Incantation: Natural Language as the Action Interface for Multi-Entity Video World Models

Anonymous Author(s)
Affiliation
Address
email

Figure 1: **Demonstrations of Incantation's cross-entity action transfer and multi-entity control in the game *Elden Ring*. *(i)*** Two bosses, **Margit** and **Crucible Knight**, each possessing character-exclusive moves, are conditioned via natural language to perform each other's actions, each executed by both its native character and the other: ***Light Blade Attack*** (Margit-exclusive, green rows) and ***Tail of the Crucible*** (Crucible Knight-exclusive, blue rows), demonstrating **Incantation**'s cross-entity generalization. ***(ii)*** **Incantation** simultaneously controls three entities (two bosses and one player) each via a distinct natural-language prompt (orange rows), trained on two-entity scenarios only.

Abstract

Modern interactive video world models have achieved impressive visual fidelity, yet lack fine-grained multi-entity control and cross-entity, cross-world generalization. We trace this gap to the *action interface*: standard control protocols (e.g. animation IDs, device inputs, scene-level captions) bind action semantics to specific entities or engines at design time. We propose *natural language as the interface* to unlock expressiveness that no prior interface can achieve, and we present **Incantation**, the first interactive video world model with per-latent-frame (0.25 s) natural-language conditioning that supports simultaneous multi-entity control and *concept-level* cross-entity transfer beyond any fixed rendering pipeline. We pair a pretrained bidirectional video backbone with frame-local text cross-attention, and enable real-time long-horizon streaming through ODE-initialized Self-Forcing distillation with a RoPE-decoupled sliding KV-cache. We surpass the Action-Index baseline on cross-entity transfer (89% vs. 43%) and out-of-vocabulary prompts (90% vs. 0%), and our 2-step student sustains 19.7 FPS at 480p with stable FVD over 2-hour rollouts. We further apply the same architecture and training recipe to *The King of Fighters*, changing only the per-entity action vocabulary slots. We will release code, checkpoints, and a 128-hour frame-accurate multi-entity action dataset, establishing natural language as a general, scalable action interface for video world models.

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

1 Introduction

Modern video diffusion models [21, 6, 39] have driven a growing line of controllable interactive world models [8, 28, 15, 2, 13, 48, 20, 40, 11] to near-cinematic fidelity, yet every such system inherits a structural limitation from the rendering pipelines it replaces: actions are bound to engine-internal animation namespaces or device-level inputs, locking action semantics to a specific entity and engine. This entity-and-engine binding forces a separate action vocabulary to be designed for every entity in every world, making cross-entity and cross-world generalization an engineering burden rather than a modeling choice. We argue that this is not an intrinsic property of multi-entity interactive video, but a property of the *action interface*: the protocol through which a user specifies what should happen on the next frame, and replacing it fundamentally expands what such a model can express.

This bottleneck dominates in the *single-viewpoint multi-entity* regime: a shared camera with two or more independently controllable entities, as in RPG (Role-Playing Game) combat and PvP (Player vs. Player) fighting. This regime is central to competitive and adversarial gameplay, yet remains structurally underserved by interactive video world models. Most controllable interactive world models confine control to a single entity, leaving the rest as passive scenery [8, 11, 20, 16], while recent multi-entity attempts sidestep the regime by dropping joint dynamics [29], abandoning the shared camera [31], or controlling only one side [1]. None of these approaches admits a protocol with both fine-grained multi-entity control and generalization across entities and worlds. This shortfall ultimately traces back to the *action interface* itself, which exhibits two conventional failure modes: **(1) Engine-internal animation labels** (per-world discrete IDs [2] and per-entity namespaces) bind each index to a specific animation at design time, so rendering any out-of-vocabulary (OOV) action is inherently inexpressible. **(2) Human-device inputs** [8, 28, 15, 11, 20, 16, 47, 40] and **scene-level captions** [9, 37, 38] operate at the granularity of the player or the holistic scene rather than the individual entity, thus lacking the critical per-entity addressability (e.g., non-player characters). A viable multi-entity interface must therefore deliver both *open-vocabulary semantics* for cross-entity semantic sharing and *per-entity addressability* for independent, simultaneous control of each entity.

To address this limitation, we propose *a per-entity natural-language action interface* as the first to satisfy both desiderata, and present **Incantation**, the first interactive video world model supporting independent and simultaneous multi-entity control under a single shared viewpoint via per-frame natural-language conditioning (**Throughout this paper, "frame" denotes a VAE-compressed latent frame unless otherwise specified; 1 latent frame corresponds to 4 pixel frames along the temporal axis; FPS denotes end-to-end pixel-frame throughput**). Our interface assigns each entity its own syntactically isolated text segment within a shared prompt template at 0.25 s temporal granularity, enabling concurrent yet independent control of all entities. Natural language shares semantics across entities by construction, inherently allowing any action to be transferred from its native entity to another via a single textual phrase (Figure 1). We term this *concept-level* cross-entity transfer: the model must synthesize both the motion and the visual concept on an entity that has no recording of the action, a capability inherently inaccessible to rendering pipelines bound to per-entity animation namespaces. To our knowledge, no prior interactive video world model has explicitly addressed cross-entity action transfer at the level of per-frame, per-entity conditioning.

Incantation realizes this interface on top of a pretrained bidirectional video diffusion backbone [39]. The core design is a **per-frame language-conditioned attention scheme**: decoupled text cross-attention is restricted exclusively to the noisy target frame and applied on top of bidirectional history self-attention, so each frame is steered by exactly its own action prompt without disturbing the backbone's pretrained priors or contaminating the committed history. We further enable real-time streaming inference by coupling ODE-initialized Self-Forcing distillation [23] with a RoPE-decoupled KV-cache sliding window, which collapses inference to two steps and keeps memory and positional geometry bounded over indefinite horizons.

Extensive experiments have demonstrated the structural advantage of **Incantation**'s natural-language interface. On cross-entity prompts (actions issued to entities that never executed them in training), **Incantation** attains 89% Action Control Accuracy (ACA), far exceeding the 43% of an Action-Index baseline whose accuracy merely tracks visual similarity rather than the action label itself. The gap widens to 90% versus 0% on OOV prompts, since the Action-Index interface cannot accept any prompt outside its fixed vocabulary. Besides its fine-grained per-frame control, **Incantation** sustains real-time long-horizon generation at 19.7 FPS with stable visual quality over $\mathbf{2}$-hour sessions, and replicates the performance on the visually unrelated *King of Fighters* (KOF) world merely

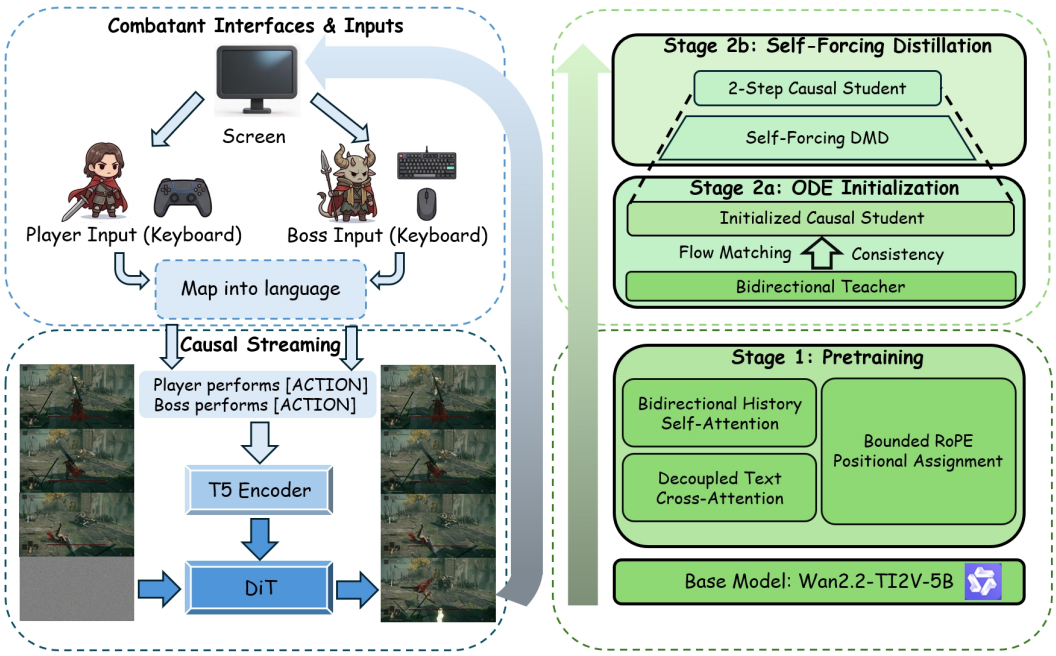

Figure 2: **Workflow of Incantation. Left: Incantation** translates combatant keyboard inputs into natural language prompts and autoregressively generates video frames in a causal streaming manner. **Right:** Training proceeds in two stages: (1) Language-Conditioned Pretraining adapts the base model for per-frame text-driven generation; (2) ODE-Initialized Self-Forcing Distillation enables real-time streaming via ODE-based flow matching initialization followed by Self-Forcing distillation.

by vocabulary substitution alone, further validating its cross-world generalization capability. Our contribution can be summarized as follows:

1. We propose **natural language as the action interface for multi-entity video world models**, the first per-entity parallel control regime with open-vocabulary semantics, and demonstrate two structural capabilities unavailable to any discrete action-index, device-input, or scene-caption interface by construction: *cross-entity action transfer* and *out-of-vocabulary coverage*.
2. We present **Incantation**, the first interactive video world model with **per-frame, per-entity language conditioning** under a single shared viewpoint, achieving real-time multi-entity control for $>$ **2** hours and reproducing its behavior on a second visually unrelated world under the same training recipe with vocabulary substitution as the only domain-specific change.
3. We construct a 128-hour gaming dataset spanning two heterogeneous worlds (*Elden Ring* and *The King of Fighters*), the first dataset with **accurate per-frame, per-entity action labels** at 0.25 s temporal granularity, directly extracted from game memory at zero temporal offset.

