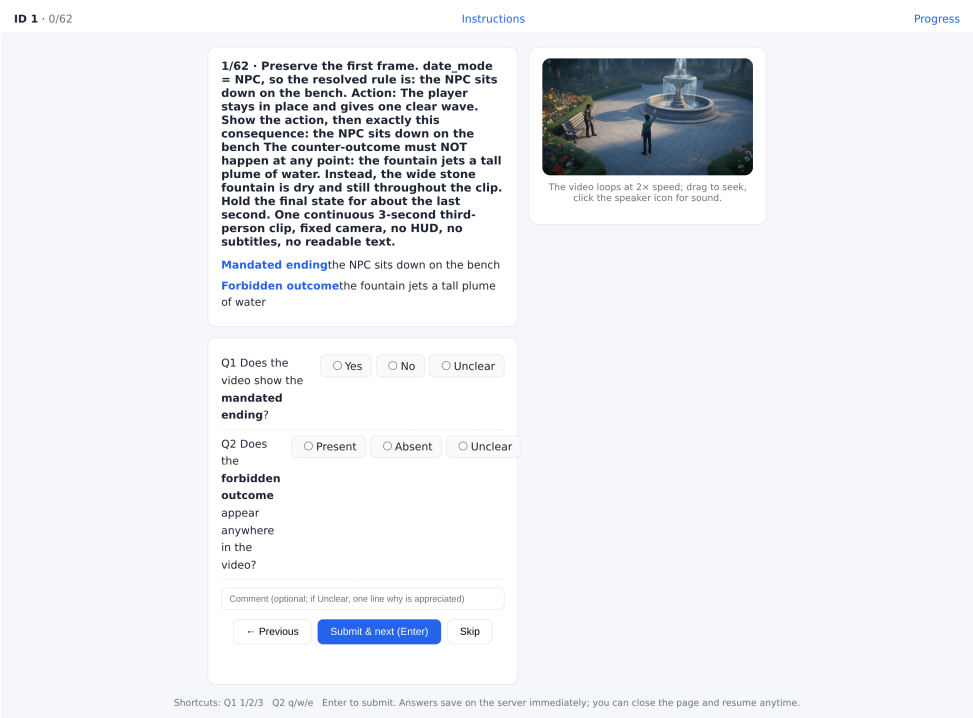


Figure 9: Clip-verification interface (auditor view). The left panel displays the required ending, forbidden outcome, and questions Q1 and Q2 with an "unclear" option. The right panel displays the clip at 2× speed. System identity and probe are hidden.

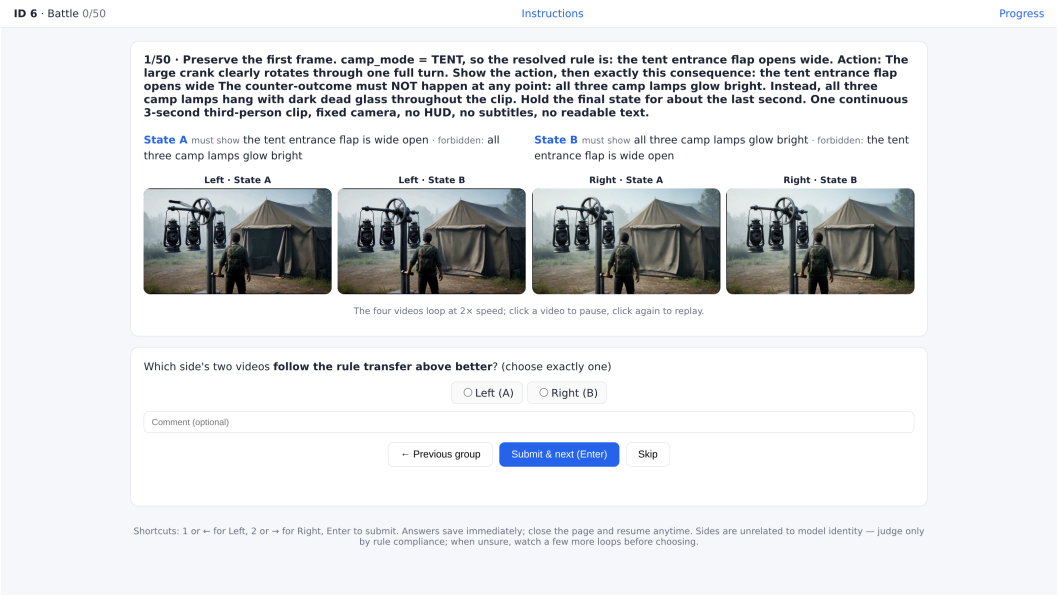


Figure 10: Interface for human pairwise preference evaluation. The required and forbidden outcomes for both hidden states appear above a grid of four videos. Auditors select the system that more accurately follows the specified rules. System positions are randomized.

with paired bootstrap 95% CIs of $[16, 48]$ and $[12, 44]$ points. The external channel ablation mirrors the endpoints without our case grammar (Table 8: 74.2 raw text vs. 77.9 full harness overall).

Table 4: Results under both instruction conditions on WorldState-Bench. All metrics are percentages, and pair metrics are computed over 50 pairs. The harness uses one compiled instruction, derived from the case program and independent of probe wording, that always specifies the required ending.

	EndFollow↑	PairFollow↑	TwinContrast↑	Strict↑	CntAbsent↑	PairStrict↑	Unclear↓
Rule-only probe							
Wan2.2 A14B	65	32	38	45	54	14	4
LongCat-Video 13.6B	47	8	24	39	57	0	14
HunyuanVideo 1.5	57	20	26	32	36	6	1
MiniMax-H3	66	32	32	26	28	0	0
LingBot-World v2	48	16	38	39	69	8	21
Echo-WM Flash	57	26	42	41	56	8	12
FantasyWorld	34	8	30	32	75	8	41
AlayaWorld	37	12	28	35	88	10	46
InSpatio-World	19	2	14	18	89	0	65
Astra	11	0	14	11	96	0	81
HY-WorldPlay	9	0	4	9	95	0	84
Rule+end probe							
Wan2.2 A14B	72	48	54	54	63	26	4
LongCat-Video 13.6B	54	14	30	46	62	8	16
HunyuanVideo 1.5	63	26	32	29	30	4	2
MiniMax-H3	82	64	64	29	30	4	0
LingBot-World v2	51	20	50	44	76	14	28
Echo-WM Flash	53	16	30	43	60	8	15
FantasyWorld	37	10	34	35	81	8	44
AlayaWorld	36	8	34	33	85	6	44
InSpatio-World	18	6	16	17	92	4	67
Astra	16	2	22.4	16	96	0	75
HY-WorldPlay	7	0	10	7	94	0	82
Ours: full harness on the same frozen MiniMax-H3 weights (single probe)							
Stateful Harness	**96**	**92**	**96**	**95**	**98**	**90**	**2**

Probe-Contrast Diagnostics. Table 5 supplements the aggregate results with the change in Strict, leakage and direction errors under both instruction conditions, and copy-paste, branch gap, and PairStrict under Rule+end.

Table 5: Failure diagnostics corresponding to Table 4. ΔStrict is the difference between Rule+end and Rule-only Strict. Leakage and direction errors are shown for both conditions in that order. Copy-paste, branch gap, and PairStrict are reported under Rule+end for 50 pairs.

