H3-WORLD: TURNING LANGUAGE UNDERSTANDING INTO WORLD CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

We present **H3-WORLD**, an efficient framework that turns the 33B MiniMax-H3 video generator into an interactive world model. Our key finding is that, as large video generators become more capable, language is emerging as a natural interface for control. MiniMax-H3, for example, already supports zero-shot control of character behavior and camera motion through natural-language instructions. Building on this, H3-WORLD turns this coarse language interface into precise, temporally grounded world control, without introducing dedicated action modules. Specifically, we represent each action as a structured combination of character and camera instructions, and align them with the corresponding temporal video latents. To make the control temporally precise, we further introduce temporal attention routing, which restricts each instruction to its intended time interval and reduces control leakage across actions. Importantly, H3-WORLD directly reuses the semantic representations learned during large-scale video pretraining and requires only lightweight adaptation. With only 8,000 gameplay samples, 10,000 LoRA optimization steps, and 0.199% trainable parameters, H3-WORLD achieves effective character and camera control while preserving strong generation quality. It also generalizes to unseen scenarios. These results show that the control capabilities emerging in large video generators can be efficiently transformed into interactive world control. Anonymous project page is available at https://anonymous.4open.science/w/H3-WORLD-website-6C01/.

1 INTRODUCTION

Language is becoming more than a way to describe generated worlds. As video generators grow more capable, language is also beginning to shape how those worlds evolve—how agents move, how cameras behave, and how scenes change over time (MiniMax, 2026; Bruce et al., 2024; Wan et al., 2025). In this sense, language is emerging as a high-level abstraction layer for visual dynamics (Figure 1). This shift is especially relevant to world models, which learn how environments evolve under actions and connect intelligent agents with their surroundings (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2020; 2021). Meanwhile, advances in diffusion transformers and large-scale video generation have made pretrained video models a natural foundation for visually rich world simulation (Peebles & Xie, 2023; Blattmann et al., 2023; Ma et al., 2024; Yang et al., 2025; Chen et al., 2024; Lin et al., 2024; Kong et al., 2024; Wan et al., 2025; MiniMax, 2026). Such models already support increasingly rich forms of interactive content, game simulation, and learned real-world simulation (Bruce et al., 2024; Yang et al., 2023; Che et al., 2025; Feng et al., 2026; Zhang et al., 2025; Sun et al., 2025).

Yet generation is not control. A pretrained video model does not, by itself, expose the precise action interface required for interactive world modeling. Existing systems therefore introduce new control pathways through discrete action embeddings, conditional modules, camera geometry, or direct backbone adaptation (Kim et al., 2020; Valevski et al., 2025; Alonso et al., 2024; Yu et al., 2025; Team et al., 2026; Gao et al., 2026; He et al., 2025; Wang et al., 2026b; Jiang et al., 2026). These pathways are typically learned from temporally aligned action-video trajectories. They require additional supervision and adaptation on top of the pretrained generator. They can also increase computation and storage, and extensive adaptation may disturb capabilities acquired during pretraining. More fundamentally, this paradigm treats control as something that must be learned largely on top of the pretrained model.

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.



Figure 1: **Overview of H3-World.** (i) *Prior*: MiniMax-H3 already exhibits coarse zero-shot control of character and camera motion. (ii) *Insight*: keyboard states are converted into per-latent instructions and routed to corresponding video latents, reusing the pretrained control prior. (iii) *Rollout*: different instructions from the same initial frame produce distinct command-following futures. All of this comes from 8,000 gameplay clips and 0.199% trainable parameters.

But large video generators may already contain part of this bridge. Modern video models learn rich representations of objects, agents, motion, and interaction from large-scale video data (Blattmann et al., 2023; Yang et al., 2025; Kong et al., 2024; Wan et al., 2025; MiniMax, 2026). MiniMax-H3, for example, can already follow coarse textual instructions for character and camera motion (MiniMax, 2026). Such instructions produce roughly correct responses with realistic visual dynamics, even without action-conditioned training. This observation is central to our work: *language already acts as a coarse control interface for strong text-to-video models*.

Based on this observation, we propose H3-WORLD, an efficient framework that turns the 33B-parameter MiniMax-H3 into an interactive world model. Our idea is simple. Instead of learning a new control representation from scratch, we express control in the language space the model already understands. Accordingly, H3-WORLD converts character and camera actions into compositional textual instructions and injects them through the native text pathway of MiniMax-H3. To make these controls precise over time, we align each instruction with its corresponding video latent interval. We further introduce temporal attention routing to preserve this alignment and prevent control leakage across time. As a result, H3-WORLD directly reuses the pretrained language and generative capabilities of MiniMax-H3 without introducing a dedicated action-specific control module. This design requires only lightweight LoRA adaptation (Hu et al., 2022; Xie et al., 2023).

Surprisingly, this lightweight adaptation is sufficient to produce effective interactive control. With only 8,000 gameplay samples and 0.199% trainable parameters, H3-WORLD learns precise character and camera control while preserving the generative capabilities of the pretrained model. More importantly, the resulting control transfers beyond the training distribution, including unseen action compositions and distinct initial observations. Together, these results suggest that much of the structure needed for interactive control can already exist inside a sufficiently capable video generator. By grounding low-level actions into its native language interface, H3-WORLD provides a direct path from pretrained video generation to interactive world modeling.

In summary, our contributions are as follows.

- We show that large video generators already support coarse language-based control, providing a strong basis for interactive world modeling.
- We introduce H3-WORLD, which turns the language understanding of large video models into world control. It represents character and camera actions as textual instructions and uses temporal attention routing for precise control, without dedicated action-specific modules.
- With only 8,000 samples and 0.199% trainable parameters, H3-WORLD achieves effective control while preserving generation quality and generalizing to unseen actions and visual scenarios.

2 Related Work

2.1 Action-Conditioned Video World Models

Interactive world models generate visual futures in response to user actions. Earlier neural and diffusion game simulators condition visual dynamics on recorded actions (Kim et al., 2020; Valevski et al., 2025; Alonso et al., 2024), while Genie learns a latent action interface from video (Bruce et al., 2024). Recent systems extend video generators toward open worlds, device control, causal rollout, persistent history, and real-time inference (Che et al., 2025; Feng et al., 2026; Zhang et al., 2025; Yu et al., 2025; Decart et al., 2024; Team et al., 2026; Gao et al., 2026; He et al., 2025; Wang et al., 2026b; Sun et al., 2025; Jiang et al., 2026). Related models study reactive characters, cross-game interaction, and explicit state-aware behavior (Wang et al., 2026a; Tong et al., 2026; Lin et al., 2026; Deng et al., 2026). These methods typically establish control through learned embeddings, feature modulation, camera geometry, or dedicated modules. Such interfaces require additional control learning and may limit robustness and generalization (Li et al., 2026; Shen et al., 2026). In contrast, H3-World injects fine-grained controls directly into the textual prompt, allowing them to be interpreted through the language understanding already acquired during video pretraining. This directly reuses the pretrained semantic prior, requires adapting only 0.199% of the parameters (MiniMax, 2026; Hu et al., 2022; Xie et al., 2023), and preserves strong generalization to unseen scenarios.