2 Related Work

Interactive Video World Models. Most interactive video world models still simulate only a single controllable entity. Following the world-model paradigm of [17, 18], recent diffusion-based engines such as GameNGen [40], DIAMOND [2] and Oasis [11], together with streaming systems including the Genie series [8, 28, 15], Matrix-Game [48, 20], MineWorld [16], WorldPlay [36], Infinite-World [43] and Hunyuan-GameCraft-2 [37], all bind every action stream to one entity; Vid2World [22] and AVID [30] further repurpose pretrained video diffusion models into action-conditioned world models under the same single-agent setup. Multi-entity attempts remain limited: Solaris [31] synchronizes multi-player Minecraft videos but emits per-player first-person streams rather than one holistic viewpoint, and COMBAT [1] renders a reactive Tekken 3 opponent inside a shared view without any directable interface for its strategy; ShareVerse [50] couples four agent-centric views on CARLA, MultiGen [29] enables editable multi-player rollouts via external memory, and LiveWorld [12] targets out-of-sight persistence, yet none delivers per-entity semantic commands.

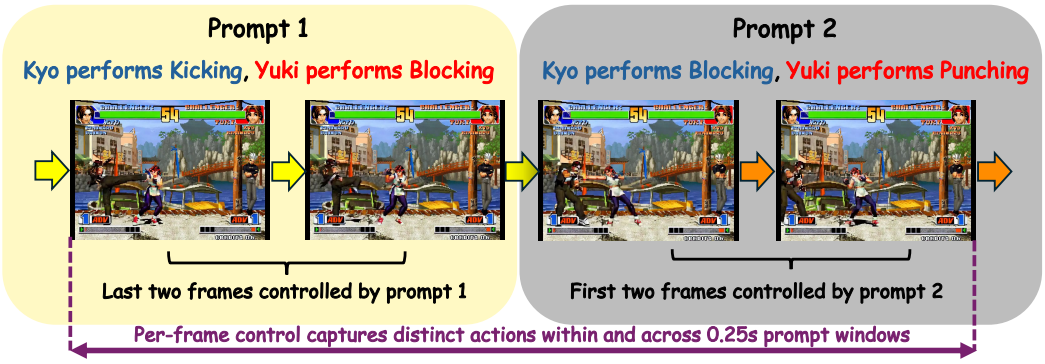

Figure 3: **Demonstrations of fine-grained multi-entity action control of Incantation in *KOF*. Incantation** precisely responds to rapid action inputs and successfully captures actions as brief as 0.25 s (e.g., *Punching*), demonstrating its fine-grained and responsive control capability.

Consequently, no existing system supports independent and simultaneous control of multiple entities within one holistic scene.

Action Interfaces of World Models. Existing world models inherit one of three action interfaces, each intrinsically limited in generality and scalability across entities and worlds. The first family encodes actions as *engine-internal animation labels*, that is, discrete identifiers exemplified by DIAMOND [2] on Atari and Counter-Strike, where every index is bound at design time to a specific in-game animation, leaving any out-of-vocabulary behavior inherently inexpressible. The second family conditions generation on *human-device inputs*, such as keyboard and mouse. Representative systems include GameNGen [40], the Genie series [8, 28, 15], Oasis [11], the Matrix-Game series [48, 20], The Matrix [13], MineWorld [16], WorldPlay [36], and GameFactory [47], all of which condition on per-frame keyboard or mouse signals tied to a single player, so the schema cannot specify *which* entity should act when multiple entities co-exist within the scene. The third family relies on *scene-level captions*, where GameGen-X [9] feeds InstructNet with whole-clip multi-modal instructions, Hunyuan-GameCraft-2 [37] follows free-form prompts such as "open the door", and LingBot-World [38] further steers global and local world events through textual prompts, each operating at the granularity of the entire scene rather than any individual subject and thus conflating distinct entities' behaviors under one global descriptor. Across the three families, no prior interface simultaneously delivers open-vocabulary semantics and per-entity addressability for independent simultaneous control of multiple co-existing entities, exposing the core gap that our work targets.

3 Incantation: Natural Language as the Action Interface

Realizing the language-as-action-interface end-to-end requires addressing two architectural challenges inherent to any language-conditioned, multi-entity interactive world model: (1) *Per-frame language conditioning* and (2) *Real-time long-horizon streaming inference*. We contribute one principled solution for each, structuring our pipeline into two stages. **Stage 1** (Section 3.1) addresses per-frame language conditioning via a per-entity prompt formulation on a bidirectional backbone with decoupled text cross-attention. **Stage 2** (Section 3.2) achieves real-time long-horizon streaming generation through a two-stage distillation (ODE initialization followed by Self-Forcing) combined with RoPE-decoupled KV-cache sliding. Throughout this work, the action interface targets the *discrete-semantic action* regime, where each per-frame action admits a textual description; continuous control signals (e.g., camera $\mathrm{SE}(3)$ trajectories) are out of scope and discussed in Appendix A.2.

3.1 Stage 1: Language-Conditioned Architecture

We adopt natural language as the action interface, which inherently decouples the conditioning signal from any specific engine or entity and thereby enables **generalization** across both entity types and world domains. Realizing this interface on top of a pretrained bidirectional video backbone [39] requires three coupled design choices: (1) how multi-entity prompts are formulated, (2) how attention

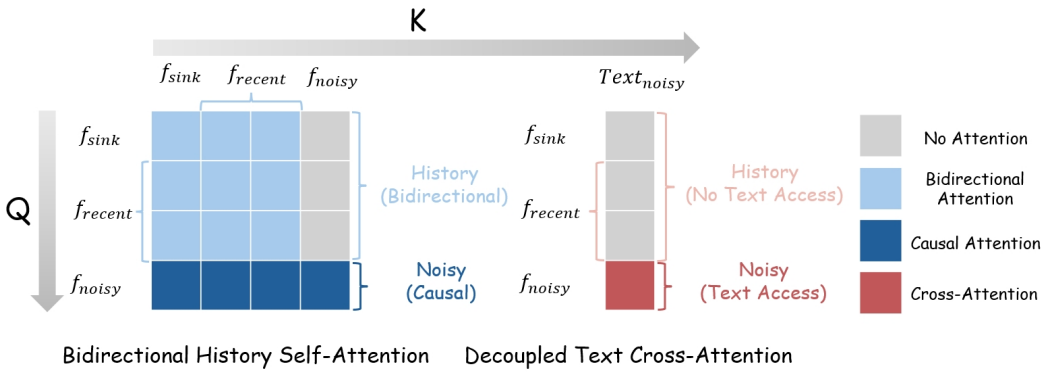

Figure 4: **Attention design.** Bidirectional self-attention is retained over history frames to preserve the spatio-temporal priors of the pretrained base model. Action cross-attention is restricted exclusively to the noisy target frame, preventing temporal cross-contamination. Together, these two constraints improve per-frame controllability without degrading generation quality.

is structured to turn high-level prompts into frame-accurate actions, and (3) how positional indices are assigned so that both training and bounded streaming inference stay in distribution.

Prompt Formulation. We represent multi-entity actions as a structured natural-language prompt with parallel, syntactically isolated slots (one per entity) at a 0.25 s granularity. As a concrete example, for two-entity control:

```
Player performs [ACTION_P]. Boss performs [ACTION_B].
```

This template supports both **simultaneous control** and **entity decoupling**: the temporal alignment of the two slots encourages the model to reason jointly about inter-entity dynamics within each frame, while the syntactic separation preserves independent control pathways for each entity. The template also extends naturally to settings with more or fewer entities by simply appending or omitting slots, requiring **no architectural modification** and demonstrating the inherent **scalability** of the natural-language interface.

Context Assembly. In the autoregressive diffusion-based video generation framework, each target frame is denoised by attending to a context window of conditioning frames passed as clean latents. We organize this window using a *Sink* + *Recent* + *Noisy* context structure; for each training step targeting frame t:

- **Sink frame** (K_s=1): the first frame of the episode, anchoring global context (arena geometry, character appearance) following the attention-sink mechanism of Xiao et al. [44].
- **Recent frames** (K_r=7): the 7 most recent clean latent tokens preceding t. Each latent token corresponds to 0.25 s of gameplay after the base model's VAE temporal compression, so the recent context spans 1.75 s of game time. We ablate K_r in Appendix A.9.
- **Noisy target** (K_n=1): the partially-denoised latent of frame t.