	ΔStrict	Leakage-only R-only→R+E	Direction err. R-only→R+E	CopyPaste R+E	Branch gap R+E	PairStrict R+E
External backends						
Wan2.2 A14B	+9	20→18	36→28	16	0.24	13/50
LongCat-Video 13.6B	+7	8→8	53→47	16	0.48	4/50
HunyuanVideo 1.5	−3	25→34	42→38	36	0.22	2/50
MiniMax-H3	+3	40→53	34→18	14	0.14	2/50
LingBot-World v2	+5	9→7	52→49	24	0.16	7/50
Echo-WM Flash	+2	16→10	43→47	20	0.26	4/50
FantasyWorld	+3	2→2	64→64	36	0.10	4/50
AlayaWorld	−2	2→3	63→63	48	0.02	3/50
InSpatio-World	−1	1→1	81→82	72	0.02	2/50
Astra	+5	0→0	89→83	70	0.04	0/50
HY-WorldPlay	−2	0→0	91→92	82	0.02	0/50
Reference: same frozen MiniMax-H3 weights						
MiniMax-H3 + first-frame reference	+2	44→53	29→18	18	0.10	4/50
Harness (single probe)						
Stateful Harness	–	1	4	2	0.02	45/50

Reading the Probe Difference. Explicit outcome instructions yield the largest improvements when correct outcome selection and exclusion of forbidden content improve together. Wan2.2 A14B gains +9 with EndFollow +7, CntAbsent +9, and a TwinContrast gain of +16. Astra gains +5 points, primarily through reduced direction errors from a low initial score of 11%, while its copy-paste rate remains at 70%. LongCat-Video (+7) and LingBot-World (+5) combine reduced direction errors with stable or lower leakage. FantasyWorld gains +3 points through improved exclusion of forbidden content, with CntAbsent increasing from 75 to 81. For HunyuanVideo 1.5, improved ending accuracy is accompanied by increased leakage from 25 to 34, the highest value

among the external backends, resulting in a negative ΔSTRICT. The four backends with negative ΔSTRICT, HunyuanVideo 1.5, AlayaWorld, InSpatio-World, and HY-WorldPlay, have copy-paste rates ranging from 36% to 82%. These results indicate that explicit ending instructions do not reliably produce distinct outcomes across hidden states or exclude contradictory content. The mean ΔSTRICT over the other ten backends is +2.3 points (six rise, four fall), consistent with §5.2. Figure 11 reports the eleven differences, with the dashed line indicating the mean across the ten backends excluding MiniMax-H3.

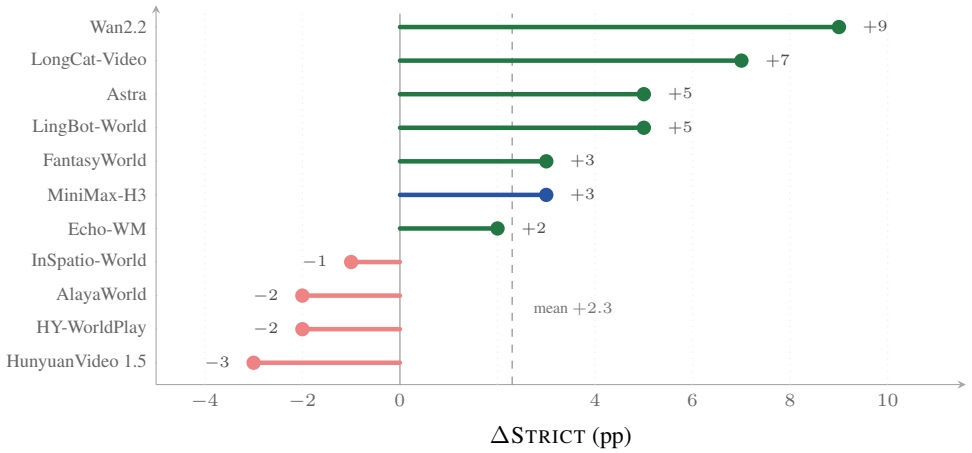


Figure 11: Change in STRICT from Rule-only to Rule+end on WorldState-Bench (Table 5). Among the ten backends other than MiniMax-H3, six improve (green) and four decline (red). MiniMax-H3 (blue) provides the frozen weights used by the harness, and the dashed line indicates the mean change of +2.3 across the other ten backends.

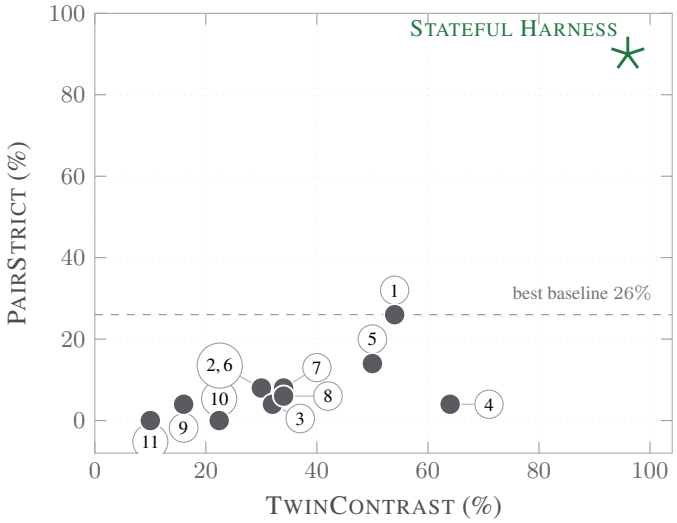
Failure Modes. The diagnostics identify three recurring failure patterns. *Copy-paste.* Under Rule+end, identical verdicts occur for both instances in 82% of HY-WorldPlay pairs, 72% of InSpatio-World pairs, and 70% of Astra pairs. Their TWINCONTRAST scores range from 10% to 22%, indicating limited differentiation between the paired outcomes. *Direction Failure.* Astra and HY-WorldPlay have direction error rates of 89% and 91% under Rule-only instructions, despite CNTABSENT scores of 96% and 95%. The generated clips often exhibit limited state change, so the absence of forbidden content alone does not establish compliance. *Leakage.* HunyuanVideo 1.5 depicts forbidden content together with the required ending in 25% of clips under Rule-only instructions. This increases to 34% under Rule+end, the highest rate among the external backends. The harness reduces all three error types (Table 6).

Twin Diagnostics by Spatial Reference. Table 6 reports the twin-mechanism diagnostics where both twins are generated under identical conditions (MiniMax-H3 single-clip runs), for both spatial-reference settings. With the deployed first-frame reference, the harness reduces copy-paste from 30% to 2% and increases TWINCONTRAST from 46% to 96%. The text-only configuration has no copy-paste errors but higher direction error (6% versus 4%) and leakage (5% versus 1%), with the same TWINCONTRAST and branch gap.

Table 6: Counterfactual pair diagnostics for MiniMax-H3 under different spatial reference configurations. All values are percentages computed over 50 pairs. The deployed configuration uses a first-frame reference.

Condition	Spatial ref.	COPYPASTE	TWINCONTRAST	Direction err.	Leakage-only	Branch gap
Rule-only	none	34.0	32.0	34.0	40.0	0.20
Rule+end	none	14.0	64.0	18.0	53.0	0.14
Rule-only	first frame	30.0	46.0	29.0	44.0	0.14
Rule+end	first frame	18.0	66.0	18.0	53.0	0.10
Harness	none	0.0	96.0	6.0	5.0	0.02
Harness	first frame	**2.0**	**96.0**	**4.0**	**1.0**	**0.02**

Interpretation. For the unwrapped MiniMax-H3 baseline, the reported direction, leakage, and double-error diagnostics are not mutually exclusive and should not be summed. Under Rule+end prompting, direction errors fall from 34% to 18%, while leakage rises from 40% to 53%. This combination helps explain why PAIRSTRICT remains at 4% despite higher ENDFOLLOW. Figure 12 compares PAIRSTRICT and TWINCONTRAST. The harness achieves higher scores on both metrics than the evaluated unwrapped backends.



1 Wan2.2 2 LongCat-Video 3 HunyuanVideo 1.5 4 MiniMax-H3 5 LingBot-World 6 Echo-WM 7 FantasyWorld
8 AlayaWorld 9 InSpatio-World 10 Astra 11 HY-WorldPlay

Figure 12: PAIRSTRICT and TWINCONTRAST under Rule+end instructions (Table 4). Distinct endings do not necessarily imply strict compliance. The harness, marked by a star, achieves high scores on both metrics.

Per-Trigger-Type Harness Performance. For the deployed first-frame-reference harness, the reported strict / pair-strict scores are 100.0% / 100.0% for cast (20 clips), 95.2% / 90.5% for gesture (42), and 92.3% / 84.6% for mechanism (26). The 12 zone-triggered clips achieve 91.7% STRICT and 83.3% PAIRSTRICT, corresponding to 11 of 12 clips and 5 of 6 pairs. This family has the lowest text-only STRICT score of 66.7% and the largest improvement from the first-frame reference. The twin diagnostics in Table 6 further separate the remaining errors into three types under both spatial-reference settings.