2.2 Language Actions and Temporal Grounding

Recent work uses language to express richer interventions and world-model actions (Yang et al., 2023; Tang et al., 2025; Mao et al., 2026; Xiong et al., 2026; Zhu et al., 2026; Ouyang et al., 2026). Incantation is most closely related, representing actions with per-latent-frame natural-language instructions and local text cross-attention (Zhu et al., 2026). Beyond world models, controllable video generation uses trajectories, camera poses, and compositional conditions to direct object and viewpoint motion (Wang et al., 2023; 2024; He et al., 2024; Yang et al., 2024; Yin et al., 2023; Villegas et al., 2022; Cai et al., 2025; Chefer et al., 2023). These works establish semantic conditioning as a useful control interface. H3-World considers a different architectural setting in which semantic conditions and video latents coexist within MiniMax-H3's packed single-stream self-attention. The model jointly denoises the full future horizon without a separate text cross-attention layer, making temporal grounding of language actions within this bidirectional sequence important.

3 Method

3.1 Overview

Given an initial observation I_0, a static semantic condition s, and a scheduled action sequence $\mathbf{a}_{1:K}$, H3-World generates the corresponding future video latents $\mathbf{V}_{1:K}$. We build on MiniMax-H3 (MiniMax, 2026), a pretrained bidirectional audio-video foundation model that jointly denoises the complete future horizon. Its pretrained parameters θ remain frozen, and ϕ denotes the lightweight adaptation parameters learned for action control. The conditional generation process is written as

$$\widehat{\mathbf{V}}_{1:K} \sim p_{\theta,\phi}\left(\mathbf{V}_{1:K} \mid I_0, s, \mathbf{a}_{1:K}\right). \tag{1}$$

MiniMax-H3 processes text, image, audio, and video tokens in a shared sequence. We retain the native audio stream but restrict the following notation to visual action control.

As shown in Figure 2, H3-World contains three components. The semantic action interface converts character and camera controls into compositional text instructions. Latent-aligned temporal binding associates each instruction with a video latent interval. Single-egress routing preserves this association throughout H3, while LoRA learns the action-conditioned visual dynamics.

3.2 Semantic Action Interface

External controls are represented as discrete control states. Each state contains eight recorded character and camera keys and one binary camera-speed flag. During training, the speed flag is derived from the estimated camera yaw rate. At inference, it is specified directly by the user. Each

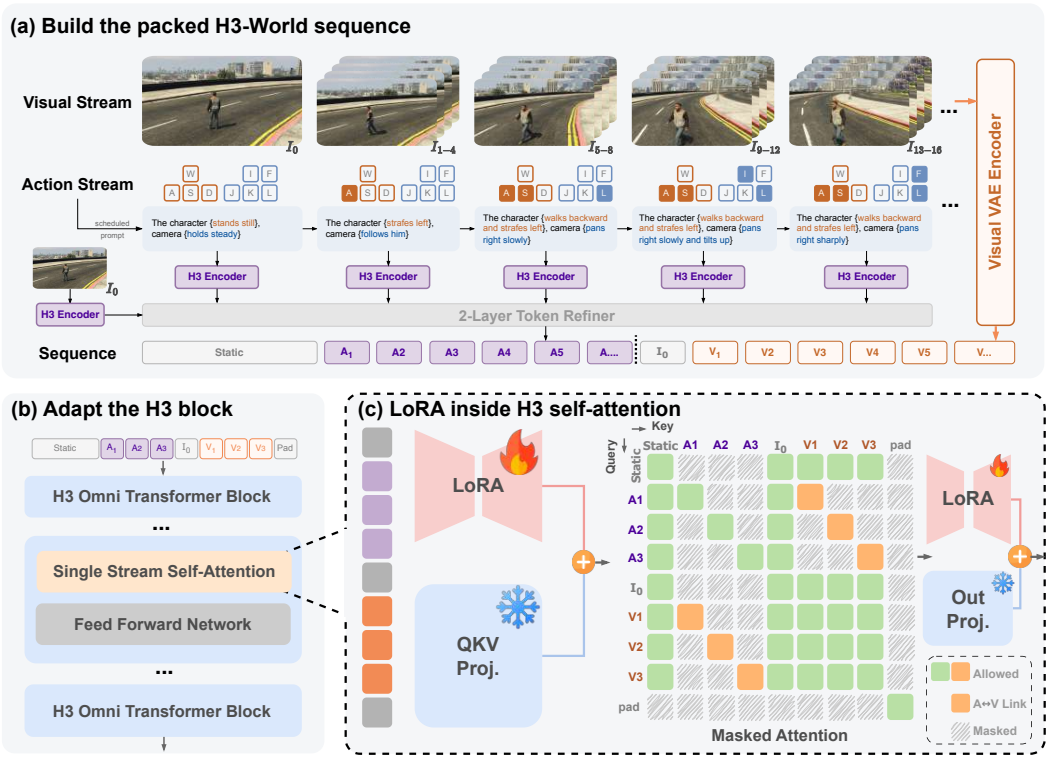

Figure 2: **Architecture of H3-WORLD.** (a) Latent-aligned action prompts are independently encoded and packed with the static semantic condition, initial observation, and video latents. (b) MiniMax-H3 processes the packed sequence through single-stream self-attention. (c) LoRA adapts the attention projections under single-egress routing, which connects each action span directly to its matched video latent and retains bidirectional attention among video latents.

native H3 video latent represents a short interval of RGB frames. We aggregate key states within each interval into one latent-level state, marking a key active if it occurs in any frame and cancelling opposing keys before prompt construction.

These states are temporally precise, but their discrete symbols carry little semantic information about the resulting visual motion. We expose the compositional structure of the control space by separating character control from camera control. For the k-th video latent interval, the action is written as

$$\mathbf{a}_k = (u_k, c_k), \qquad u_k \in \mathcal{U}, \qquad c_k \in \mathcal{C}, \tag{2}$$

where $\mathcal{U}$ contains the character-control commands and $\mathcal{C}$ contains the camera-control commands. The resulting action space contains nine character clauses and sixteen camera clauses.

We map each control pair to a short textual instruction

$$p_k = \mathcal{T}_{\text{char}}(u_k) \,\|\, \mathcal{T}_{\text{cam}}(c_k), \tag{3}$$

where $\|$ denotes clause concatenation. For example, a backward-left character command combined with a slow right-pan camera command produces the prompt "the man walks backward and strafes left, camera pans right slowly." The character and camera clauses follow a shared grammatical structure across the dataset. This representation places external controls in the native text-conditioning space of H3 and preserves the factorization of the original action space.

The action space is compact, but the training data cover only a sparse and highly imbalanced subset of the valid combinations. Figure 3 summarizes the joint distribution of character and camera clauses in the training set. The Cartesian product contains $9 \times 16 = 144$ combinations, of which 135 are structurally valid under our control specification. The training data covers 83 valid combinations and leaves 52 valid combinations unseen. The observed support therefore satisfies

$$\mathcal{A}_{\text{train}} \subsetneq \mathcal{A}_{\text{valid}} \subsetneq \mathcal{U} \times \mathcal{C}. \tag{4}$$