Per-Frame Language-Conditioned Attention. The conventional approach, with causal self-attention over all visual tokens plus full text cross-attention, introduces two failure modes under per-frame language conditioning: **(1) Destruction of pretrained priors.** The Wan 2.2 base model was pretrained with full bidirectional attention; its weights encode symmetric co-occurrence statistics. Imposing a global causal mask discards these priors, requiring costly re-adaptation. **(2) Temporal cross-contamination.** Each action prompt a_t describes exclusively what occurs at time t. Allowing a_t to cross-attend to history frames causes it to retroactively corrupt committed past representations, producing spurious action echoes in adjacent frames. We address both issues with a dedicated attention mechanism for per-frame language conditioning (Figure 4): **(1) Bidirectional history attention.** We apply *full bidirectional self-attention* over the $(K_s + K_r)$ history tokens, preserving the base model's pretrained co-occurrence statistics. A causal boundary separates history from the

noisy target, enforcing correct temporal ordering at generation time. **(2) Decoupled text cross-attention.** The per-frame action prompt a_t cross-attends *exclusively* with the noisy target token; history frames are masked out entirely. This prevents temporal cross-contamination: the current annotation cannot influence committed past representations. Ablation study appears in Appendix A.8.

Bounded RoPE Position Assignment. The naive sequential position assignment lets token indices grow unboundedly during streaming inference, placing them outside the range seen during training; this is a **RoPE out-of-distribution (OOD)** problem that fundamentally breaks long-horizon generation. We instead introduce two independent bounds: a sliding window size K_r (how many recent frames the KV cache holds) and a position cap $C \geq K_r$ (the largest local RoPE index any token can receive). The sink frame is permanently anchored at position 0, the noisy target at $\min(p_t, C)$ where p_t is its absolute frame index, and the K_r recent frames occupy the consecutive positions immediately preceding the target. K_r caps per-step compute and memory; C caps the positional range exposed to the model, and is set so every position used at inference also occurs during training. Together the two prevent RoPE OOD and enable the KV-cache sliding mechanism at inference (Section 3.2).

Training Setup. We fine-tune Wan 2.2 TI2V-5B [39] end-to-end on 16 H100 GPUs using Fully Sharded Data Parallel (FSDP) and mixed-precision training. We employ a two-resolution curriculum: 1,000 warmup steps at 256×448 (learning rate 2×10^{-5}), followed by 50,000 steps at 480×832 (learning rate 1×10^{-5}), with a global batch size of 64. Training data are described in Section 4.1.

3.2 Stage 2: Real-Time Streaming Inference

Real-time streaming inference is a prerequisite for any world model that aspires to support genuine interaction. The Stage 1 bidirectional teacher, however, requires 50 denoising steps per frame and attends over a full visual context, neither of which is compatible with real-time play. Stage 2 addresses two coupled bottlenecks for this challenge: (1) reducing per-frame compute via distillation, and (2) bounding per-frame memory via KV-cache sliding while preserving positional coherence.

ODE Initialization Before Distillation. The teacher was pretrained with bidirectional history attention, which grants rich spatio-temporal priors but is fundamentally incompatible with the strictly causal attention required by streaming inference. Before distillation, we must reconcile this mismatch. We initialize a causal student from the teacher's weights and align their predicted velocity fields via a flow-matching consistency objective [27]:

$$\mathcal{L}_{\text{ODE}} = \mathbb{E}_{\tau, v_0, \epsilon}\big[\| f_\theta(v_\tau;\, \text{causal}) - f_{\text{teacher}}(v_\tau;\, \text{bidir})\|_2^2\big]. \tag{1}$$

In practice, this objective closes the attention-mask gap within 1,000 steps at 480×832 resolution (16 H100 GPUs, learning rate $= 5\times10^{-6}$, batch size 128).

Self-Forcing Distillation. Building on the ODE-initialized student, we apply Self-Forcing [23] distillation to reduce inference to just **2 steps**. During training, the student conditions on *its own previously generated frames* rather than ground-truth frames, directly suppressing the compounding errors that would otherwise accumulate over autoregressive rollout.

RoPE-Decoupled KV-Cache Sliding Window. Under the bounded RoPE scheme in Section 3.1 for OOD prevention, a bounded KV-cache sliding window is required to enable real-time streaming inference. However, the bounded relative positional indices are time-dependent: after each eviction, surviving keys must be reassigned updated local relative positions. If RoPE-rotated keys are cached, their embeddings remain anchored to stale indices and become inconsistent with the current query, causing temporal flickering in the generated video. We therefore cache raw keys *before* RoPE rotation and apply RoPE on-the-fly with up-to-date local relative positions. Let p_i^{abs} and p_t^{abs} denote the absolute positions of cached frame i and the current query t, respectively, with C the local position cap defined in Section 3.1. Our local relative position assignment and RoPE-decoupled attention are:

$$p_i^{\text{local}} = \text{clamp}\big(p_i^{\text{abs}} - \delta,\ 0,\ C\big), \quad \delta = \max\big(0,\ p_t^{\text{abs}} - C\big). \tag{2}$$

$$\text{Attn}(q_t, k_i) = \text{Softmax}\{\big(q_t \cdot R(p_t^{\text{local}})\big)\big(k_i^{\text{raw}} \cdot R(p_i^{\text{local}})\big)^\top / \sqrt{d}\}. \tag{3}$$

When the buffer is full, the oldest non-sink frame is evicted while the sink frame is permanently retained at $p_{\text{sink}}^{\text{local}}$=0. The clamp cap C keeps every local position within the range exercised during

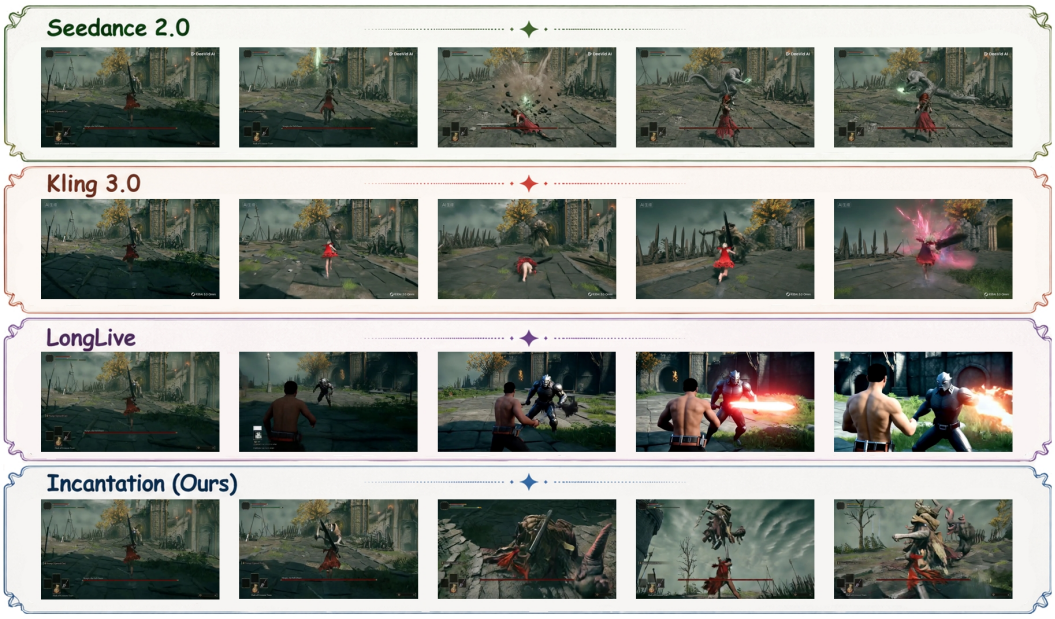

Figure 5: **Qualitative comparison of Incantation against leading video generation models on *Elden Ring*. Seedance 2.0** [33] and **Kling 3.0** [24] achieve high visual fidelity yet fail on fine-grained player–boss interactions; **LongLive** [45] partially captures multi-entity dynamics but loses action fidelity and visual coherence. Only **Incantation** delivers precise per-frame multi-entity action control with genuine interactive modeling (prompts in Appendix A.5). **Existing world models are excluded as baselines, as none supports multi-entity modeling within a single holistic scene.**

training, ensuring long-horizon generation remains fully OOD-free. Together, our design guarantees: (a) $\mathcal{O}(K_s + K_r)$ bounded memory; (b) all positions in-distribution; (c) artifact-free evictions.