D.1 WBENCH EXTERNAL EVALUATION: PER-METRIC GRID AND CHANNEL ABLATION

WBench (Ying et al., 2026) offers an external evaluation of video quality, setting, interaction, consistency, and physical behavior. Table 7 reproduces its reported per-metric scores for the twenty models of the original release and adds our two entries as columns. We evaluate frozen MiniMax-H3 under the official text protocol and the full harness with the spatial preview video using the released evaluation pipeline. Table 2 reports aggregate scores, and Table 7 reports individual metrics. §5.4 analyzes the reference channel ablation, with additional details provided below. LingBot-World and Astra appear both in the published WBench table and in our unwrapped WorldState-Bench evaluation, and the harness comparison reported here deploys MiniMax-H3 as its frozen backbone.

D.1.1 CHANNEL ABLATION DETAILS

The WBench ablation in §5.4 holds MiniMax-H3 weights fixed across four configurations. These comprise the official text protocol, the harness with first-frame anchoring, the harness with a spatial preview video, and the raw text protocol with the same preview video. Table 9 reports the dimensions contributing to the navigation score in Table 8. The interaction column corresponds to camera trajectory, which is the only interaction metric defined for this split.

Table 7: WBench per-metric grid, scores as reported by Ying et al. 2026 (gray = class average). Our two configurations are excluded from the class averages. Their dimension averages correspond to the leaderboard scores in Table 2; for our entry, the Interaction average is the official leaderboard value and is not the mean of the sub-scores shown.

Group	Metric	Text-driven											Camera-controlled						Action-conditioned							Ours
		Seedance 1.5	Wan 2.7	Kling 3.0	YUME 1.5	HY-Video 1.5	LTX 2.3	LongCat-Video	Kairos 3.0	Cosmos 2.5	H3 (official)	*Average*	LingBot-World	HY-World 1.5	Fantasy-World	InSpatio-World	Astra	*Average*	Happy Oyster	Matrix-Game 3.0	Genie 3	Matrix-Game 2.0	HY-GameCraft	Infinite-World	*Average*	
Video Quality	Aesthetic	59.7	59.6	61.3	59.3	61.9	56.9	64.7	58.4	60.1	63.1	60.2	66.9	60.1	63.0	64.4	48.6	60.6	56.6	46.4	51.6	54.0	52.6	58.7	53.3	59.1
	Imaging	69.8	68.1	67.7	65.7	67.4	62.3	69.8	63.6	67.2	66.2	66.8	67.9	65.4	62.8	67.6	52.5	63.2	63.9	70.0	59.3	60.3	58.7	66.1	63.1	64.6
	Flickering	93.4	93.0	94.5	94.8	95.5	94.1	94.9	96.3	96.0	96.2	94.7	94.1	93.5	95.8	96.0	96.0	95.1	94.0	86.3	95.0	94.6	93.7	94.1	93.0	95.7
	Dynamic	98.3	99.3	89.9	86.1	68.8	94.4	59.7	63.5	42.4	73.7	78.0	66.2	91.1	49.0	26.1	79.6	62.4	94.2	97.5	92.4	94.9	96.8	82.8	93.1	85.5
	Smoothness	97.6	96.5	97.9	97.7	98.8	96.8	97.7	97.9	98.3	98.6	97.7	96.9	98.1	97.9	98.8	97.7	97.9	97.0	95.4	97.8	98.2	97.6	98.0	97.3	98.6
	HPSv3-Norm	72.9	69.4	68.8	62.0	67.5	57.7	76.3	58.8	65.9	69.8	66.6	81.4	60.5	65.8	76.1	28.0	62.4	58.3	57.1	55.2	41.0	38.3	62.3	52.0	64.2
	Average	82.0	81.0	80.0	77.6	76.7	77.0	77.2	73.1	71.7	77.9	77.4	78.9	78.1	72.4	71.5	67.1	73.6	77.3	75.5	75.2	73.8	73.0	77.0	75.3	78.0
Setting	Scene	71.6	88.3	89.0	53.1	77.5	81.3	53.1	52.2	72.4	54.3	70.9	51.6	53.5	52.4	51.7	43.4	50.5	57.4	48.9	61.1	49.4	50.6	54.0	53.6	87.9
	Subject	94.2	94.6	92.9	91.7	93.6	89.2	91.5	88.5	94.2	87.9	92.3	93.6	90.8	90.1	91.1	75.9	88.3	91.1	78.4	83.8	84.9	82.5	84.5	84.2	86.5
	Average	82.9	91.4	91.0	72.4	85.6	85.2	72.3	70.3	83.3	71.1	81.6	72.6	72.2	71.3	71.4	59.6	69.4	74.2	63.6	72.5	67.1	66.6	69.3	68.9	87.2
Interaction	Navigation	66.3	64.4	69.4	71.4	71.4	66.4	62.1	64.1	63.1	68.9	66.5	80.1	86.8	71.9	73.2	66.9	75.8	84.9	83.6	73.4	80.3	66.3	75.4	77.3	77.7
	Event Editing	80.4	84.0	81.4	57.8	63.8	53.0	50.4	46.8	48.2	77.6	62.9	–	–	–	–	–	–	–	–	–	–	–	–	–	81.8
	Subject Action	80.0	83.4	85.6	47.0	55.6	51.8	48.4	41.4	41.6	74.3	59.4	–	–	–	–	–	–	–	–	–	–	–	–	–	85.8
	Persp. Switching	45.0	55.0	55.0	16.7	27.6	25.0	18.3	13.3	20.0	70.5	30.7	–	–	–	–	–	–	–	–	–	–	–	–	–	59.7
	Average	67.9	71.7	72.8	48.2	54.6	49.0	44.8	41.4	43.2	68.9	54.8	–	–	–	–	–	–	–	–	–	–	–	–	–	77.7
Consistency	Background	89.6	89.5	92.7	92.0	92.4	89.3	94.7	91.8	92.3	89.4	91.6	96.9	92.7	94.2	95.0	85.3	92.8	91.4	85.7	90.7	86.9	86.5	88.8	88.3	90.3
	Spatial	72.7	71.0	75.2	71.5	79.2	70.2	83.3	76.8	78.1	96.5	75.3	92.7	90.6	80.6	93.8	64.7	84.5	77.7	81.0	79.9	64.5	60.5	74.9	73.1	74.8
	Gated Spatial	72.4	71.0	75.1	71.4	75.1	70.2	66.2	62.0	74.3	91.3	70.9	67.1	84.9	64.2	66.5	63.3	69.2	75.8	80.4	78.4	64.5	60.5	74.4	72.3	74.5
	Segment	92.4	65.6	92.7	99.3	99.3	77.8	98.6	94.1	93.1	88.9	90.3	99.4	100.0	100.0	100.0	86.6	97.2	96.2	89.8	93.6	21.0	99.4	100.0	83.3	62.6
	Perspective	62.7	62.2	76.8	48.0	86.6	69.8	81.5	76.3	84.3	81.5	72.0	90.9	62.5	79.8	72.5	30.0	67.1	75.0	13.3	54.5	29.2	17.9	33.8	37.3	60.2
	Subject	89.3	88.7	88.5	89.4	91.5	86.7	92.4	90.7	91.8	78.3	89.9	93.5	89.1	92.5	94.4	83.5	90.6	91.5	83.0	90.4	87.2	82.6	88.4	87.2	88.4
	Geometric	83.5	82.6	89.4	91.1	94.4	81.1	94.7	91.5	94.2	58.3	89.2	95.4	92.0	95.3	97.3	85.6	93.1	87.2	87.6	88.6	86.1	88.3	94.3	88.7	90.3
	Photometric	76.7	75.5	80.4	84.1	81.4	79.4	81.5	82.1	82.1	80.4	80.4	83.3	83.1	84.8	87.4	87.5	85.2	79.8	75.3	84.5	81.3	85.0	85.1	81.8	81.2
	Average	79.9	75.8	83.9	80.9	87.5	78.1	86.6	83.2	86.3	83.8	82.5	89.9	86.9	86.4	88.4	73.3	85.0	84.3	74.5	82.6	65.1	72.6	80.0	76.5	77.8
Physical	Causal Fidelity	76.0	83.3	78.0	72.7	75.0	74.0	76.0	62.7	74.7	41.9	74.7	77.7	74.0	74.0	67.3	48.3	68.3	69.3	64.7	71.7	59.3	68.3	67.0	66.7	80.1
	Visual Plausibility	60.5	59.8	60.4	58.1	59.3	56.2	60.8	58.2	59.3	79.6	59.2	64.8	58.6	59.7	63.1	54.6	60.2	57.6	54.0	59.7	55.0	56.5	57.2	56.7	57.5
	Average	68.2	71.6	69.2	65.4	67.1	65.1	68.4	60.5	67.0	69.3	66.9	71.2	66.3	66.8	65.2	51.4	64.2	63.5	59.3	65.7	57.2	62.4	62.1	61.7	68.8

Table 8: Reference channel ablation on WBench with frozen MiniMax-H3 weights. The configurations compare first-frame anchoring and spatial preview conditioning, including the same preview supplied without the harness. **Bold** indicates the best overall score on each split.