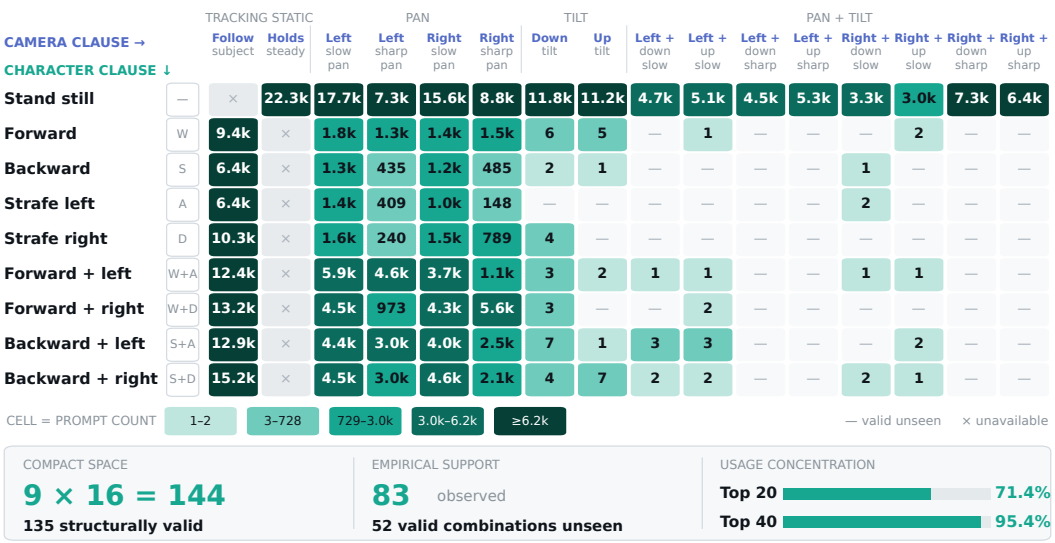

Figure 3: **Training coverage of the action space.** Of 135 valid character–camera pairs, 83 occur in 296,000 prompts and 52 remain unseen. Cells show prompt counts by quintile; dashes denote unseen pairs and crosses unavailable ones. The top 20 and 40 pairs account for 71.4% and 95.4% of prompts.

The observed distribution is highly imbalanced: the top 20 and 40 combinations account for 71.4% and 95.4% of all prompts, respectively, leaving a long tail of infrequent valid pairs. This sparse joint support provides a natural setting for compositional generalization, since an unseen pair can still consist of character and camera clauses observed separately during training.

The semantic action interface specifies *what* visual transition to produce; time-varying control also requires specifying *when* each instruction should apply. This grounding becomes essential when controls change within a rollout. We establish this correspondence in the H3 token sequence.

3.3 Latent-Aligned Temporal Binding

A single video-level prompt provides only a shared semantic condition for the entire clip, which is insufficient when the requested action changes over time. We therefore assign an independent action prompt p_k to each native video latent interval. Each prompt is encoded by the shared H3 encoder $\mathcal{E}$ and then processed by a shared two-layer token refiner $\mathcal{R}$

$$\mathbf{A}_k = \mathcal{R}\left(\mathcal{E}(p_k)\right). \tag{5}$$

Here, $\mathbf{A}_k$ denotes the action span associated with the k-th temporal interval. The token refiner applies block-diagonal attention over the scheduled prompts. Tokens within the same action span interact bidirectionally, while different action spans are processed independently. This preserves the temporal identity of each instruction before the action tokens enter the H3 backbone.

The initial observation is encoded through two complementary paths. The multimodal H3 encoder jointly processes the static semantic condition s and the initial observation I_0 to produce static semantic tokens $\mathbf{S}$. In parallel, the visual VAE maps I_0 to a first-frame condition $\mathbf{C}_0$ that preserves fine-grained appearance. The target video is encoded into latent intervals $\mathbf{V}_{1:K}$. During training, these intervals contain the noised target latents used by the native H3 denoising objective.

We then pack the semantic context, action spans, visual condition, and target video latents into a single H3 sequence

$$\mathbf{X} = [\mathbf{S}; \mathbf{A}_1; \ldots; \mathbf{A}_K; \mathbf{C}_0; \mathbf{V}_1; \ldots; \mathbf{V}_K; \mathbf{P}], \tag{6}$$

where $\mathbf{P}$ denotes masked padding tokens. Under the native temporal partition of the H3 VAE, each action span $\mathbf{A}_k$ is paired with its corresponding video latent $\mathbf{V}_k$. This establishes an explicit action–video correspondence at the same temporal granularity used by the video representation.

We further encode this correspondence through temporal positions. Let $\tau(\cdot)$ denote the temporal coordinate used by the H3 positional encoding. Each action span is assigned a position that follows

the temporal order of its matched video latent while remaining in the text-side positional range. For every matched pair,

$$\tau(\mathbf{A}_k) = \tau(\mathbf{V}_k) - \Delta, \qquad \Delta > 0. \tag{7}$$

All action–video pairs therefore share the same relative temporal offset. This preserves the pretrained text-before-video ordering of H3 while providing a consistent positional cue that explicitly links each action span to its matched video interval.

Temporal alignment alone does not constrain information flow in H3's bidirectional self-attention. An action span can still directly influence video latents outside its matched interval.

3.4 Single-Egress Routing and Low-Rank Adaptation

We construct a deterministic single-egress routing mask that gives each action span $\mathbf{A}_k$ a unique path to its matched video latent $\mathbf{V}_k$. This preserves temporal correspondence while preventing direct influence on unmatched video latents. Unlike positional alignment, routing explicitly constrains how action information enters the video stream.

As an attention key, $\mathbf{A}_k$ is visible only to itself and its matched video latent $\mathbf{V}_k$. As a query, it accesses the static context, first-frame condition, native audio context, itself, and $\mathbf{V}_k$, while other action spans and unmatched video latents remain masked. This asymmetric restriction prevents direct cross-interval action leakage while preserving the shared context needed to interpret each instruction.

After entering through $\mathbf{V}_k$, the control signal propagates across the generation horizon through H3's bidirectional video attention. This preserves motion continuity and scene consistency while localizing each action to its matched video latent. We call this *single-egress routing*.

The routing mask and span partition introduce no learnable parameters. LoRA (Hu et al., 2022) learns the action-conditioned transformations along the permitted attention paths. We apply low-rank updates to the QKV and output projections of the H3 self-attention blocks and to the two-layer token refiner. The pretrained H3 backbone, H3 encoder, visual VAE, and all remaining components remain frozen, while training retains the native H3 denoising objective.

4 Experiments

4.1 Experimental Setup

Data. We construct the training and evaluation sets from ABot-World-Explorer-500h (Jiang et al., 2026). The training set contains 8,000 gameplay clips, while a separate set of 150 clips is held out for evaluation. The two sets are disjoint at both the clip and episode levels, preventing direct overlap between training and evaluation episodes. Each clip contains 124 frames at 24 fps with a resolution of 832×480. Following Section 3.3, each clip provides 37 latent-aligned action prompts.

Training and inference. We train rank-32 LoRA parameters for 10,000 optimization steps with a learning rate of 1×10^{-4} on four GPUs. At inference, we generate 124 frames with 50 denoising steps. Pretrained H3 comparisons use the same settings with LoRA disabled.

Compared interfaces. We compare four interfaces trained on the same clips for the same number of steps. SolarWM (Huang et al., 2026) encodes camera pose and injects it into the backbone. ReactiveGWM (Wang et al., 2026a) and FiLM (Perez et al., 2018) instead map keyboard states to learned vectors that are added to, or used to modulate, video features. H3-World represents each action as per-latent text and delivers it through the pretrained text pathway.

Evaluation protocol. We evaluate whether commanded actions are reflected in the generated video. An inverse dynamics model D (Baker et al., 2022) predicts the nine-dimensional binary action state from a generated video $\mathbf{V}$. Let $\mathbf{a}_k$ and $\hat{\mathbf{a}}_k$ denote the commanded and predicted binary labels for control dimension k over the evaluation set. For each control group g, we report the mean per-dimension F_1,

$$\mathrm{Ctrl}(g) = \frac{1}{|g|} \sum_{k \in g} F_1(\mathbf{a}_k, \hat{\mathbf{a}}_k)\,, \tag{8}$$

where g covers character movement, camera yaw, camera pitch, and pan speed. We additionally report empirical ceilings from recorded clips and floors from permuted control labels. The accuracy of D is reported in Appendix B.1. We use two evaluation protocols. For *controlled intervention*, we fix the initial observation, seed, and sampling configuration and vary only the scheduled action. For *held-out rollout*, we condition H3-WORLD on the initial observation and recorded action sequence, using the recorded actions as ground-truth control labels.