4 Experiments

Our experiments are organized around two research questions. *(i) With all else held equal, does the language interface offer capabilities unreachable by an Action-Index baseline?* Section 4.1 introduces our testbed, baselines, and evaluation protocol; Section 4.2 then answers along three axes (in-distribution parity, cross-entity transfer, and out-of-vocabulary coverage), each designed to rule out a distinct confounder. *(ii) Does the same architecture sustain real-time inference and reproduce these gains in another visually unrelated world?* Section 4.3 addresses this by jointly reporting system-level metrics across *Elden Ring* and *The King of Fighters*. **In addition, we conduct extensive ablation studies, with full results deferred to Appendix A.8–A.9 due to page constraints.**

4.1 Experimental Setup

Testbed and Dataset. Our testbed spans two heterogeneous worlds: *Elden Ring* (3D action RPG, photorealistic) and *The King of Fighters* (KOF; 2D pixel-art). For *Elden Ring*, we collect 30 h of Margit and 15 h of Crucible Knight boss-fight footage, with per-frame triplets $(v_t, a_t^{\text{player}}, a_t^{\text{boss}})$ read directly from engine memory at zero temporal offset and player/boss vocabularies of 13 and 47 actions. For *KOF*, we gather $\sim$5,000 60-second fighter-pair clips ($\approx$83 h) (detailed in Appendix A.6).

Baselines. We compare two conditioning variants that differ only in their conditioning pathways, with all other factors held *identical*. The **Natural Language (NL) variant (ours)** encodes per-frame structured prompts via the model's pretrained text encoder into decoupled cross-attention layers, while the **Action-Index variant** instead represents each entity's action as a one-hot over the joint vocabulary, projected through a learnable linear layer. This capacity asymmetry is inherent, as equalizing it would artificially impose NL-level expressiveness into the Action-Index variant, rendering them fundamentally equivalent. Crucially, the joint vocabulary spans both entities, making

Table 1: **Quantitative results of NL vs. Action-Index ACA across Axis 1 and 2.** We report mean ACA over 20 trials per action (Axis 1) and per action pair (Axis 2). NL outperforms Action-Index under both in-distribution and cross-entity settings, with a 6 pp advantage on seen actions and a 46 pp advantage on unseen cross-entity transfers. Full per-action and per-pair breakdowns are provided in Appendix A.11 (Table 7) and Appendix A.12 (Table 8), respectively.

Evaluation Axis	**NL ACA (%)**	Action-Index ACA (%)
Axis 1: In-Distribution Parity	**95**	89
Axis 2: Cross-Entity Semantic Transfer	**89**	43

cross-entity action indices technically injectable into either entity's context — eliminating input-layer incompatibility as a confounding explanation for any cross-entity failure. Thus, Action-Index should be viewed as a steel-man abstraction of common discrete control interfaces, including keyboard/controller inputs, animation IDs, and one-hot action tokens: it is stronger than raw low-level controls because it receives semantic action labels, and stronger than typical entity-local ID spaces because we use a shared joint vocabulary in which every transferred action remains addressable by a valid index. When it fails under cross-entity transfer, the failure therefore reflects the absence of compositional semantics in index-bound interfaces rather than lack of access to the target action. Both variants build on Wan 2.2 TI2V-5B [39], with causal-masked self-attention for real-time streaming inference. **As the first single-viewpoint multi-entity world model with per-frame, per-entity language actions, Incantation has no directly comparable baseline.**

Metrics and Protocols. Our primary metric is **ACA** (Action Control Accuracy), defined as the fraction of generated clips judged consistent with their prompt by blinded annotators. Owing to the absence of an established *automated* evaluation protocol for this *nascent* task, we adopt two rigorous blinded subjective evaluation protocols, which better align with human perception, to assess action control for compositional steering (Section 4.2) and trajectory fidelity (Section 4.3), respectively. Since these two protocols evaluate different aspects, their absolute ACA values are therefore not directly comparable. Full evaluation details are provided in Appendix A.6.

4.2 Natural Language vs. Action-Index: Evidence Across Three Axes

We evaluate whether natural language (NL) constitutes a genuinely superior action interface over Action-Index through three controlled axes. **Axis 1** establishes a fair baseline by confirming that NL and Action-Index perform comparably on actions seen during training, ruling out model capacity or optimization as explanations for any subsequent gap. **Axis 2** tests whether NL generalizes action semantics across different entities—ruling out memorization as the source of any observed advantage. **Axis 3** exposes a structural limitation of Action-Index by construction: NL can express any action through free composition, whereas Action-Index cannot receive prompts outside its fixed vocabulary.

Axis 1: In-Distribution Parity. We evaluate NL and Action-Index on the five most frequent actions in the training set, which collectively dominate each entity's training data volume and ensure strong supervision for both interfaces. For each action, we report the ACA over 20 trials under varied random setups. As presented in Table 1, NL leads Action-Index by 6 pp in aggregate on seen actions, thereby ruling out long-tail artifacts as a confounding explanation.

Axis 2: Cross-Entity Semantic Transfer. To assess whether NL contributes semantic compositionality beyond the Action-Index interface, we examine whether the model can correctly interpret prompts for entity-action pairs that were *never* encountered during training. We test this on a hybrid model jointly trained on Margit and the Crucible Knight (disjoint action sets), evaluating five cross-entity action pairs. Each cross-entity prompt differs from its in-distribution counterpart by a single entity-identity word (NL) or a one-hot index swap (Action-Index), ensuring that any performance drop reflects failed semantic generalization rather than exposure to unfamiliar vocabulary. We conduct 20 trials per action pair and report mean ACA across all pairs. As shown in Table 1, NL outperforms Action-Index by 46 pp in mean ACA (89% vs. 43%), demonstrating that it is linguistic compositionality that enables robust cross-entity semantic transfer in a way discrete indexing fundamentally cannot.

Table 2: **Quantitative results across two visually unrelated worlds (Elden Ring, KOF).** With the same architecture and training recipe across worlds, the 2-step student achieves $\mathbf{74}/\mathbf{67}\times$ speedup over its teacher (Elden Ring/KOF), preserves ACA within 3 pp, and improves FVD. Seedance **2.0** and LongLive are evaluated only by trajectory-conditioned ACA under the same 0.25 s per-entity labels; FVD/latency are omitted as non-comparable. Ablations and timing details appear in Appendices A.8, A.9, and A.17.

World	Model	Steps	FVD ↓	ACA (%) ↑	Latency ↓
Elden Ring	Teacher (bidir.)	50	206.2	93.2	12058.7 ms/frame
	Student (causal)	**2**	**138.6**	**90.4**	**163.4** ms/frame
	Seedance **2.0** [33]	—	—	46.7	—
	LongLive [45]	4	—	20.3	206.6 ms/frame
KOF	Teacher (bidir.)	50	170.1	94.9	10986.0 ms/frame
	Student (causal)	**2**	**162.9**	**94.0**	**165.2** ms/frame

As an auxiliary automatic check, a VLM pairwise judge also favours NL on the same cross-entity pairs (62% vs. 37% win rate), corroborating the human ACA trend (Appendix A.12, Table 9).

Axis 3: Out-of-Vocabulary Coverage. The third axis concerns prompts that extend, modify, or rephrase the training vocabulary while remaining compositionally meaningful (e.g., `Double light blade throw` → `Dual light blade throw`). Here the NL-vs-Action-Index gap is *structural* rather than quantitative: *the Action-Index interface has no input slot for any such prompt*, so supporting any single one would require modifying the input-layer vocabulary fundamentally. We construct four such probes, each with a single-word edit of one of the entity's top-3 frequent training prompts, giving Action-Index the strongest possible base embedding for a steel-man comparison (full set in Appendix A.13). Because no edit matches any predefined action index, the Action-Index interface scores exactly $\mathbf{0}\%$ regardless of model capacity, whereas NL achieves $\mathbf{90}\%$ aggregate ACA across the four probes in 40 trials in total. Stronger Action-Index baselines (e.g., factorized entity×action tables) likewise reduce to either NL or our joint-vocabulary implementation (Appendix A.10), leaving the structural weakness of the Action-Index interface intact. **Therefore, OOV coverage is unique to NL by construction: no scaling of an Action-Index interface can close this gap.**

4.3 Real-Time System and Cross-World Replication

To validate the cross-world transfer of **Incantation**, we retrain on KOF under *identical* architecture and hyperparameters as in Elden Ring, only modifying the action-vocabulary slots. Table 2 compares the bidirectional teacher and its Self-Forcing causal 2-step student at 480×832 on both worlds. On Elden Ring, the student achieves a $\mathbf{74}\times$ speedup over the teacher at a comparable accuracy, while actually improving visual fidelity. On KOF, the same recipe under vocabulary substitution alone yields an analogous performance on a visually unrelated world. In addition, **Incantation** supports real-time streaming at 19.7 FPS end-to-end, enabled by TAEHV [7], a tiny VAE (detailed in Appendix A.17). Although the training context spans only 1.75 s, the student maintains stable generation quality at much longer horizons: across continuous 30- to 118-minute sessions, FVD stays in a tight band (mean 166.0, range $[162, 171]$) with no degradation over time (Appendix A.15).

5 Conclusion

We present **Incantation**, the first interactive video world model to adopt natural language as a *per-frame, per-entity* action interface, overcoming the expressiveness constraints of conventional interfaces. **Incantation** achieves accurate multi-entity control in both cross-entity and out-of-vocabulary scenarios, and sustains real-time streaming at 19.7 FPS over 2-hour continuous horizons. **Limitations: (1) Annotation Channel.** We read training labels from game memory because games offer frame-accurate per-entity supervision at zero cost; this is a testbed choice, not an interface property. The NL interface consumes per-entity captions from any source—VLM auto-labelers, tele-operation logs, or robot proprioception—without architectural change. **(2) Continuous Controls.** Our interface targets semantic actions; future hybrid controllers could combine language with continuous channels for precise camera $\mathrm{SE}(3)$ or force/velocity control. See Appendix A.2 for details.