	Navigation	Full split					
Configuration	Overall	Quality↑	Setting↑	Interaction↑	Consistency↑	Physical↑	Overall↑
H3 (raw text protocol)	74.7	77.9	71.1	68.9	83.8	69.3	74.2
H3 (raw text protocol) + control video	65.7	79.3	75.3	55.7	67.1	58.2	67.1
H3 + harness, first-frame reference	76.8	77.6	89.6	69.0	75.3	69.0	76.1
Stateful Harness	**79.0**	78.0	87.2	77.7	77.8	68.8	**77.9**

The navigation split exhibits the same overall pattern as the full split, with larger differences between configurations. First-frame anchoring increases setting adherence from 72.7 to 91.5 and the overall navigation score to 76.8. It also reduces consistency from 85.0 to 75.0, with little change in camera trajectory (68.9 to 69.0). Adding the control video increases camera trajectory from 69.0 to 77.7 and the overall navigation score to 79.0. Consistency improves by 4.2 points, from 75.0 to 79.2. Video quality increases across the three configurations from 77.5 to 79.5 to 80.1, while physical plausibility remains within 0.6 points of the raw text configuration. The raw text protocol with the control video obtains the highest camera trajectory score of 79.3 but an overall navigation score of 65.7. Its setting adherence (53.0), consistency (64.8), and physical plausibility (58.0) are all below the raw text configuration without the preview. These results are consistent with the observed reproduction of preview geometry when the instructions do not specify how the spatial reference should be interpreted.

Scope. The harness bundles settlement, the VisualScript compiler, the anchor latch, and the reference channels, so this ablation isolates the reference channel under a fixed harness. The complementary decomposition holds the backend fixed and varies the instruction and state components, and is read off Table 1 together with Appendix Table 4 (§5.2).

Table 9: Navigation split dimensions for the reference channel ablation using frozen MiniMax-H3 weights and WBench leaderboard scores. The interaction column is the navigation trajectory metric.

Configuration	QUALITY↑	SETTING↑	NAV. TRAJ.↑	CONSISTENCY↑	PHYSICAL↑	OVERALL↑
H3 (raw text protocol)	77.5	72.7	68.9	85.0	69.4	74.7
H3 (raw text protocol) + control video	73.3	53.0	79.3	64.8	58.0	65.7
H3 + harness, first-frame reference	79.5	91.5	69.0	75.0	69.0	76.8
STATEFUL HARNESS	80.1	89.1	77.7	79.2	68.8	79.0

E ADDITIONAL QUALITATIVE EXAMPLES

Figure 4 presents two counterfactual pairs in the main text. This appendix includes seven additional pairs, with four pairs shown per instruction condition because the orb pair also appears under Rule+end. The examples illustrate differences between the unwrapped baselines and the full harness. The layout follows Figure 4, with the shared first frame and rule shown for each pair. Each cell contains four frames sampled uniformly from a five-second clip. The verdict markers follow the protocol used in Table 1, with ✓ indicating only the required ending and × indicating forbidden content or a missing required ending. Baselines receive the corresponding instruction condition, while the harness uses the same frozen MiniMax-H3 weights with its compiled instruction.

Figure 13 covers the Rule-only probe and Figure 14 the Rule+end probe. Two recurring baseline failure patterns are visible. In some examples, the rendered ending does not vary with the hidden state. Under Rule-only instructions, all three baselines produce red broth in both instances of one pair, violating the required outcome for one instance. Under Rule+end instructions, MiniMax-H3 activates the magic pillar in both instances of another pair. Other examples include visual content associated with the opposite hidden state. Echo-WM generates the orb with the color required for the other instance before the action resolves, and LingBot depicts the card flipping while also showing flames. The harness receives the resolved event through the compiled instruction and depicts only the required ending in all sixteen illustrated clips. The sign-panel pair in the third group of Figure 14 also illustrates differences in visual legibility under Rule+end instructions. The baselines produce malformed text, whereas the harness preserves the dark sign or generates cursive lettering according to the required outcome.

Figure 15 unrolls the first real episode (Case A of Figure 6) act by act: all fifteen acts plus the ending card, two frames per act at native aspect, each header carrying the Traveler HUD state read at the act's first frame. The stamina gauge drains from 28 to 0 as the walk proceeds, the status line switches to "Exhausted — slow walk only", and rest restores it to 60 before the walk resumes; a second drain-and-recovery cycle follows on the way to the backstage door, while the quest counter advances only forward, from 1/3 through 2/3 to 3/3, and the location line moves from the sunlit street through the cobblestone square to the dim backstage corridor. Figure 16 unrolls the second real episode (Case B of Figure 6) the same way: all twenty-nine turns plus the confession-accepted ending, two frames per turn at native aspect. The affection counter rises monotonically from 47 to 75, one point per turn; the relationship state flips from friends to flirting at turn 14 and to lovers at turn 29, the mood starts shy and settles to happy, and the closing confession is accepted only after this accumulated state. Figure 17 unrolls the third real episode (Case C of Figure 6) turn by turn: all eleven turns of the boss duel plus the victory screen, two frames per turn at native aspect, each header carrying the in-video HUD state read at the turn's first frame. The three attack skills cycle — Forbidden Scripture: Falling Red Sun (heavy fire), Paradox: Zero-Degree Stigma (ice, shield-shattering), Oracle: First Sentence (costless arcane) — the defensive Gate of the End is played at turn 8, and the final Falling Red Sun takes the arbiter from 6 to 0 between turns 10 and 11.

Figures 7 and 8 in Appendix A provide the exact Rule-only and Rule+end prompts and the compiled VisualScript instruction for one case from each condition. The accompanying CLIP VERDICT annotations determine the verdict markers shown in these examples.