4.2 ACTION CONTROLLABILITY

Table 1: **Action controllability on 150 held-out clips.** Recorded gameplay provides the empirical ceiling and permuted control states the floor; the best method per channel is shown in **bold**.

Method	Trainable ↓	Movement ↑	Yaw ↑	Pitch ↑	Fast ↑
Recorded gameplay *(ceiling)*	–	0.780	0.945	0.866	0.895
H3-WORLD	65.6M	**0.534**	**0.850**	0.416	**0.758**
SolarWM (Huang et al., 2026)	2,075M	0.353	0.817	**0.481**	0.705
ReactiveGWM (Wang et al., 2026a)	68.0M	0.344	0.278	0.133	0.316
FiLM (Perez et al., 2018)	70.4M	0.383	0.312	0.209	0.384
Permuted control states *(floor)*	–	0.221	0.284	0.083	0.212

Table 1 shows that H3-WORLD provides strong and parameter-efficient control without explicit camera-pose supervision. Despite using only 3.2% of SolarWM's trainable parameters, H3-WORLD achieves the highest scores among learned interfaces on movement, yaw, and pan speed, while SolarWM performs better on pitch. On yaw, for example, H3-WORLD reaches 0.850 compared with 0.817 for SolarWM. These results indicate that the pretrained text pathway can serve as an effective control interface without requiring explicit geometric inputs or additional pose supervision.



Figure 4: **Action-conditioning interfaces on a held-out clip.** GT is the recorded reference motion. ReactiveGWM-style additive conditioning and FiLM respond weakly or inconsistently as the controls vary, while H3-WORLD produces coordinated character and camera changes.

Qualitative results support the same trend. Figure 4 shows that direct feature-space conditioning responds weakly or inconsistently as actions change, whereas H3-WORLD produces coordinated character and camera motion. Controlled interventions in Figure 5 further isolate the effect of the command. With the initial observation and sampling seed fixed, opposite actions produce distinct trajectories, while faster camera commands induce larger viewpoint changes. Additional rollouts with recorded action sequences are provided in Appendix C.1.



Figure 5: **Controlled action intervention.** Initial observation, seed, and sampling configuration are fixed across rows, so only the instruction differs. Opposite commands produce distinct lateral motion, while fast pans induce larger viewpoint changes than slow pans.

4.3 From Pretrained Prior to Temporal Control

We next examine how the pretrained motion prior of H3 contributes to the temporally localized control learned by H3-World. We compare three settings. **(i) Global Prompt** uses frozen H3 with a single global motion instruction. **(ii) Zero-LoRA** uses the per-latent interface with LoRA disabled. **(iii) H3-WORLD** uses the fully adapted model. The camera pans left for the first 15 temporal latents and right for the remaining 22. Appendix B.4 provides the quantitative flow analysis.

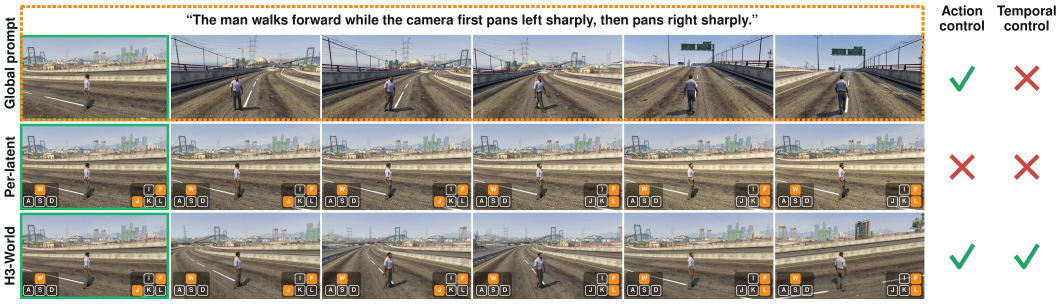

Figure 6: **Action response before and after adaptation.** Only H3-World follows both intervals; Global Prompt lacks temporal localization, while Zero-LoRA remains largely static.

Figure 6 separates sensitivity to motion instructions from temporally grounded control. Global Prompt exhibits coarse directional response but does not localize it to the requested interval, while the per-latent interface alone in Zero-LoRA produces little response. After adaptation, **H3-WORLD** follows both scheduled directions and reverses accordingly when their order is flipped.

A constant-action control in Appendix B.4 further isolates the role of adaptation. With a fixed camera direction, Global Prompt and H3-WORLD show comparable directional responses, confirming that pretrained H3 already contains a coarse motion prior. Together, these results show that adaptation is required to convert this prior into temporally grounded control.

4.4 Ablation Study

We ablate the two components responsible for temporal grounding. Removing *temporal binding* collapses the per-latent instructions into one clip-level instruction, eliminating explicit temporal assignment. Removing *routing* retains the per-latent instructions but allows every video latent to

attend to all of them, permitting direct information flow across temporal intervals. All variants use the same data, adapter size, and evaluation protocol.

Table 2 shows that both components are important for temporal control. Removing either component substantially degrades camera control, with yaw dropping by about 47–49% and pitch by 50–61%, while movement drops by only 14–18%. This contrast suggests that sustained locomotion is relatively tolerant to diffuse conditioning, whereas camera motion depends on precise temporal grounding.

Table 2: **Interface ablations.** Each variant removes one temporal-grounding component while retaining the same action vocabulary and 65.6M adapter.

Method	Mov. ↑	Yaw ↑	Pitch ↑	Fast ↑
H3-WORLD	**0.534**	**0.850**	**0.416**	**0.758**
w/o routing	0.438	0.448	0.209	0.473
w/o temporal binding	0.459	0.433	0.162	0.492

4.5 GENERALIZATION

Compositional generalization. Of the 135 structurally valid character–camera combinations, 83 appear in training and 52 remain unseen. We evaluate unseen combinations whose character and camera clauses each appear in training but never together. As shown in Figure 7a, H3-WORLD follows both the character and camera components of these unseen commands on held-out gameplay and out-of-distribution observations, demonstrating generalization beyond the observed joint support.

Visual generalization. We further evaluate the same action interface on six visually distinct initial observations. Figure 7b shows four representative examples across diverse viewpoints, environments, and rendering styles. H3-WORLD follows the requested controls while preserving scene structure, subject appearance, and visual style.



(a) Compositional action generalization. (b) Visual generalization to diverse initial observations.

Figure 7: **Generalization of the semantic action interface.** (a) H3-WORLD follows unseen character–camera compositions on both held-out gameplay and out-of-distribution observations. (b) The same interface transfers across four visual domains while preserving scene appearance.

5 CONCLUSION

We present H3-WORLD, an efficient framework that turns MiniMax-H3 into an interactive world model by expressing actions as compositional text instructions and temporally grounding them through single-egress routing. With lightweight LoRA adaptation, H3-WORLD reuses pretrained motion semantics to achieve effective character and camera control while generalizing to unseen action compositions and visual domains. These results suggest that language can serve as a practical interface for converting pretrained video generators into controllable world models. Our current study is limited to short, fixed-length rollouts and representative generalization tests; future work will extend H3-WORLD toward longer-horizon, persistent, and real-time interaction.

AI USE STATEMENT

In this work, we used generative AI tools only for language editing of the manuscript, i.e. improving grammar, wording, and readability of text drafted by the authors. We did not use generative AI tools for research ideation, for deriving the method, for designing or running the experiments, or for producing the reported results and figures. Code was written by the authors, with occasional AI-assisted autocompletion that was reviewed and tested by the authors. All AI-assisted text was reviewed by the authors, and we take full responsibility for the final content of this work, including all text, claims, and artifacts.