References

[1] Anmol Agarwal, Pranay Meshram, Sumer Singh, Saurav Suman, Andrew Lapp, Shahbuland Matiana, Louis Castricato, and Spencer Frazier. COMBAT: Conditional world models for behavioral agent training. *arXiv preprint arXiv:2603.00825*, 2026.

[2] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in Atari. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[3] Ali Baheri. Logic-guided vector fields for constrained generative modeling. *arXiv preprint arXiv:2602.02009*, 2026.

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

[5] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[6] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[7] Ollin Boer Bohan. Taehv: Tiny autoencoder for hunyuan video. `https://github.com/madebyollin/taehv`, 2025.

[8] Jake Bruce, Michael 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 *International Conference on Machine Learning (ICML)*, 2024.

[9] 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 (ICLR)*, 2025.

[10] Jacob K. Christopher, Michael Cardei, Jinhao Liang, and Ferdinando Fioretto. Neuro-symbolic generative diffusion models for physically grounded, robust, and safe generation. In *Proceedings of the International Conference on Neuro-Symbolic Systems*, volume 288 of *Proceedings of Machine Learning Research*, pages 188–213. PMLR, 2025.

[11] Decart AI and Etched AI. Oasis: A universe in a transformer. `https://oasis-model.github.io/`, 2024.

[12] Zicheng Duan, Jiatong Xia, Zeyu Zhang, Wenbo Zhang, Gengze Zhou, Chenhui Gou, Yefei He, Feng Chen, Xinyu Zhang, and Lingqiao Liu. LiveWorld: Simulating out-of-sight dynamics in generative video world models. *arXiv preprint arXiv:2603.07145*, 2026.

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

[14] Artur d'Avila Garcez and Luis C. Lamb. Neural-symbolic learning and reasoning: A survey and interpretation. In *Neuro-Symbolic Artificial Intelligence: The State of the Art*, volume 342 of *Frontiers in Artificial Intelligence and Applications*, pages 1–51. IOS Press, 2022.

[15] Google DeepMind. Genie 3: A new frontier for world models. `https://deepmind.google/blog/genie-3-a-new-frontier-for-world-models/`, 2025. Google DeepMind Blog.

[16] Junliang Guo, Yang Ye, Tianyu He, Haoyu Wu, Yushu Jiang, Tim Pearce, and Jiang Bian. MineWorld: A real-time and open-source interactive world model on Minecraft. *arXiv preprint arXiv:2504.08388*, 2025.

[17] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[18] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025.

[19] Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. LM-Infinite: Zero-shot extreme length generalization for large language models. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 3991–4008, 2024.

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

[21] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[22] Siqiao Huang, Jialong Wu, Qixing Zhou, Shangchen Miao, and Mingsheng Long. Vid2World: Crafting video diffusion models to interactive world models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[23] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[24] Kuaishou Technology. Kling AI launches 3.0 model, ushering in an era where everyone can be a director. `https://ir.kuaishou.com/news-releases/news-release-details/kling-ai-launches-30-model-ushering-era-where-everyone-can-be`, February 2026. Accessed: 2026-05-01.

[25] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[26] Wang Lin, Liyu Jia, Wentao Hu, Kaihang Pan, Zhongqi Yue, Wei Zhao, Jingyuan Chen, Fei Wu, and Hanwang Zhang. Reasoning physical video generation with diffusion timestep tokens via reinforcement learning. *arXiv preprint arXiv:2504.15932*, 2025.

[27] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

[28] Jack Parker-Holder, Philip Ball, Jake Bruce, Vibhavari Dasagi, Kristian Holsheimer, Christos Kaplanis, Alexandre Moufarek, Guy Scully, Jeremy Shar, Jimmy Shi, et al. Genie 2: A large-scale foundation world model. `https://deepmind.google/blog/genie-2-a-large-scale-foundation-world-model/`, 2024. Google DeepMind Blog.

[29] Ryan Po, David Junhao Zhang, Amir Hertz, Gordon Wetzstein, Neal Wadhwa, and Nataniel Ruiz. MultiGen: Level-design for editable multiplayer worlds in diffusion game engines. *arXiv preprint arXiv:2603.06679*, 2026.

[30] Marc Rigter, Tarun Gupta, Agrin Hilmkil, and Chao Ma. AVID: Adapting video diffusion models to world models. In *International Conference on Learning Representations (ICLR)*, 2025.

[31] Georgy Savva, Oscar Michel, Daohan Lu, Suppakit Waiwitlikhit, Timothy Meehan, Dhairya Mishra, Srivats Poddar, Jack Lu, and Saining Xie. Solaris: Building a multiplayer video world model in Minecraft. *arXiv preprint arXiv:2602.22208*, 2026.

[32] Davide Scassola, Sebastiano Saccani, Ginevra Carbone, and Luca Bortolussi. Zero-shot conditioning of score-based diffusion models by neuro-symbolic constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 20302–20309, 2025.

[33] Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, Mojie Chi, Xuyan Chi, Jian Cong, Qinpeng Cui, Fei Ding, Qide Dong, Yujiao Du, Haojie Duanmu, Junliang Fan, Jiarui Fang, Jing Fang, Zetao Fang, Chengjian Feng, Yu Gao, Diandian Gu, Dong Guo, Hanzhong Guo, Qiushan Guo, Boyang Hao, Hongxiang Hao, Haoxun He, Jiaao He, Qian He, Tuyen Hoang, Heng Hu, Ruoqing Hu, Yuxiang Hu, Jiancheng Huang, Weilin Huang, Zhaoyang Huang, Zhongyi Huang, Jishuo Jin, Ming Jing, Ashley Kim, Shanshan Lao, Yichong Leng, Bingchuan Li, Gen Li, Haifeng Li, Huixia Li, Jiashi Li, Ming Li, Xiaojie Li, Xingxing Li, Yameng Li, Yiying Li, Yu Li, Yueyan Li, Chao Liang, Han Liang, Jianzhong Liang, Ying Liang, Wang Liao, J. H. Lien, Shanchuan Lin, Xi Lin, Feng Ling, Yue Ling, Fangfang Liu, Jiawei Liu, Jihao Liu, Jingtuo Liu, Shu Liu, Sichao Liu, Wei Liu, Xue Liu, Zuxi Liu, Ruijie Lu, Lecheng Lyu, Jingting Ma, Tianxiang Ma, Xiaonan Nie, Jingzhe Ning, Junjie Pan, Xitong Pan, Ronggui Peng, Xueqiong Qu, Yuxi Ren, Yuchen Shen, Guang Shi, Lei Shi, Yinglong Song, Fan Sun, Li Sun, Renfei Sun, Wenjing Tang, Boyang Tao, Zirui Tao, Dongliang Wang, Feng Wang, Hulin Wang, Ke Wang, Qingyi Wang, Rui Wang, Shuai Wang, Shulei Wang, Weichen Wang, Xuanda Wang, Yanhui Wang, Yue Wang, Yuping Wang, Yuxuan Wang, Zijie Wang, Ziyu Wang, Guoqiang Wei, Meng Wei, Di Wu, Guohong Wu, Hanjie Wu, Huachao Wu, Jian Wu, Jie Wu, Ruolan Wu, Shaojin Wu, Xiaohu Wu, Xinglong Wu, Yonghui Wu, Ruiqi Xia, Xin Xia, Xuefeng Xiao, Shuang Xu, Bangbang Yang, Jiaqi Yang, Runkai Yang, Tao Yang, Yihang Yang, Zhixian Yang, Ziyan Yang, Fulong Ye, Bingqian Yi, Xing Yin, Yongbin You, Linxiao Yuan, Weihong Zeng, Xuejiao Zeng, Yan Zeng, Siyu Zhai, Zhonghua Zhai, Bowen Zhang, Chenlin Zhang, Heng Zhang, Jun Zhang, Manlin Zhang, Peiyuan Zhang, Shuo Zhang, Xiaohe Zhang, Xiaoying Zhang, Xinyan Zhang, Xinyi Zhang, Yichi Zhang, Zixiang Zhang, Haiyu Zhao, Huating Zhao, Liming Zhao, Yian Zhao, Guangcong Zheng, Jianbin Zheng, Xiaozheng Zheng, Zerong Zheng, Kuan Zhu, and Feilong Zuo. Seedance 2.0: Advancing video generation for world complexity, 2026. URL `https://arxiv.org/abs/2604.14148`.

[34] Hikaru Shindo, Quentin Delfosse, Devendra Singh Dhami, and Kristian Kersting. BlendRL: A framework for merging symbolic and neural policy learning. In *International Conference on Learning Representations (ICLR)*, 2025.

[35] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529:484–489, 2016.

[36] 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.

[37] 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.

[38] 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.

[39] Wan Team, 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.

[40] Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *International Conference on Learning Representations (ICLR)*, 2025.