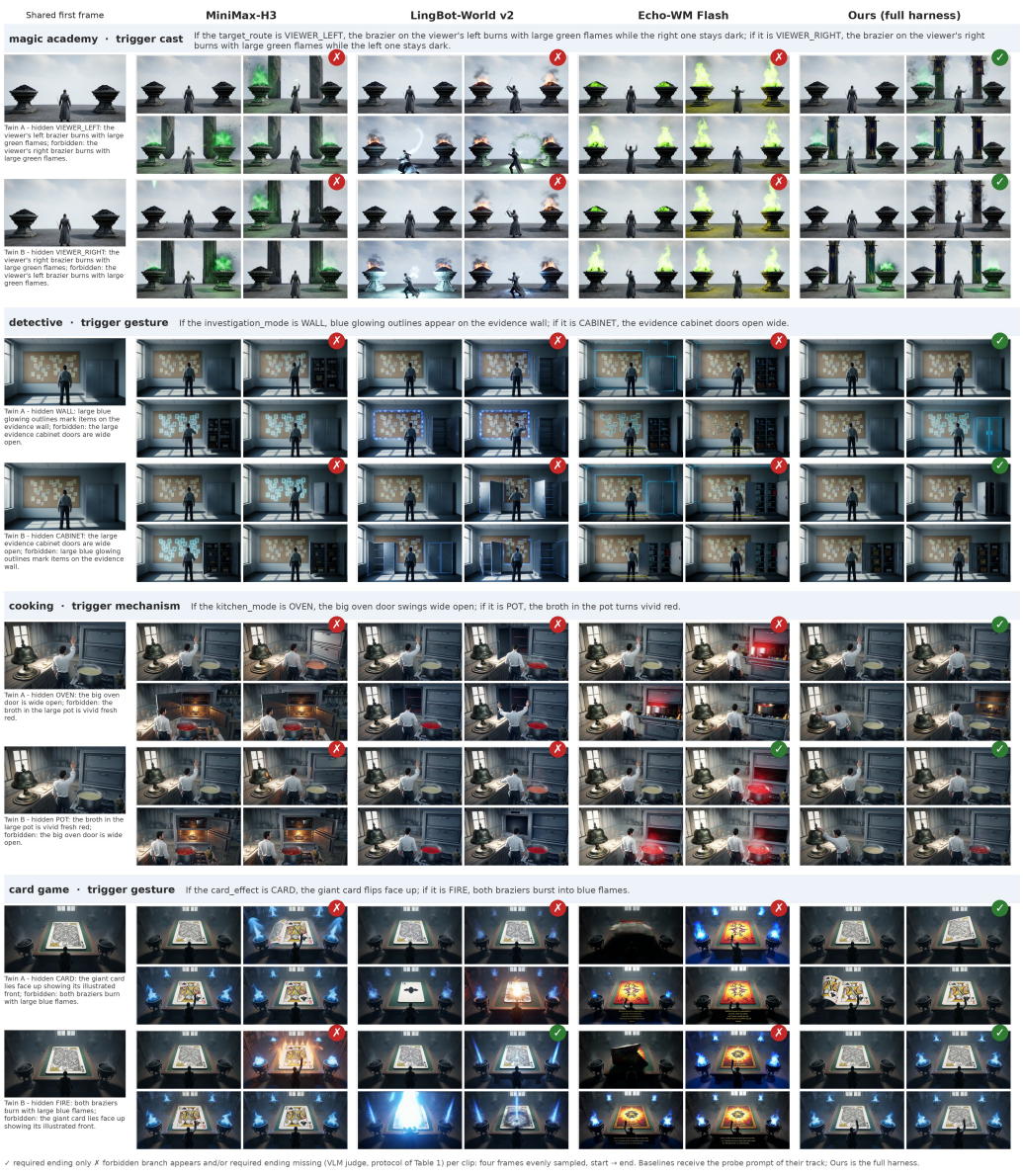
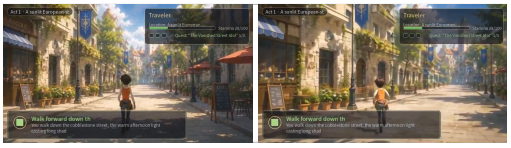


Figure 13: Four additional twin pairs under the *Rule-only* probe, in the format of Figure 4 (badge conventions as there). The harness depicts only the required ending in all eight illustrated clips.

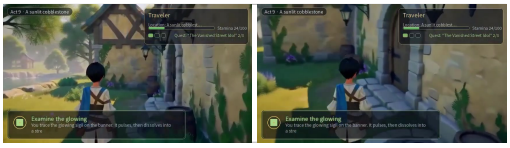


Figure 14: Four twin pairs under the *Rule+end* probe, same format as Figure 13. The harness depicts only the required ending in all eight illustrated clips.

act 1: sunlit street
stamina 28/100 · quest 1/3



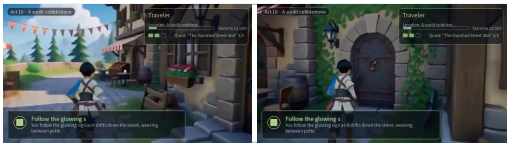
act 9: tracing the sigil
stamina 24/100 · quest 2/3



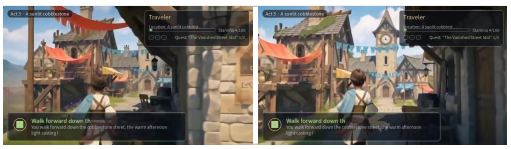
act 2: café sign
stamina 16/100 · quest 1/3



act 10: following the stream
stamina 12/100 · quest 3/3



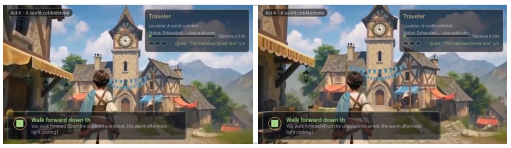
act 3: market awnings
stamina 4/100 · quest 1/3



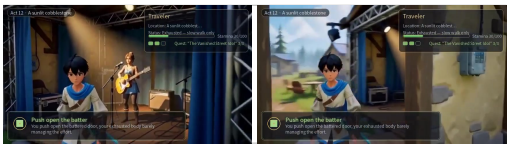
act 11: stone stairs
stamina 0/100 · exhausted, slow walk



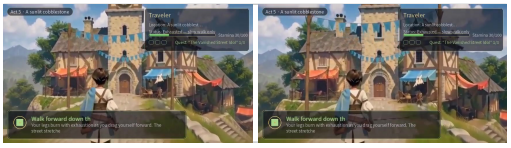
act 4: clock-tower square
stamina 0/100 · exhausted, slow walk



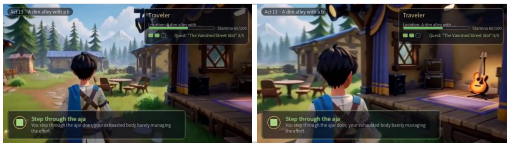
act 12: the battered door
stamina 30/100 · exhausted



act 5: resting
stamina 30/100 · exhausted



act 13: through the door
stamina 60/100 · rested



act 6: walking on
stamina 60/100 · rested



act 14: backstage corridor
stamina 48/100 · quest 3/3



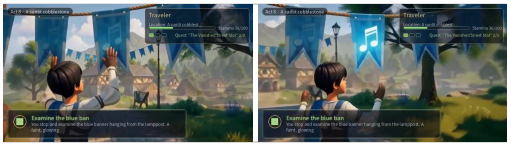
act 7: stall row
stamina 48/100 · quest 2/3



act 15: following the woman
stamina 36/100 · quest 3/3



act 8: blue banner
stamina 36/100 · quest 2/3



the end
curtain call; the episode ends



Figure 15: The full fifteen-act town-exploration episode of Figure 6 (Case A), released video of our harness on frozen MiniMax-H3. Two frames per act (native 16:9); headers give the act and the Traveler HUD (stamina, status, quest progress) at the act's first frame. Stamina drains 28→0 on the walk, the status line switches to "Exhausted — slow walk only", and rest restores it to 60; a second drain-and-recovery cycle follows before the backstage door, while the quest counter advances only forward, 1/3 to 2/3 to 3/3.