ETHICS STATEMENT

Our study uses publicly released gameplay video data (Jiang et al., 2026) and does not involve human subjects, personally identifiable information, or sensitive attributes. H3-WORLD adapts a pretrained video generator into an interactive world model; like other controllable video generation systems, it could in principle be misused to synthesize misleading visual content. Our released artifacts target game-style simulation and interactive world modeling research, and we inherit the usage restrictions of the underlying pretrained model and dataset licenses.

REPRODUCIBILITY STATEMENT

All components needed to reproduce our results are described in the main text. Section 3.2 specifies the action vocabulary and the prompt-construction rules, Section 3.3 defines the latent-aligned packing and mirrored temporal positions, and Section 3.4 defines the single-egress routing mask and the LoRA configuration. Section 4.1 lists the data source, train/eval split, clip resolution and length, LoRA rank, learning rate, optimization steps, sampling settings, and the optical-flow diagnostic used for quantitative comparisons. Our framework builds on the publicly available MiniMax-H3 checkpoint (MiniMax, 2026) and the publicly released ABot-World-Explorer-500h data (Jiang et al., 2026). Code and model weights will be released upon publication.

REFERENCES

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

Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654, 2022.

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

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

Minghong Cai, Xiaodong Cun, Xiaoyu Li, Wenze Liu, Zhaoyang Zhang, Yong Zhang, Ying Shan, and Xiangyu Yue. Ditctrl: Exploring attention control in multi-modal diffusion transformer for tuning-free multi-prompt longer video generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7763–7772. IEEE, 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.

Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM transactions on Graphics (TOG)*, 42(4):1–10, 2023.

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

Etched Decart, Quinn McIntyre, Spruce Campbell, Xinlei Chen, and Robert Wachen. Oasis: A universe in a transformer. *URL: https://oasis-model. github. io*, 2(3):6, 2024.

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.

Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pp. 363–370. Springer, 2003.

Ruili Feng, Han Zhang, Zhilei Shu, Zhantao Yang, Longxiang Tang, Zhicai Wang, Andy Zheng, Jie Xiao, Zhiheng Liu, Ruihang Chu, et al. The matrix: Infinite-horizon world generation with real-time moving control. *Advances in Neural Information Processing Systems*, 38:87318–87344, 2026.

Zelin Gao, Qiuyu Wang, Jiapeng Zhu, Jingye Chen, Zichen Liu, Qingyan Bai, Jiahao Wang, Yufeng Yuan, Hanlin Wang, Yichong Lu, et al. Infinite worlds with versatile interactions. *arXiv preprint arXiv:2607.07534*, 2026.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

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

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

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

Junchao Huang, Guian Fang, Shengju Qian, Xianghao Kong, Zhuoran Zhao, Wei Huang, Yihua Du, Zixin Zhang, Justin Cui, Yuchao Gu, Yukang Chen, Xinting Hu, Tianyu He, Shaoshuai Shi, Zhuotao Tian, Xin Wang, Mike Zheng Shou, and Li Jiang. Solarwm: Open data and scalable training for long-horizon video world models. *arXiv preprint arXiv:2609.02886*, 2026.

Fan Jiang, Zhaoxu Sun, Mengchao Wang, Ziyu Zhu, Chiyu Wang, Yunpeng Zhang, Wenlin Liu, Yun Wang, Xue Zheng, Rui Sun, et al. Abot-world-0: Infinite interactive world rollout on a single desktop gpu. *arXiv preprint arXiv:2607.19191*, 2026.

Seung Wook Kim, Yuhao Zhou, Jonah Philion, Antonio Torralba, and Sanja Fidler. Learning to simulate dynamic environments with gamegan. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1228–1237. IEEE, 2020.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Qi Li, Xingyi Yang, and Xinchao Wang. Badwam: When world-action models dream right but act wrong. *arXiv preprint arXiv:2607.15207*, 2026.

Bin Lin, Yunyang Ge, Xinhua Cheng, Zongjian Li, Bin Zhu, Shaodong Wang, Xianyi He, Yang Ye, Shenghai Yuan, Liuhan Chen, et al. Open-sora plan: Open-source large video generation model. *arXiv preprint arXiv:2412.00131*, 2024.

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.

Xin Ma, Yaohui Wang, Xinyuan Chen, Gengyun Jia, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024.

Xiaofeng Mao, Zhen Li, Chuanhao Li, Xiaojie Xu, Kaining Ying, and Kaipeng Zhang. Yume1. 5: A text-controlled interactive world generation model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7752–7761, 2026.

MiniMax. MiniMax H3. https://github.com/MiniMax-AI/MiniMax-H3, 2026. Open-weight omni-modal audio-video generation model. Accessed: 2026-08-28.

Liangyang Ouyang, Ruicong Liu, Xuangeng Chu, Kaipeng Zhang, and Yoichi Sato. Helloworld: Enabling socially interactive characters in video world models. *arXiv preprint arXiv:2608.05070*, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Linghui Shen, Mingyue Cui, and Xingyi Yang. Badworld: Adversarial attacks on world models. *arXiv preprint arXiv:2606.16519*, 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 modeling. *arXiv preprint arXiv:2512.14614*, 2025.

Junshu Tang, Jiacheng Liu, Jiaqi Li, Longhuang Wu, Haoyu Yang, Penghao Zhao, Siruis Gong, Xiang Yuan, Shuai Shao, Linfeng Zhang, et al. Hunyuan-gamecraft-2: Instruction-following interactive game world model. *arXiv preprint arXiv:2511.23429*, 2025.

Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, et al. Advancing open-source world models. *arXiv preprint arXiv:2601.20540*, 2026.

Zizhao Tong, Yeying Jin, Hongfeng Lai, Zeqing Wang, Zhaohu Xing, Kexu Cheng, Haoran Xu, Zhao Pu, Shangwen Zhu, Ruili Feng, et al. Scope: Simulating cross-game operations in playable environments for fps world models. *arXiv preprint arXiv:2605.23345*, 2026.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *International Conference on Learning Representations*, volume 2025, pp. 73754–73776, 2025.

Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022.

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.

Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. *Advances in Neural Information Processing Systems*, 36:7594–7611, 2023.

Zeqing Wang, Danze Chen, Zhaohu Xing, Zizhao Tong, Yinhan Zhang, Xingyi Yang, and Yeying Jin. Reactivegwm: Steering npc in reactive game world models. *arXiv preprint arXiv:2605.15256*, 2026a.

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, Kaichen Huang, Baixin Xu, Fei Kang, Mengyin An, Peiyu Wang, Biao Jiang, Yichen Wei, et al. Matrix-game 3.0: Real-time and streaming interactive world model with long-horizon memory. *arXiv preprint arXiv:2604.08995*, 2026b.

Enze Xie, Lewei Yao, Han Shi, Zhili Liu, Daquan Zhou, Zhaoqiang Liu, Jiawei Li, and Zhenguo Li. Difffit: Unlocking transferability of large diffusion models via simple parameter-efficient fine-tuning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4207–4216. IEEE, 2023.

Zhexiao Xiong, Yizhi Song, Hao Kang, Qing Yan, Liming Jiang, Jenson Yang, Zhoujie Fu, Stathi Fotiadis, Angtian Wang, Zichuan Liu, et al. Actworld: From explorable to interactive world model via action-aware memory. *arXiv preprint arXiv:2606.17730*, 2026.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

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*, pp. 1–12, 2024.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, volume 2025, pp. 83048–83077, 2025.