[41] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575:350–354, 2019.

[42] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.

[43] Ruiqi Wu, Xuanhua He, Meng Cheng, Tianyu Yang, Yong Zhang, Zhuoliang Kang, Xunliang Cai, Xiaoming Wei, Chunle Guo, Chongyi Li, and Ming-Ming Cheng. Infinite-World: Scaling interactive world models to 1000-frame horizons via pose-free hierarchical memory. *arXiv preprint arXiv:2602.02393*, 2026.

[44] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*, 2024.

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

[46] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6613–6623, 2024.

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

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

[49] Hongyu Zhao, Siyu Zhou, Haolin Yang, Zengyi Qin, and Tianyi Zhou. Neuro-symbolic synergy for interactive world modeling. *arXiv preprint arXiv:2602.10480*, 2026.

[50] Jiayi Zhu, Jianing Zhang, Yiying Yang, Wei Cheng, and Xiaoyun Yuan. ShareVerse: Multi-agent consistent video generation for shared world modeling. *arXiv preprint arXiv:2603.02697*, 2026.

Table 3: Systematic comparison of interactive video world models. ✓ = supported, ✗ = not supported, $\sim$ = partial. *Multi-entity* requires independent and simultaneous control of two distinct entities. *Semantic NL* requires per-frame natural language action conditioning.

Model	Multi-entity	Semantic NL	$\geq$16FPS	$>$5min
GameNGen [40]	✗	✗	✓	✓
Genie 2 [28]	$\sim$	✗	✓	✗
Genie 3 [15]	$\sim$	✗	✓	$\sim$
The Matrix [13]	✗	✗	✓	✓
Matrix-Game 2.0 [20]	✗	✗	✓	$\sim$
LingBot-World [38]	✗	✗	✓	✓
Solaris [31]	✓	✗	$\sim$	$\sim$
Incantation (Ours)	✓	✓	✓	✓

A Appendix

A.1 Comparison with Related Work

Table 3 presents a systematic comparison of **Incantation** against representative interactive video world models along four dimensions: multi-entity control, semantic natural language interface, real-time frame rate ($\geq$16FPS), and long-horizon generation ($>$5min).

Explanation of absence of world model baselines. As established in Table 3 and Section 2, while a small number of existing world models accommodate concurrent multi-entity modeling to some extent, none achieves independent and simultaneous control of multiple entities within a single holistic scene. Because **Incantation** is, to our knowledge, the first system to address this setting, no directly comparable baseline exists for quantitative evaluation.

Additional Related Work for Efficient Streaming Video Generation. Several lines of work advance streaming video generation from complementary angles. On the diffusion side, Flow Matching [27] and Distribution Matching Distillation (DMD) [46] substantially reduce the number of inference steps. On the autoregressive side, Self-Forcing [23] eliminates exposure bias caused by the training-inference discrepancy. For long-horizon memory management, StreamingLLM [44] and LM-Infinite [19] bound memory usage via attention-sink tokens and sliding-window KV caches. Our work integrates these advances together to achieve real-time long-horizon streaming generation.

A.2 Limitations and Future Work

Incantation has three limitations that point toward concrete future directions. First, our training labels are obtained by direct in-engine memory instrumentation, since games are the only domain that simultaneously offers frame-accurate, per-entity, zero-cost supervision at interactive frame rates; this is a deliberate testbed choice and is orthogonal to the language interface itself, which only consumes per-entity action captions and is agnostic to how those captions are produced. Closing the annotation channel for non-instrumented domains, namely real-world video and closed-source engines, reduces to producing per-entity captions from an alternative source such as vision–language auto-labelers, tele-operation logs, or robot proprioception, all of which integrate without architectural change. We therefore view this as a data-side problem rather than a structural restriction of **Incantation**. Second, open-vocabulary expressiveness is ultimately bounded by the pretrained text encoder: composing concepts already within its training distribution generalizes naturally to unseen combinations, whereas truly novel tokens require explicit encoder adaptation. Our current interface also targets semantic action control rather than numerically precise continuous controls, such as camera $\mathrm{SE}(3)$ trajectories or force/velocity commands in robotic manipulation. This does not require replacing the language interface: a natural extension is to keep language as the high-level per-entity semantic channel and add a parallel continuous-control module, whose embeddings can be fused with the same per-frame conditioning layers used by **Incantation**. Third, episode-level state beyond the generator's $\sim 1.75\,\mathrm{s}$ context window is maintained by a hand-specified persistent-state module (Appendix A.16); replacing it with a learned, world-agnostic alternative remains open. In strictly single-agent scenarios, the language interface also degenerates into a relabelling of discrete inputs and offers no representational advantage over conventional action identifiers. Extending **Incantation** to non-game interactive-video domains, where per-entity annotations must be inferred rather than read from engine memory, constitutes the most immediate direction for future work.

A.3 Additional Qualitative Rollouts

We include additional qualitative rollouts to make the generated interactive worlds easier to inspect visually. Figures 6 and 7 complement the quantitative evaluation by showing representative long-horizon behavior in the two domains used in the paper.

A.4 Full Action Vocabulary

We summarize the per-entity action vocabularies used for prompt conditioning in Table 4. The player vocabulary $\mathcal{A}_{\text{player}}$ consists of 13 actions covering locomotion, defensive rolls, weapon attacks, and terminal states. Margit's native repertoire $\mathcal{A}_{\text{boss}}^{\text{Margit}}$ contains 30 actions, while the Crucible Knight contributes 17 additional non-overlapping moves. We obtain the joint boss vocabulary $\mathcal{A}_{\text{boss}}^{\text{joint}}$ with $|\mathcal{A}_{\text{boss}}^{\text{joint}}| = 47$ by deduplication across the two bosses, and we adopt this joint vocabulary throughout the experiments so that any cross-entity action index is technically injectable into either boss's context.

We obtain these vocabularies by manually aggregating the raw animation-state IDs read from engine memory. The raw stream is not an action vocabulary in the human-meaningful sense: a single human action such as a heavy slash unrolls into a sequence of typically six or more consecutive raw IDs corresponding to its sub-phases (e.g., `windup` $\rightarrow$ `strike` $\rightarrow$ `recovery` $\rightarrow$ `idle`), and a representative recording session already exposes 111 distinct player IDs and 53 distinct boss IDs even before the full dataset is exhausted. Domain-expert aggregation from the raw IDs to the 13/47-action vocabulary is therefore a prerequisite shared by any Action-Index baseline rather than an advantage of NL conditioning, and our vocabularies define the action-level ground truth on which both NL and Action-Index baselines are evaluated. We release the full raw-ID-to-action mapping with the dataset (see Appendix A.10 for the implications on stronger Action-Index baselines).

A.5 Baseline Prompt Settings

We specify the text prompts that we feed to the three video generation baselines compared in Figure 5, namely **Seedance 2.0** [33], **Kling 3.0** [24], and **LongLive** [45], as follows.

Figure 6: **Elden Ring rollout from a continuous Margit session.** We show 40 frames sampled from the generated stream starting at the 1-minute mark. The sequence illustrates long-horizon visual stability and fine-grained player–boss interaction in a complex 3D adversarial scene.

Environment: Stormveil Castle bridge, overcast sky, cinematic combat.
Agents: Player (Greatsword user) vs. Boss (Margit, the Fell Omen). **Player:**

- 0.00 s – 1.50 s: Move forward
- 1.50 s – 2.50 s: Roll forward
- 2.50 s – 3.50 s: Greatsword thrust
- 3.50 s – 4.50 s: Roll forward
- 4.50 s – 5.50 s: Greatsword thrust
- 5.50 s – 6.25 s: Roll backward
- 6.25 s – 7.25 s: Greatsword thrust
- 7.25 s – 8.00 s: Roll left
- 8.00 s – 8.75 s: Roll right
- 8.75 s – 10.00 s: Move forward

Boss:

- 0.00 s – 2.50 s: Jump and mid-air slam
- 2.50 s – 4.00 s: Tail swipe

Figure 7: **KOF rollout under the same architecture and training recipe.** We show 30 frames from a KOF rollout. The sequence illustrates that the same per-entity language-conditioning recipe also supports visually distinct 2D fighting gameplay.

- 4.00 s – 5.00 s: Jump back to disengage
- 5.00 s – 7.50 s: Jump and mid-air slam
- 7.50 s – 9.00 s: Tail swipe
- 9.00 s – 10.00 s: Horizontal slash

Since the three commercial baselines all incorporate built-in prompt-enhancement modules, we adopt this explicit timestamp-structured format to ensure a fair and controlled comparison with **Incantation** under matched per-entity action schedules.

A.6 Experimental Setup Details

Why games are the testbed. We choose games as the testbed because the interface claim requires frame-accurate multi-entity action labels at interactive frame rates, and games are the only domain that offers all three properties simultaneously. Specifically, per-frame animation state is readable from engine memory at zero annotation cost, the action vocabularies are bounded yet non-trivial, and the evaluation criteria are unambiguous. In contrast, driving and embodied-manipulation datasets lack frame-level entity-wise annotations, and narrative-video datasets lack adversarial multi-entity dynamics.