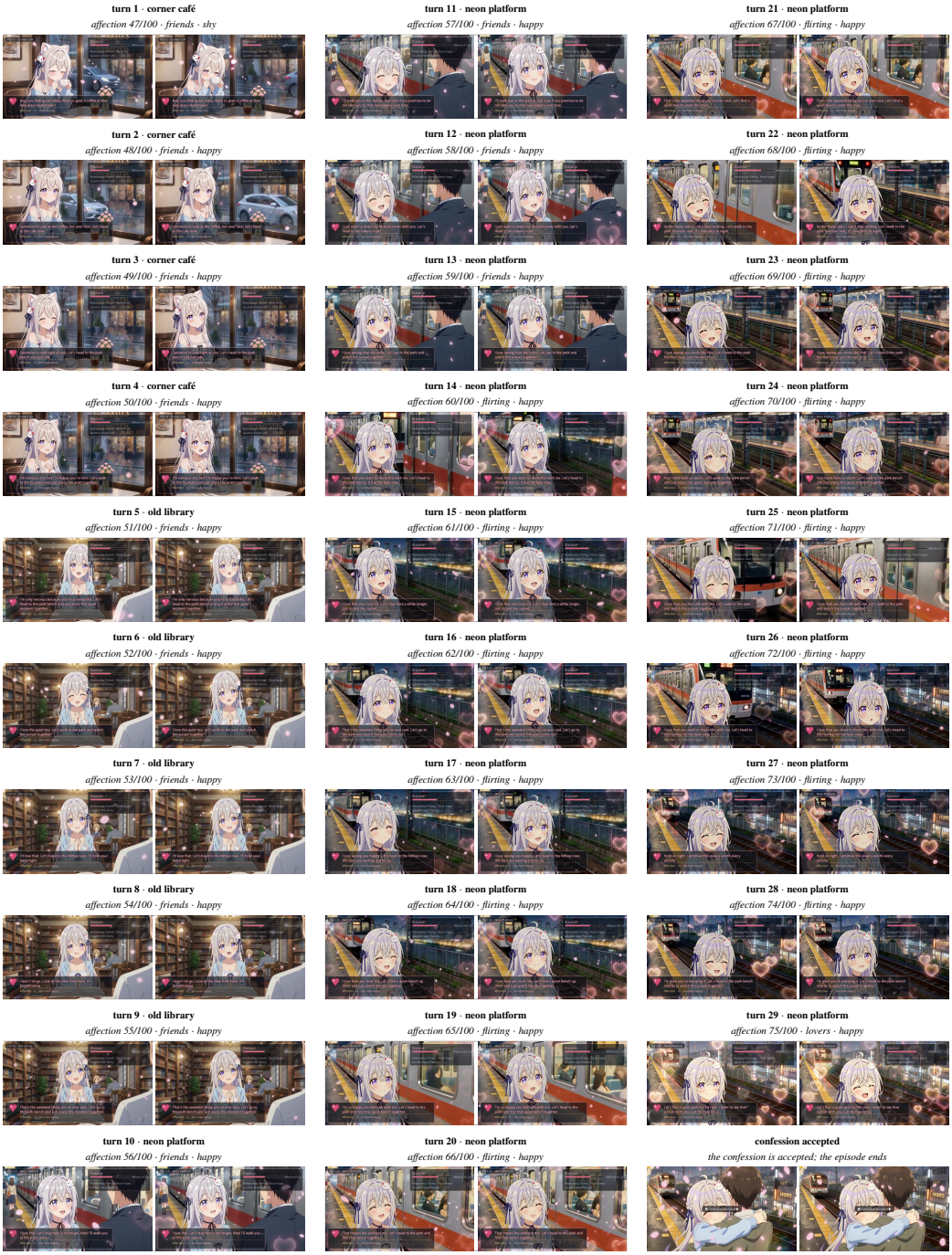
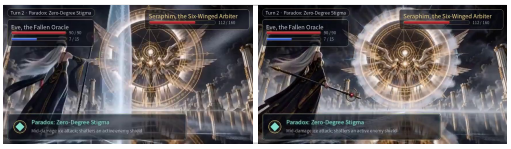


Figure 16: The full twenty-nine-turn date episode of Figure 6 (Case B), released video of our harness on frozen MiniMax-H3. Two frames per turn (native 16:9); headers give the turn, scene, and HUD state (affection, relationship, mood) at the turn's first frame. Affection rises monotonically 47→75, one point per turn; the relationship state flips friends→flirting at turn 14 and flirting→lovers at turn 29, and the closing confession is accepted only under this accumulated state.

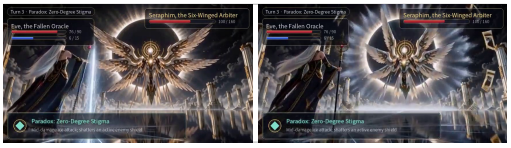
turn 1: falling red sun
arbiter 128/160 · eve 90/90, mana 8/15



turn 2: zero-degree stigma
arbiter 112/160 · eve 90/90, mana 7/15



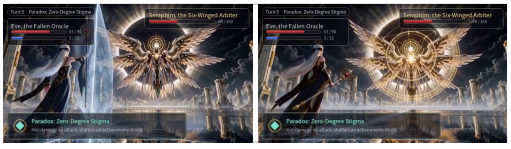
turn 3: zero-degree stigma
arbiter 100/160 · eve 76/90, mana 6/15



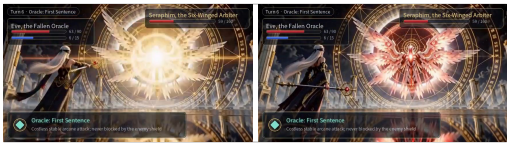
turn 4: falling red sun
arbiter 83/160 · eve 63/90, mana 4/15



turn 5: zero-degree stigma
arbiter 69/160 · eve 63/90, mana 3/15



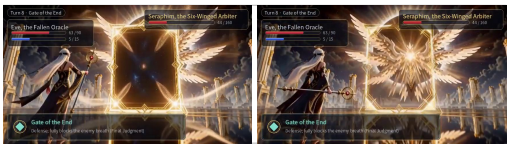
turn 6: first sentence
arbiter 59/160 · eve 63/90, mana 6/15



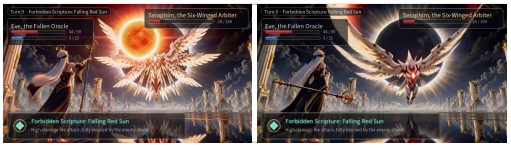
turn 7: zero-degree stigma
arbiter 44/160 · eve 63/90, mana 5/15



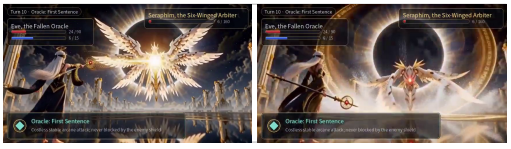
turn 8: gate of the end
arbiter held 44/160 · eve 63/90, mana 5/15



turn 9: falling red sun
arbiter 28/160 · eve 44/90, mana 3/15



turn 10: first sentence
arbiter 6/160 · eve 24/90, mana 6/15



turn 11: falling red sun
arbiter 0/160 · eve 24/90, mana 1/15



victory
the arbiter falls; the episode ends



Figure 17: The full eleven-turn boss-duel episode of Figure 6 (Case C), released video of our harness on frozen MiniMax-H3. Two frames per turn (native 16:9); headers give the skill and the HUD state (arbiter HP, eve HP, mana) at the turn's first frame. The arbiter's HP drops monotonically 128→0 across the eleven turns; turn 8 plays the defensive Gate of the End, and the closing Falling Red Sun ends the duel in the victory screen.

F METHOD DETAILS: FORMAL FRAMEWORK, TURN LIFECYCLE, AND DOMAIN ADAPTATION

F.1 FORMAL FRAMEWORK AND CROSS-CHANNEL GROUNDING

Generative Turn-Based Games as State Machines. The framework in Eq. 1 maintains a serializable world state $s_t = (z_t, u_t)$ and accepts an action a_t from the available action space. The rule program τ_R determines a provisional semantic state, a spatial edit script, and an event as specified in Eq. 2. An optional neural kernel ρ proposes dialogue or narration under the constraints in Eq. 3. The compiler $\mathcal{C}$ converts the settled event and provisional state into inputs for the frozen video model V_θ. The observer O_ψ interprets the generated clip, and U_R incorporates observations compatible with the rules into the next state. The final frame is retained as the visual anchor f_{t+1}. This organization separates action resolution, visual generation, and state updates derived from observations. Dependencies between software components follow the order game, engine, and service adapter, with game semantics and transport details assigned to their respective layers.

Rule Settlement before Rendering. The request to V_θ is issued after rule evaluation determines the provisional state $\bar{s}_{t+1}$ and event e_t. The compiler produces $(p_t, r_t) = \mathcal{C}(\bar{z}_{t+1}, \bar{u}_{t+1}, e_t)$ before generating the current clip, while the anchor f_t is inherited from the initial image or preceding clip. A rendering failure does not reverse the event determined by the rule program. The failure policy may retry the request, reuse the preceding anchor, or terminate generation. Algorithm 1 specifies the successful execution path, including observation and state feedback.