Shengming Yin, Chenfei Wu, Jian Liang, Jie Shi, Houqiang Li, Gong Ming, and Nan Duan. Dragnuwa: Fine-grained control in video generation by integrating text, image, and trajectory. *arXiv preprint arXiv:2308.08089*, 2023.

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

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

Shangwen Zhu, Qianyu Peng, Zhao Pu, Zhilei Shu, Xiangrui Ke, Zhaohu Xing, Zizhao Tong, Zeqing Wang, Xinyu Cui, Zian Zheng, et al. Incantation: Natural language as the action interface for multi-entity video world models. *arXiv preprint arXiv:2605.18601*, 2026.

A Implementation Details

This appendix records what is needed to rebuild the experiments of Section 4: how the clips were selected, how an action becomes text, how the action spans are placed in the H3 sequence, and how each compared interface was trained.

A.1 Dataset Construction

Clips are drawn from ABot-World-Explorer-500h (Jiang et al., 2026). We scan every episode into overlapping 124-frame windows and score each window by how much of it is spent inside sustained action segments, which we define as the fraction of the clip covered by segments of at least 1.5 seconds. This measure, rather than segment count or minimum segment length, is what identifies a held action: a player releases one key a few frames before pressing the next, so long holds are separated by short transitional states.

Selecting windows uniformly would reproduce the raw distribution of the source, in which 19% of windows contain no character motion at all and 45% hold the locomotion keys for less than half the clip. We therefore assign each window to an action category and fill a fixed quota per category, summing to the 8,150 selected windows: 3,400 walking, 2,000 turning, 1,300 camera-only, 1,000 walking-with-pitch, and 450 still or unconstrained, with the four cardinal directions balanced inside each category. Within a category, windows are ranked by sustained-action coverage. Both one-action and two-action windows are kept, so the model sees clips in which a single action is held and clips in which the action changes once; three sustained actions do not fit in 124 frames.

At most one window is taken per episode. This maximises scene diversity and makes the split episode-disjoint by construction. The 8,150 selected windows come from 8,150 distinct episodes, of which 8,000 form the training set and 150 the evaluation set, preventing direct episode overlap between training and evaluation. Every clip is 124 frames at 24 fps and 832×480, and contributes 37 latent-aligned action prompts, for $8{,}000 \times 37 = 296{,}000$ training prompts in total.

A.2 The Action Dictionary

Table 3 enumerates the action dictionary and gives the frequency of every clause over the 296,000 training prompts. Character motion is factorized into two directional axes, while camera control uses pan, tilt, and a binary speed modifier. Opposing directional keys cancel when simultaneously active, so each directional axis contributes three states. The resulting dictionary is small and fully enumerable rather than open-ended.

Table 3: **Action dictionary and training frequencies.** Percentages denote the fraction of the $8{,}000 \times 37 = 296{,}000$ per-latent training prompts in each state. The camera-speed distribution is conditioned on panning because the speed bit only qualifies a pan.

Axis	Neither / default (%)	Clause	%	Clause	%
Character slot	$3 \times 3 = 9$ clauses				
sagittal	53.2	`W` walks forward	24.5	`S` walks backward	22.6
lateral	53.7	`A` strafes left	21.8	`D` strafes right	24.6
both idle	*stands still* 45.3%				
Camera slot	16 clauses				
pan	44.1	`J` pans left	28.7	`L` pans right	27.5
tilt	78.8	`I` tilts down	10.7	`K` tilts up	10.5
speed	*slowly* 58.8	`F` sharply	41.2	*(pans only)*	
both idle	*follows him* 28.9%	*holds steady*	7.4%		
Dictionary	$9 \times 16 = 144$ nominal	$\rightarrow$ 135 structurally valid	$\rightarrow$	83 observed in training	

The camera-speed axis is the one control bit not read directly from the keyboard. During training it is derived from the estimated yaw rate: writing $\Delta\mathrm{yaw}_i$ for the per-frame yaw change inside latent

interval i, the fast-pan bit is

$$\mathtt{F}_i \;=\; \mathrm{pan}_i \;\wedge\; \big(|\Delta\mathrm{yaw}_i| \geq \tau_{\mathrm{fast}}\big), \qquad \tau_{\mathrm{fast}} = 0.225, \tag{9}$$

where pan_i indicates that a pan key is active. The bit only qualifies a pan, which is why its frequency in Table 3 is conditioned on panning. At inference the user sets it directly and Eq. 9 is not used.

The dictionary also explains the reduction in Eq. 4. The Cartesian product contains $9 \times 16 = 144$ character–camera pairs, of which nine are structurally invalid because their clauses conflict. The camera clause *follows him* presupposes character motion, whereas *holds steady* presupposes no character motion, so each is incompatible with character clauses that assert the opposite. This leaves 135 structurally valid pairs, of which 83 appear in the training set.

A.3 Latent Partition and Sequence Composition

A video latent does not correspond to a fixed number of RGB frames. The H3 VAE tiles the 124 generated frames with 37 latents in a repeating, non-uniform pattern. The first latent of every group of five covers one frame, while the remaining four cover four frames each:

$$\big|\mathrm{span}(k)\big| = \begin{cases} 1, & k \equiv 1 \pmod 5, \\ 4, & \text{otherwise}, \end{cases} \qquad k = 1, \ldots, 37. \tag{10}$$

Across 37 latents, this gives $7 \times 17 + 5 = 124$ frames in total. The key states inside $\mathrm{span}(k)$ are aggregated into the latent-level state that produces p_k, so the aggregation window contains four frames for most latents and one frame at each group boundary.

The mirrored temporal offset Δ of Eq. 7 is a single constant, $\Delta = 201$, shared by all 37 action–latent pairs. Action spans inherit the fractional, non-uniform temporal coordinates of their matched video latents rather than an integer index, so the pairing survives the non-uniform partition of Eq. 10 exactly.

Table 4: **Composition of the packed H3 sequence** for one training clip. Temporal binding replaces one 13-token clip-level instruction with 37 latent-aligned action spans, increasing the occupied sequence length from 15,813 to 16,259 tokens, a 2.8% overhead, without adding parameters. Counts are exact for a representative clip; action spans vary slightly with clause length.

Segment	Contents	Tokens	Note
$\mathbf{S}$	static semantic condition	566	prompt and initial observation
$\mathbf{A}_{1:37}$	action spans	459	8–14 each, mean 12.4
$\mathbf{C}_0$	first-frame condition	390	15×26 patches
$\mathbf{V}_{1:37}$	video latents	14,430	37×390
–	native audio	414	retained unmodified
Total occupied		16,259	padded to 16,448
w/o temporal binding		15,813	one 13-token clip instruction
Additive bias and FiLM		15,800	no action text

Table 4 gives the resulting sequence. The 37 action spans occupy 459 tokens, compared with 13 tokens for a single clip-level instruction. Because the remaining sequence is unchanged, temporal binding increases the total occupied length from 15,813 to 16,259 tokens, or 2.8%. The routing mask and span partition add no learnable parameters, so the ablation of Section 4.4 isolates a structural choice rather than a capacity difference.

A.4 Training and Inference

All models are trained with the native H3 denoising objective and identical optimisation settings. LoRA (Hu et al., 2022) of rank 32 is applied to the QKV and output projections of the H3 self-attention blocks and to the two-layer token refiner, giving 65.6M trainable parameters against a frozen 33B backbone, or 0.199%. We use AdamW at a learning rate of 1×10^{-4} with weight decay 0.01, one clip per device and no gradient accumulation, gradient checkpointing enabled, for 10,000 optimisation steps on four GPUs, checkpointing every 2,000 steps. The backbone, the H3 encoder, the visual VAE and the audio stream stay frozen throughout.