Data pipeline. We assemble the training data from three distinct entity domains:

Table 4: Margit's native action vocabulary, with $|\mathcal{A}_{\text{player}}| = 13$ and $|\mathcal{A}_{\text{boss}}^{\text{Margit}}| = 30$. The Crucible Knight contributes additional non-overlapping actions, yielding the joint boss vocabulary $|\mathcal{A}_{\text{boss}}^{\text{joint}}| = 47$ that we use throughout the experiments.

Player Actions (13)	**Boss Actions (30)**
Standing, Move (4dir), Roll (4dir), Greatsword Sweep, Greatsword Thrust, Death, Execution	Moving (forward / left / right), Jump and mid-air slam, Horizontal slash, Heavy overhead slash, Staff slam, Uttering curse, Quick slam, Double light blade throw, Staff upswing, Tail swipe, Double light blade slash, X-shaped slash, Charged staff thrust, Forward charge, Jump back to disengage, + 13 additional combo/variant moves

- **Elden Ring – Margit:** We collect 30 hours of boss-fight gameplay and segment it into $\sim$10,000 high-quality 5-second clips at 16 FPS after filtering and quality-based pruning, with 10% held out by recording date for evaluation.
- **Elden Ring – Crucible Knight:** We collect 15 hours of comparable footage segmented and filtered identically ($\sim$5,000 clips), and we use it jointly with Margit for the cross-entity evaluation.
- **The King of Fighters (KOF):** We collect $\sim$5,000 60-second fighter-pair clips at 16 FPS ($\approx$83 hours in total), and we use this corpus to validate the cross-world transfer of the architecture.

For Elden Ring, we obtain all per-frame labels by reading the engine's `current_animation` field at runtime via direct memory instrumentation, which yields zero-offset action triplets $(v_t, a_t^{\text{player}}, a_t^{\text{boss}})$ with $a_t \in \mathcal{A}_{\text{player}} \times \mathcal{A}_{\text{boss}}^{\text{joint}}$. We have $|\mathcal{A}_{\text{player}}| = 13$ and $|\mathcal{A}_{\text{boss}}^{\text{joint}}| = 47$, where the joint boss vocabulary is the deduplicated union of Margit and the Crucible Knight (Margit's native subset contains 30 actions; see Appendix A.4). For KOF, we read per-frame labels from the emulator's animation-state register under the same zero-cost protocol.

Annotation protocol. For every reported ACA number, we pool all 20 trials per condition (5 starting frames $\times$ 4 seeds) across all conditions, randomly shuffle them, and have three annotators independently rate each clip on a three-point ordinal scale (**0**: action absent; **1**: partial execution; **2**: full execution). Before rating, we strip both the conditioning-variant label (NL vs. Action-Index) and the prompt-source identity (in-distribution vs. cross-entity vs. OOV), so that all clips are rated under fully blinded conditions, as shown in the annotation interface in Figure 8. We take the per-trial score as the median of the three ratings, and we report ACA(≥ 1) as the fraction of clips whose median is at least 1.

Prompt-injection rollout protocol (Axes 1–3). We adopt the following prompt-injection protocol for all interface-evaluation rollouts in Axes 1–3, where the goal is to isolate the model's compositional steering capability:

(i) **Starting frame.** We sample a starting frame uniformly at random from the held-out test split and decode it into the model's visual context window.

(ii) **Un-conditioned warm-up.** The model then generates approximately 2 seconds (32 frames) of un-conditioned video, during which we set both the player and boss action prompts to the neutral idle/standing token at every step. The model is therefore conditioned only on its own visual history. This warm-up serves two purposes. First, it places the model in a steady-state denoising regime before the prompted phase begins, which avoids the artifacts typical of cold-started rollouts. Second, it severs any residual visual cue from the test-set continuation that might otherwise inform the model that a particular action is about to occur; without this, the model could in principle reproduce the target action by extrapolating the test-set trajectory rather than by genuinely responding to the prompt.

(iii) **Prompt injection.** At $t = 2$ s, we replace the neutral prompt with the target action prompt and hold it constant for the remaining 3 seconds of the clip.

(iv) **Rating.** Annotators rate the resulting 5-second clip against the target action over the post-injection window.

We apply the same warm-up and injection schedule to both the NL and Action-Index conditioning variants under matched starting frames and seeds, so that the interface comparison is conducted under identical visual conditions. We do not search over the warm-up duration: the 2-second value is fixed before annotation begins.

Trajectory-conditioned protocol (system metrics). We adopt a separate trajectory-conditioned protocol for Table 2 and the system-level ablations in Appendices A.8 and A.9, where the goal is to measure how faithfully the model tracks a fully specified ground-truth action trajectory rather than its compositional steering ability:

- *Sample.* We use 100 held-out 10-second clips per model.
- *Conditioning.* Rollouts begin at the first frame of each clip, and at every frame we set both the player and boss prompts to the ground-truth caption derived from engine memory. We use no warm-up phase and no prompt switch.
- *Rating.* We rate each rollout as binary correct or incorrect against its source clip's action sequence, and ACA reports the fraction correct.

The two protocols therefore evaluate complementary capabilities, namely compositional steering versus trajectory fidelity, and their absolute ACA values should not be directly compared.

Training. We fine-tune the Stage 1 teacher for 51k iterations (1k warmup at 256×448 followed by 50k at 480×832) using AdamW with peak learning rate 1×10^{-5} and global batch size 64 on $16\times$H100 80 GB GPUs. We then perform ODE initialization for 1,000 steps at 480×832 (learning rate 5×10^{-6}, batch size 128, $16\times$H100), followed by Self-Forcing distillation for 15k iterations at learning rate 2×10^{-6}.

Video resolution. We generate all videos at 480×832 (480p) and 16 FPS. We compress context frames into latents through the base model's VAE at a $4\times$ spatial downsampling and a $4\times$ temporal downsampling.

Hit detection. We fine-tune Qwen3-VL-2B-Instruct [5] with LoRA on the vision–language connector and the cross-attention modules. We aggregate frame-level predictions into a per-window classification through majority voting.

End-to-end throughput. We measure the diffusion student's wall-clock latency at 160 ms per frame on a single H100 80 GB, where the per-frame loop is dominated by the diffusion pass with KV-cache sliding and RoPE decoupling, the VAE decode, and the surrounding I/O. We will release detailed per-component timings alongside the open-source code.

A.7 Annotator Reliability

We rate 400 generated clips in total (200 cross-entity and 200 in-distribution) under the protocol of Appendix A.6. Each clip is scored independently by three blinded annotators on the $\{0, 1, 2\}$ ordinal scale with conditioning-variant and prompt-source labels stripped, and we take the per-clip ACA score as the binary indicator that the median of the three ordinal ratings is at least 1. In this subsection we report inter-rater consistency restricted to the 5-pair cross-entity subset (the Axis 2 pairs in Table 1) and the 5-action in-distribution subset (the Axis 1 actions in Table 1).

Within-1 ordinal agreement. On the $\{0, 1, 2\}$ scale, raters disagree by more than one tier on fewer than 7% of clips across either split. Specifically, the within-1 agreement on the cross-entity subset is 96.5%, 96.0%, and 95.0% for the three rater pairs $A_1 \times A_2$, $A_1 \times A_3$, and $A_2 \times A_3$; on the in-distribution subset, the corresponding numbers are 93.3%, 95.8%, and 94.2%. Consequently, aggregating by the median of the three ratings before binarizing at the ≥ 1 threshold inherits a noise floor of at most one-tier disagreement on at most 7% of clips, which is materially smaller than every NL vs. Action-Index gap reported in the paper.

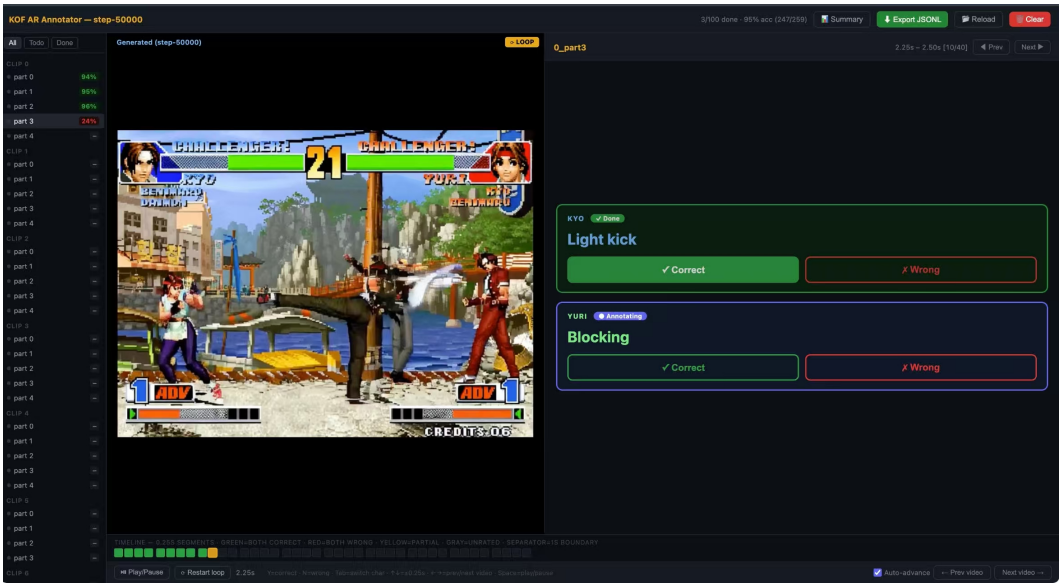

Figure 8: **Annotation interface for the human evaluation of Action Control Accuracy (ACA).** Each trial presents the annotators with a generated video clip alongside the per-entity target action label (here: KYO—*Light kick*; YURI—*Blocking*). We strip conditioning-variant identities (NL vs. Action-Index) and prompt-source labels before rating, which ensures a fully blinded evaluation. Each annotator rates each entity's action on the three-point ordinal scale (**0**: absent; **1**: partial; **2**: full); we then take the per-clip ACA(≥ 1) as the binary indicator that the median of the three ratings is at least 1.