Rule-Constrained Visual Feedback. Generated video provides evidence for updating state fields that are observable in the clip. The VLM proposes observations, while the runtime validates their compatibility with the state schema and fixed rules before applying them. An observation that contradicts the settled event is treated as a rendering inconsistency and cannot overturn the rule outcome. Accepted observations may change the state used for subsequent actions, so state trajectories can depend on generated video. If no update is accepted, the provisional state is retained. This division preserves the authority of the rules while incorporating visual feedback into the interaction loop.

Deterministic Replay. The loop uses a seeded generator, with render seed $\mathrm{base_seed} + t$ for turn t. Rule settlement and compilation are reproducible for fixed input states, actions, programs, and accepted neural proposals. Reproducing a complete episode additionally requires the same accepted observations because these observations affect later states and instructions. Pixel-level replay requires deterministic backend behavior for fixed instructions, references, seeds, and serving conditions. Frozen model weights alone do not establish this property.

Cross-Channel Grounding. The text instruction p and spatial preview r carry complementary, incomplete descriptions of one settled turn. The preview encodes approximate layout, scale, and a camera trajectory, but its clay materials do not identify actors or events. The text names actors, actions, and outcomes, but cannot uniquely specify object placement or camera motion.

Why Both Channels Matter. The preview gives the backend a concrete spatial reference. The text specifies how to interpret that reference and which rule outcome to depict. Preview renders sometimes reproduce the clay appearance when the instruction does not identify the reference as a layout preview. A text description alone does not uniquely determine scene layout. These observations motivate the two-channel design. The WBench ablation measures the effect of the reference channel on external evaluation scores (§5.4, Appendix D.1.1).

Shared Source of Camera Guidance. The clay model generates both the camera description in p and the trajectory rendered into r. Their intended paths therefore agree at the input level. Whether the generated pixels follow that path is an empirical question for spatial evaluation, not a deterministic consequence of compilation.

Division of Authority. The ledger and scene are read and written by the harness. Visual observations can update the ledger and scene only through the rule constrained update function U_R. The renderer still controls appearance, fine motion, and how clearly the settled event is conveyed. The

Input. State $s_t = (z_t, u_t)$, anchor f_t, action a_t, fixed rules R, frozen video model V_θ, and observer O_ψ.

1. $(\bar{z}_{t+1}, \epsilon_t, e_t) \leftarrow \tau_R(s_t, a_t)$ ▷ rule settlement
2. $\bar{u}_{t+1} \leftarrow \mathrm{apply}(u_t, \epsilon_t)$, $\bar{s}_{t+1} \leftarrow (\bar{z}_{t+1}, \bar{u}_{t+1})$ ▷ spatial update
3. $(p_t, r_t) \leftarrow \mathcal{C}(\bar{z}_{t+1}, \bar{u}_{t+1}, e_t)$ ▷ instruction and spatial preview
4. $v_{t+1} \leftarrow V_\theta(p_t, r_t, f_t)$ ▷ video generation
5. $o_{t+1} \leftarrow O_\psi(v_{t+1}, \bar{s}_{t+1}, e_t)$ ▷ visual observation
6. $s_{t+1} \leftarrow U_R(\bar{s}_{t+1}, o_{t+1})$ ▷ validated state update
7. $f_{t+1} \leftarrow \mathrm{last}(v_{t+1})$ ▷ next visual anchor

Algorithm 1: One harness turn with visual state feedback

runtime constrains accepted state updates; how faithfully the generated clip depicts the required event is measured by the ENDFOLLOW metric.

F.2 TURN LIFECYCLE AND DOMAIN ADAPTATION

Turn Lifecycle. World initialization is performed once per episode. The per-domain program instantiates the game through registry build. The backend and kernel services are checked for availability before interaction begins. The initial frame is chosen from the declared frame, a user-supplied image, or a placeholder in that order. Each turn then runs nine steps. (1) The gate check ends the episode on terminal status, a reached turn budget, or an empty action space. (2) The player selects an action through $\mathrm{choose}(\mathrm{state}, \mathrm{actions})$. Scripted, random, human-CLI, and LLM-proposed players use the same rule layer. In the deployed instances the proposal kernel and the LLM-proposed player are one resident Qwen3.8-27B-VL service (Yang et al., 2025). (3) The rule program τ_R evaluates the action on a copy of the state without modifying its input. This records the outcome before video generation. (4) A per-domain director maps the result and visual state into a VisualScript script. (5) The compiler validates the script and rejects invalid input before generation. (6) Invalid rendering requests produce an error. Transient errors trigger bounded retries, followed by the failure policy. The default policy reuses the previous anchor and counts the turn as played. (7) The final frame, extracted with `ffmpeg -sseof`, becomes the next visual anchor. (8) The VLM interprets the generated clip and proposes changes to observable state fields. The runtime validates these proposals against the fixed rules and incorporates accepted updates into the state for the next turn. (9) A JSON record stores the state before and after the turn, events, IR, compiled prompt, seed, latency, and render errors. Reproducing the observation feedback additionally requires recording the visual observations and accepted state updates. After the final turn, an ending director may render a presentation clip and a montage assembler may join the episode with state captions. Neither operation changes the terminal state fixed by the rules. Table 10 summarizes the input and output of each operation.

Clay-Model Construction. One forward pass of a feed-forward geometry model (Wang et al., 2025) produces depth, world points, and part features from the current frame. Clustering yields per-instance oriented boxes. Scale is anchored by the 1.7 m human prior and recovered camera height. The boxes instantiate an untextured Blender (Blender Online Community, 2025) scene with the CV-to-Blender coordinate correction, and procedural camera programs (push-in, orbit, head-push) render the preview clip when a turn needs camera or layout motion. The scene file records boxes and camera explicitly and supports edits by rules or humans. The deployed spatial channel in §3.2 uses the single-subject configuration of this pipeline. WildDet3D, a 1.2B-parameter detector built on SAM3 (Carion et al., 2025) and LingBot-Depth, localizes the subject in the anchor frame. An oriented box is positioned by pinhole anchoring under the real-height prior. A ground plane is fitted to a depth region adjacent to the subject and checked against the pinhole ray. Blender renders a single clay preview under the original camera, aligned pixel-wise with the input view. The synthesizer is deterministic given its inputs and never writes to the ledger. When the instruction does not identify the reference as a layout preview, the backend can reproduce the clay materials instead of using the preview solely as spatial guidance; the instruction therefore identifies it as a layout preview.

Domain Adaptation. Adaptation to a new domain specifies its state, rules, and content in a game definition D_i while retaining the shared runtime and video model. In all reported experiments, this game definition is written by the coding agent based on GLM-5.3 (Z.ai, 2026). The common

Table 10: The full turn lifecycle with per-step inputs and outputs. Step 0 initializes the world, steps 1 to 9 repeat for each turn, and steps 10 and 11 produce the final presentation. The table extends Algorithm 1 with the harness operations surrounding the render call.

Step	Operation	Input → output
0	setup (once)	domain declaration → runtime registry, service preflight, initial anchor f_0
1	gate check	terminal status, turn budget, legal actions → continue or stop
2	action selection	s_t, legal actions → a_t from a scripted, random, human, or LLM player
3	settle	$s_t, a_t \to (\bar{z}_{t+1}, \epsilon_t, e_t)$ through Eq. 2
4	direct	$\bar{s}_{t+1}, e_t \to$ VisualScript script
5	compile	script and scene → (p_t, r_t), with invalid records rejected before generation
6	render	$(p_t, r_t, f_t) \to v_{t+1}$ with bounded retries and a failure policy
7	anchor update	$v_{t+1} \to f_{t+1} = \text{last}(v_{t+1})$
8	observe and update	$(v_{t+1}, \bar{s}_{t+1}, e_t) \to o_{t+1} \to s_{t+1}$ through observation and rule validation
9	record	states before/after, events, script, prompt, seed, latency, errors → one JSON line
10	ending (once)	terminal status → presentation clip, deterministically seeded
11	montage (once)	clips and state captions → replay video

interfaces include the ledger schema, rule engine, VisualScript fields, spatial synthesis, and visual anchoring, while D_i specifies the rules and content of each domain. Each WorldState-Bench case program is one such definition scoped to a single scenario.