Text conditions are encoded once and cached before training, so the text encoder is not resident during optimisation and every variant sees exactly the same encoded conditions. At inference we generate 124 frames with 50 denoising steps at 832×480, with the same seed and sampling configuration for every method. Frozen-H3 comparisons use this configuration with the LoRA updates disabled.

A.5 Compared Interfaces

H3-World, Additive bias, and FiLM receive the same 37×9 per-latent control states, derived from the same recorded clips using the same aggregation. They differ only in how these controls enter the backbone. SolarWM instead receives camera geometry through its native pose-conditioning interface. All methods otherwise use the same evaluation clips and generation protocol where applicable.

H3-World renders each control state as text through Eqs. 2 and 3 and delivers it through the pretrained text pathway. Its 65.6M trainable parameters are the LoRA adapter alone; the temporal binding and the routing mask are deterministic constructions.

Additive bias, following ReactiveGWM (Wang et al., 2026a), maps the control state to a learned vector added to the video features of every block. We implement it as one embedding table per block, $50 \times 9 \times 5376 = 2.4$M parameters, zero-initialised so that training starts from the unmodified backbone. With the shared adapter this totals 68.0M.

FiLM (Perez et al., 2018) modulates the same features multiplicatively and additively, using a scale and a shift per block, $2 \times 50 \times 9 \times 5376 = 4.8$M parameters, also zero-initialised, for 70.4M in total.

SolarWM (Huang et al., 2026) conditions on camera geometry rather than on key states, consuming per-frame camera-to-world poses and intrinsics, and trains a rank-384 adapter over the attention and feed-forward projections, with 2,075M trainable parameters. All evaluation clips satisfy the admissible camera range required by its pose-conditioning interface.

B Evaluation Protocol

Section 4.1 defines controllability as the agreement between the commanded action and the action an observer recovers from the generated video. This appendix specifies the observer, the aggregation that turns its predictions into Eq. 8, the floor those numbers are read against, and the separate measurement used for temporal placement.

B.1 Inverse Dynamics Model

The controllability metric of Section 4.2 reads the commanded action back out of generated pixels with an inverse dynamics model (Baker et al., 2022). This appendix reports its accuracy, so that the ceiling and floor of the main table can be read against a known measurement error.

The model is a ResNet-18 over four grayscale frames sampled inside each latent interval, trained to predict the nine-bit action of that interval. It is trained on recorded gameplay disjoint from both the generation training set and the evaluation clips, with AdamW under a one-cycle schedule for eight epochs at batch size 256. Recovering an action from frames is easier than predicting one, because the task is non-causal and may use the frames on both sides of a transition, so an accurate observer can be fitted from far less data than a policy would need (Baker et al., 2022).

Two properties of Table 5 matter for the main results. First, accuracy is a misleading summary here: the two pitch keys fire on 4% of latents, so always predicting "not pressed" already scores 96%. We therefore report macro-averaged F_1, which weights every key equally and cannot be inflated by the negative class. Second, the pitch channel is the least reliable one ($F_1 = 0.799$ and 0.876), which may partly explain the smaller separation between methods in the pitch column of Table 1.

B.2 Controllability Score

The same class imbalance that rules out accuracy also dictates how the per-dimension F_1 of Eq. 8 is aggregated. We pool the confusion counts over every scored interval before forming the score, rather than scoring each clip and averaging: with the pitch keys active in 4% of latents, most clips contain

Table 5: Inverse dynamics model on held-out recorded gameplay. Per-key F_1 at threshold 0.5, grouped into four control groups: Character movement (W,A,S,D), Camera yaw (J,L), Camera pitch (I,K), and Pan speed (F).

Control channel	Keys	Positives	Mean $F_1 \uparrow$	Range
Camera yaw	J, L	3,441	0.972	0.971–0.972
Pan speed	F	1,088	0.910	–
Character movement	W, A, S, D	5,654	0.859	0.822–0.895
Camera pitch	I, K	459	0.838	0.799–0.876
Overall	all nine	–	0.885	mean accuracy 0.957

no positive instance of a given key, and a per-clip F_1 is undefined there. Writing $a_k(c,i) \in \{0,1\}$ for the commanded state of dimension k in latent i of clip c, and $\hat{a}_k(c,i)$ for the prediction of D at threshold 0.5,

$$\mathrm{TP}_k = \sum_{c,i} a_k \hat{a}_k, \qquad \mathrm{FP}_k = \sum_{c,i} (1 - a_k)\, \hat{a}_k, \qquad \mathrm{FN}_k = \sum_{c,i} a_k (1 - \hat{a}_k), \tag{11}$$

$$F_1(\mathbf{a}_k, \hat{\mathbf{a}}_k) \;=\; \frac{2\,\mathrm{TP}_k}{2\,\mathrm{TP}_k + \mathrm{FP}_k + \mathrm{FN}_k}. \tag{12}$$

The sums in Eq. 11 run over all 150 evaluation clips and all 37 latents, that is 5,550 scored intervals per control dimension. Substituting Eq. 12 into Eq. 8 gives the four group scores of Table 1 and Table 2. Where a single summary is quoted we use the macro average over all nine dimensions,

$$\text{macro-}F_1 \;=\; \frac{1}{9} \sum_{k=1}^{9} F_1(\mathbf{a}_k, \hat{\mathbf{a}}_k)\,, \tag{13}$$

which is not the mean of the four $\mathrm{Ctrl}(g)$: that would weight the single-key pan-speed group as heavily as the four-key movement group.

B.3 Permuted Control States

A controllability score has no absolute zero. A model that ignores its instructions entirely still scores above zero, because the rate at which each key is pressed already predicts part of the label sequence. The floor row of Table 1 measures that residual. We rescore each generation against the control states of a *different* clip, obtained by rotating the clip index under a permutation σ_r, and average over $R = 5$ rotations:

$$\mathrm{Floor}(g) \;=\; \frac{1}{R} \sum_{r=1}^{R} \mathrm{Ctrl}(g\,;\, \hat{\mathbf{a}}(c),\, \mathbf{a}(\sigma_r(c)))\,. \tag{14}$$

We use non-zero cyclic rotations so that every generation is paired with the control sequence of a different clip while preserving each key's marginal rate exactly. Averaging over $R = 5$ rotations reduces dependence on any particular mismatched pairing. Eq. 14 therefore measures the residual score obtainable without the correct generation–control correspondence while preserving the marginal control statistics. The resulting floor depends on the control group, ranging from 0.284 for camera yaw, whose keys are active in most clips, to 0.083 for camera pitch. A fixed baseline such as 0.5 would therefore misstate the difficulty of each column.

B.4 Temporal Response Measurement

Section 4.3 asks whether a camera command is executed *at the requested time*, which controllability alone cannot answer. We therefore measure horizontal image motion directly. For a clip with frames $\{I_t\}_{t=1}^{T}$, let u_t be the horizontal component of the dense optical flow (Farnebäck, 2003) from I_t to I_{t+1}, and accumulate its spatial mean over a frame interval $[a, b)$:

$$S(a,b) \;=\; \sum_{t=a}^{b-1} \frac{1}{|\Omega|} \sum_{p \in \Omega} u_t(p), \tag{15}$$

where Ω is the image domain. A camera panning right moves scene content left, so a rightward command requires $S < 0$ and a leftward command $S > 0$. Writing t_s for the frame at which a schedule reverses, the pair $\big(S(1, t_s),\ S(t_s, T)\big)$ states both what the model did and when it did it: following a reversing schedule requires the two values to carry opposite signs in the commanded order, which no single-interval statistic can verify.

Table 6: **Action response before and after adaptation.** The camera schedule reverses at the boundary after latent 15. *Responds* measures directional separation when one camera direction is held throughout. The remaining columns report Eq. 15 before and after the switch, with the required sign shown in the header.

Method	Responds	Left, then right		Right, then left	
		Before (+)	After (−)	Before (−)	After (+)
Recorded gameplay	–	+118.6	−136.3	−68.0	+141.2
Global Prompt	301.8	0.0	−17.3	−11.9	+24.1
Zero-LoRA	–	−0.1	0.0	Unresponsive	
H3-WORLD	300.5	+52.7	−106.0	−58.7	+121.0

Table 6 reports the measurement. *Responds* is the directional separation when one camera direction is held throughout the clip, which isolates whether the model reacts to a motion instruction at all; the remaining columns are Eq. 15 before and after the switch, with the sign the schedule requires given in the header.

C ADDITIONAL QUALITATIVE RESULTS

The figures below are the full versions of the comparisons condensed in Section 4.2 and Section 4.5.

C.1 ROLLOUTS WITH RECORDED ACTION SEQUENCES

Figure 8 compares generations against recorded gameplay under the same initial observation and the same recorded action sequence. This is the setting scored by the controllability metric of Table 1: the recorded actions supply the control labels, and Eq. 12 is accumulated over exactly these rollouts.

C.2 GENERALIZATION

Figure 7 condenses the two generalization studies of Section 4.5. Their full versions follow. Figure 9 gives every compositional pair, and Figure 10 gives all six initial observations, each receiving both a character-control and a camera-control command through the same interface.



Figure 8: **Following recorded controls on held-out clips.** Each pair shares the same initial observation and recorded action sequence, with ground truth above and H3-WORLD below.

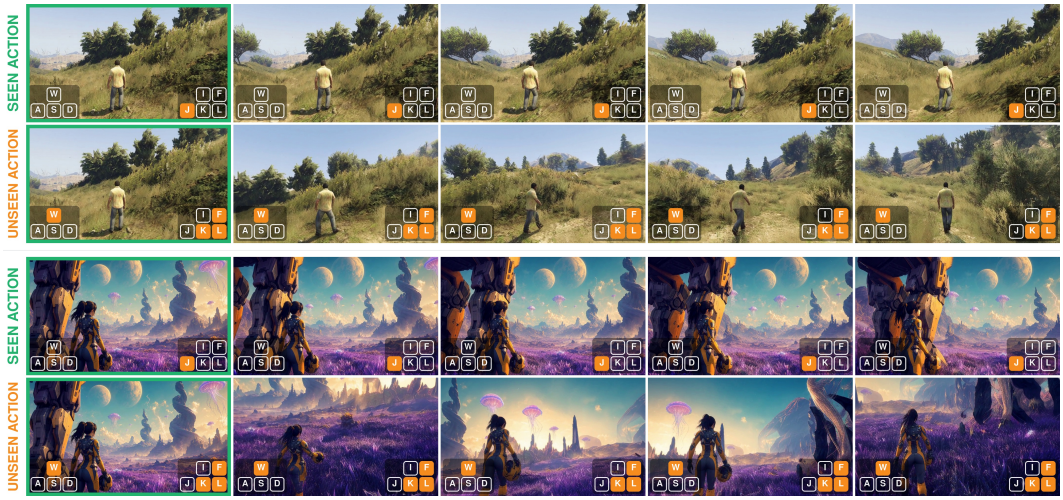

Figure 9: **Compositional action generalization.** Each group pairs a seen action with an unseen composition of observed character and camera clauses on held-out gameplay (top) and out-of-distribution observations (bottom).

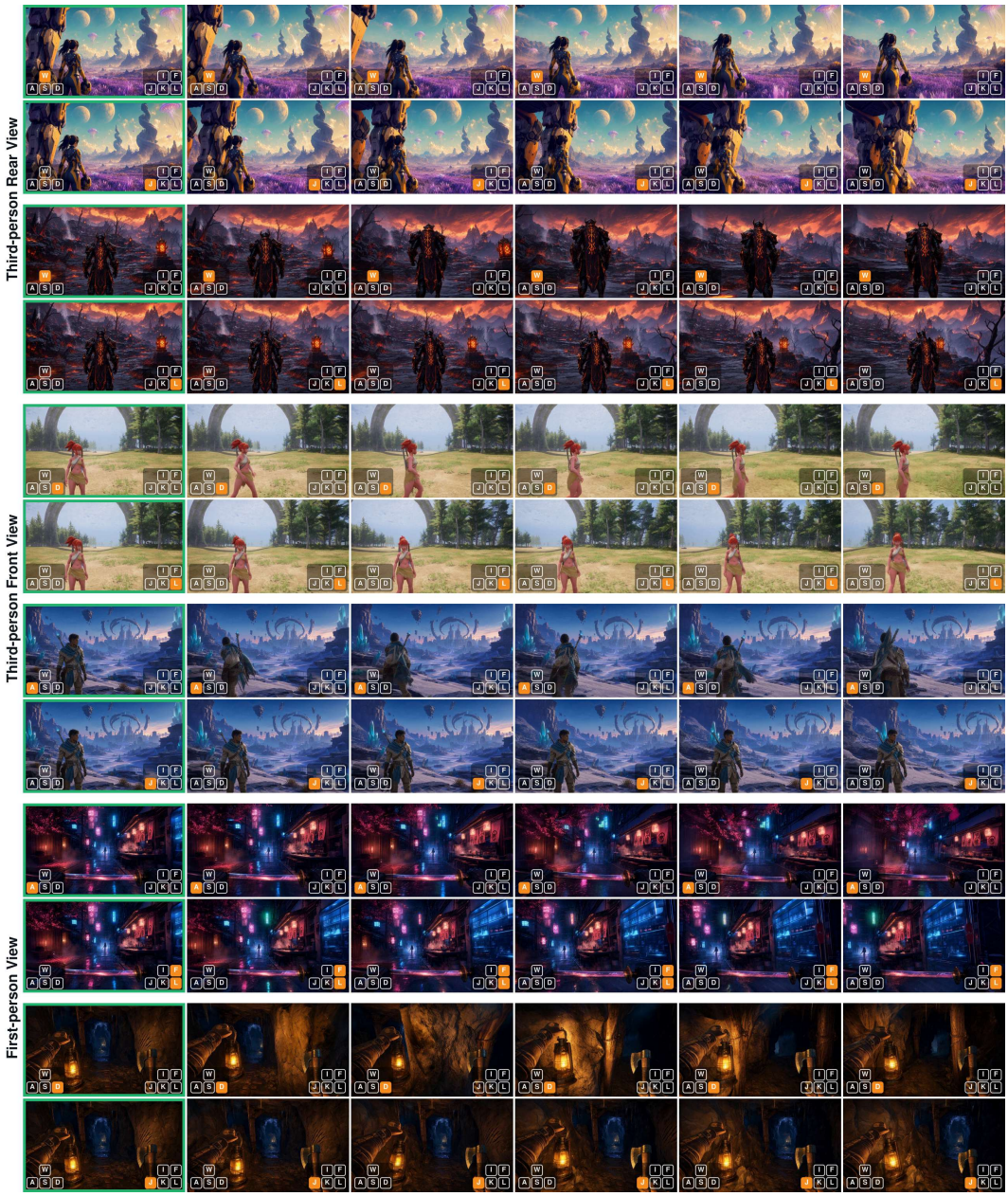

Figure 10: **Visual generalization to diverse initial observations.** Each scene receives one character-control command and one camera-control command through the same learned interface, without per-domain tuning.