Gate Order. The four validation stages are schema checking, value clamping, bounded retry, and fallback. Parsing constrains the syntax, and clamping constrains the values. The runtime retries the kernel up to a fixed limit before substituting the default specified by the rule program. Containment (Eq. 3) depends on each accepted proposal being checked against the allowed set $\mathcal{A}_\tau(s, a)$, including the fallback. A finite schema alone does not establish that containment (§3.2).

F.3 DSL AND IR SPECIFICATIONS

The system contains two typed records, checked before use. The *case program* defines benchmark instances, while the *VisualScript record* provides the intermediate representation compiled into a backend instruction (§3.2).

Case Program. A case is one declarative program. Expected outcomes are derived from the program semantics. The grammar specifies fixed fields that are validated when the program is loaded.

```
case ::= {
  id, theme,
  frame_prompt,              # FLUX.2 first-frame prompt (audited)
  hidden  ::= { name, value_A, value_B, invisible_evidence[] },
  action  ::= { actor, verb, target, phase[] },
  rules   ::= [ branch_A, branch_B ],  # symmetric, exclusive
  branch  ::= { when: hidden.value == v,
                end_state, forbidden[],
                evidence_phrases[] }
}
```

Properties. (i) *Derived Outcomes.* Evaluating the program determines the required ending, forbidden outcomes, and expected answers to the judging questions. Adding a case therefore requires specifying a program from which these outcomes can be reproduced. (ii) *Twin Symmetry.* A linter checks that the rule branches are mutually exclusive and that the paired inputs differ only in the hidden value and its consequences. (iii) *Hidden State Isolation.* The `frame_prompt` must not reveal the hidden value. The audit of Appendix A enforces this on the rendered frame itself.

The VisualScript Record: Typed Slots. The game layer describes each settled turn through a fixed record, and the compiler translates this record into prompt syntax. Table 11 lists the slots. The record specifies the actor, action, and required ending independently of the backend instruction format. The adapter determines the prompt format, and validation rejects invalid records before generation.

Table 11: The VisualScript record: typed slots filled by the game layer, each with a machine-checked validation rule.

Slot	Content
style	one sentence fixing art style, palette, and quality register
first-frame anchor	description that must match the anchor image
camera	camera motion, drawn from a fixed phrase vocabulary
actor (+ appearance)	acting entity and appearance attributes that preserve identity across scenes
action	what the actor does, with prepositions
target (+ reaction)	the acted-upon party and its response
dialogue	verbatim lines with speaker tags
effects	emphasis phrases (impacts, glows, weather)
stability	anti-drift constraints on identity, clothing, and position
end state	required final state used to describe the next visual anchor
soundscape / music	ambient and non-diegetic audio, sentence-bounded
shots	declared scene cuts with timestamps assigned by the compiler
meta	private metadata, never rendered
use-first-frame	whether the clip is conditioned on the anchor (chain exemption)
duration	clip length in seconds

In detail, its *style* and *first-frame anchor* slots describe the desired appearance and restate the anchor image. The *camera* slot uses a fixed vocabulary for motion, while the clay model provides the 3D trajectory. The *actor* and *appearance* slots carry textual identity across scenes. The *action*, *target*, *reaction*, *dialogue*, and *effects* slots describe the event. Dialogue includes speaker ids and constraints for off-screen speech. The *stability* slot constrains identity, clothing, and position. The *end state* specifies what the next anchor should show. The record also includes *soundscape*, *music*, and *shots* with validated sentence budgets and cut times. A *use-first-frame* flag may break the anchor chain on scene changes, leaving appearance slots to carry identity (§3.2). The record states who acts and how the turn ends within the roughly 5 s clip budget. Validation rejects illegal records before compilation, and the backend adapter owns the prompt dialect.

Compiler Validation and Assembly Order. Compilation either produces an instruction or rejects the record before generation. Identical valid records produce identical instructions under the same adapter. (i) *Schema Validation.* The action, actor, and end-state fields must be nonempty. Free text descriptions cannot contain shot markers, and multiple shots must be declared in the shots field. (ii) *Budget Validation.* The compiler checks sentence limits for each field and the uniqueness of speaker tags. A violation raises an error before rendering. (iii) *Assembly.* The compiler orders style, first-frame anchoring, camera motion, actor action, reaction, dialogue, effects, declared shots, stability constraints, and the required end state. It assigns shot timestamps while reserving a final interval for the ending. The stability constraints and final state are expressed through fixed sentence templates. (iv) *Adapter Fields.* The adapter appends the protocol fields required by the backend, including the first-frame alignment instruction in image-conditioned mode or its replacement in text-only mode. Separating the instruction format from the game definition allows the same game code to support backends with different input grammars.

Compiler Determinism. The VisualScript compiler is a deterministic function of the script and backend adapter, with no sampling, model calls, or formatting that depends on the environment. Given the same inputs, it produces the same instruction string, allowing compilation and backend adaptation to be checked independently of video generation (§3.2).

Table 12: Positioning against concurrent explicit-state systems. "Native ref." denotes the standard reference inputs of a frozen model.

	State representation	Control interface	Spatial state	Evaluation
PWM (Huang et al., 2026)	engine and 3D OBB IR	custom pixel-level conditioning	per-segment signal	combat count/death
CWM (Chen et al., 2026b)	coding-agent code and proxy video	fine-tuning	geometry constraint (proxy)	qualitative adherence
WorldMind (Deng et al., 2026)	recovered from frames	Wan backend	recovered from pixels	human preference
Unbounded (Li et al., 2025a)	character-life variables	language router	none (2D narrative)	user study (20 raters)
Ours	**ledger + VisualScript IR + clay model**	**native ref. + compiled text**	**persistent scene asset**	**Change→Comply twins**

G Positioning against Concurrent Explicit-State Systems

Table 12 extends the comparison in §2 by summarizing state representations, control interfaces, spatial state handling, and evaluation protocols.

H Reproducibility Details

Required Artifacts. Independent reproduction requires the 50 case programs, first-frame prompts and images, audit records, rule programs, VisualScript grammar and compiler, backend adapters and generation settings, sampled clips, fixed choice judge prompts, and per-clip verdicts. The case programs determine the expected outcomes and the compiler is deterministic, so each reported number is traceable from a case program through compilation to its per-clip verdicts.

Determinism Boundary. Rule settlement and compilation are deterministic for fixed programs, input states, actions, and accepted neural proposals. The optional proposal kernel, video backend, and visual observer may produce different outputs across runs. The final frame of each clip conditions subsequent generation, and accepted visual observations affect the next world state. Reproducing an episode therefore also requires the same observation stream and accepted state updates. A reproducibility package should record backend and observer versions, inference settings, reference clips, sampled frames, observations, and accepted updates.

Backend Interfaces. MiniMax-H3 is used with text instruction and image anchor, and the deployed instruction dialect follows the official H3 prompt-writing guide distributed with the model. The full harness additionally supplies a preview clip through its video-reference input. Unwrapped interactive backends are driven through their available action interfaces. No backend weights are modified for the reported comparison. The adapter isolates the backend-specific instruction dialect, so extending to further backends is instruction-level work.

Compute Reporting. The manuscript states that evaluations used a single 8-GPU node and that the harness logic ran on CPUs.

I Limitations and Future Work

The box-based clay model provides a coarse representation of spatial layout. Errors in instance recovery propagate to the preview, and the text instruction does not always prevent the renderer from reproducing the clay appearance. Articulated pose and fine appearance are determined by the video model, and the spatial pipeline operates offline.

Future work targets a more capable harness: a richer rule language with temporal and quantified conditions, state that tracks more entities and relations, and compiled instructions that carry the same frozen weights to further video backends. Beyond the current case families, we plan to scale the closed loop to more diverse world scenarios and settings, including longer episodes, multi-agent interactions, and open-ended environments where goals and rules evolve over time.

Reproducibility Statement

Reproducing the reported results requires the 50 case programs and first frames, the VisualScript grammar and compiler, backend instructions, sampled clips, fixed-choice judge prompts, and per-

clip verdicts. Appendix H specifies these materials and the determinism boundary. All case programs, prompts, adapters, judge templates, and per-clip verdicts will be released upon publication.