Paired McNemar on cross-entity: $\mathbf{Z = 6.38}$, $\mathbf{p < 10^{-10}}$. For each (pair, starting frame, seed) triple, we obtain one NL clip and one Action-Index clip evaluated under identical visual conditioning and rated by the same three blinded annotators under the median-of-three scheme. We pool the matched triples across all five pairs and apply the McNemar test on the resulting binary scores. The test yields 49 triples on which NL succeeds and Action-Index fails, against 3 triples on which Action-Index succeeds and NL fails, giving $Z = 6.38$ and $p < 10^{-10}$ (one-sided). Therefore, among the clips on which the two interfaces produce different outcomes, the language interface succeeds on more than an order of magnitude as many clips as the Action-Index baseline, and the cross-entity gap cannot be attributed to cell-level fluctuation.

A.8 Stage 1: Conditioning Architecture Ablation

We ablate the two key Stage 1 design choices in a 2×2 factorial that crosses the type of history self-attention (bidirectional vs. causal) with the scope of text cross-attention (noisy frame only vs. all frames). We train all four variants from the same Wan 2.2 TI2V-5B checkpoint with identical hyperparameters (batch size 64, learning rate 2×10^{-5}, resolution 256×448), and we evaluate them on the held-out test split every 5k steps up to 35k steps under the trajectory-conditioned protocol of Appendix A.6.

Table 5: **Stage 1 architecture ablation (2×2).** Each cell reports FVD↓ at the best checkpoint, selected by the lowest FVD. †: training never stabilizes, with FVD exceeding 1,100 at all checkpoints.

	Text: noisy only	**Text: all frames**
Bidir. history (ours)	**201.9**	197.1
Causal history	245.1	$1157.9^{\dagger}$

Three findings emerge from this ablation. **(1) Bidirectional history attention consistently outperforms causal history attention** (201.9 vs. 245.1 FVD). This confirms that forcing causal masking on history tokens breaks the bidirectional inductive bias inherited from the Wan 2.2 pretrained weights.

(2) Decoupling text cross-attention to the noisy frame is essentially free (201.9 vs. 197.1 FVD, within noise), which demonstrates that our design preserves semantic clarity at zero quality cost. **(3) Causal history combined with full cross-attention is unstable** (FVD $> 1{,}100$ throughout training). In this configuration, injecting the current-frame action label a_t into committed causal history keys contaminates those representations with future action information, which supports the temporal cross-contamination analysis in Section 3.1. We further plot the per-step training curves in Figure 9, and they confirm that these findings hold throughout training rather than only at a single checkpoint.

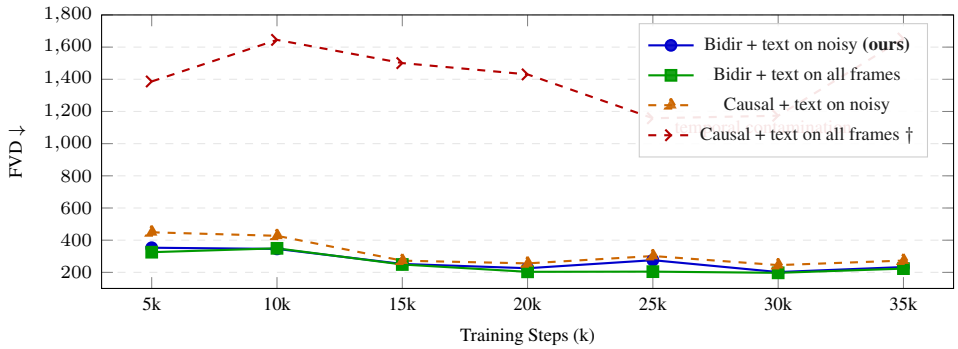

Figure 9: **Stage 1 ablation: FVD vs. training steps.** Companion to Table 5. The causal-history plus full-cross-attention configuration (†) collapses throughout training, whereas the other three configurations converge to FVD ~ 200.

A.9 Stage 2: KV-Cache and Bounded RoPE Ablation

We sweep the two Stage 2 design choices on the same distilled student. Group A omits the bounded sliding window entirely and retains the full history, which incurs unbounded VRAM growth. Group B fixes the sliding window at `kv_window`$= 7$ to match the training value $K_r = 7$, and we sweep the local-RoPE cap within this group. Group C fixes the cap at `cap`$= 16$ and we sweep the window size. We report the held-out FVD under the trajectory-conditioned protocol of Appendix A.6 at both 10 s and 30 s horizons.

Table 6: **Stage 2: KV-cache and RoPE ablation.** We report FVD↓ on the 10 s and 30 s held-out test sets. ‡: full history retained in VRAM, with OOM risk on long sequences. KV sliding is the dominant factor: without it, FVD more than doubles at 30 s. With sliding enabled, the RoPE cap has negligible effect on FVD because the sliding window already bounds the relative positions to the training range. All results are obtained using the native Wan 2.2 VAE.

Group	Configuration	FVD ↓ (10 s)	FVD ↓ (30 s)
A: no sliding‡	No cap	439.6	996.9
B: kv$= 7$	No cap	137.7	141.2
	cap$= 16$ (ours)	**138.6**	**139.2**
C: cap$= 16$	kv$= 4$	140.5	141.9

We summarize three findings. **(1) KV sliding dominates, with benefits compounding at long horizon.** Without sliding (Group A), FVD rises from 439.6 at 10 s to 996.9 at 30 s, a $2.3\times$ degradation as the unbounded history strains memory and accumulates rollout errors. With sliding enabled (Groups B and C), FVD remains essentially flat across both horizons, with at most 3 FVD points of change. **(2) The RoPE cap has negligible empirical effect under sliding.** The no-cap and cap$= 16$ rows of Group B differ by only 0.9 FVD at 10 s and 2.0 FVD at 30 s. The reason is that, under a $K_r = 7$ sliding window, the relative distance between the noisy target and any recent frame is bounded to $[1, 7]$ exactly by construction, which already covers the training range for target–recent attention. The local-RoPE cap C instead bounds the relative distance from the target to the sink frame, which would otherwise grow unboundedly as p_t^{abs} advances. Empirically, the no-cap baseline still performs well because attention mass on the sink is small (the sink primarily encodes static

scene anchors), so the OOD positional regime there has limited effect on FVD. **(3) Window size $\mathbf{K_r = 7}$ aligns with training.** With $\texttt{kv}= 4$ (Group C), FVD stays within 3 points of the $\texttt{kv}= 7$ baseline at both horizons (140.5 vs. 138.6 at 10 s; 141.9 vs. 139.2 at 30 s). This confirms that, once the relative-position bound is in place, the exact window length exerts only a second-order effect on quality.

A.10 Stronger Action-Index Baselines Collapse into NL

A natural intermediate baseline that might appear stronger than our hash-projected Action-Index interface is one whose embeddings are factorized into separate entity and action tables and initialized from a frozen text encoder such as CLIP, Qwen, or Wan's own. We argue that any such baseline is, by construction, a vocabulary-restricted special case of NL conditioning and therefore does not constitute a separate operating point. We analyze its two natural instantiations as follows. **Frozen variant.** When we keep the text-initialized embeddings frozen, the baseline inherits exactly the pretrained semantic prior that NL exploits, and we therefore expect it to recover most of the Tier II and Tier III cross-entity gap. However, its inference vocabulary remains closed, so its OOV coverage on the four probes of Section 4.2 stays exactly $\mathbf{0}\%$. No choice of initialization can change a structural slot count. **Trainable variant.** When we unfreeze the embeddings, the text-initialized entries specialize to their training-time entity context within a few thousand steps, and the variant is empirically dominated by the joint-vocabulary Action-Index baseline reported in Section 4.2. In either case, a factorized text-initialized Action-Index interface either collapses into NL with a hand-fixed sub-vocabulary (frozen) or reverts to the joint-vocabulary baseline (trainable).

A.11 Axis 1: Full In-Distribution Score Distributions

We extend the main-body Axis 1 summary (Table 1) with the per-trial $2/1/0$ score distributions for both interfaces in Table 7. We evaluate both NL and Action-Index on the same 20 trials per action (5 starting frames $\times$ 4 seeds), with three blinded annotators and median rating. We select the five evaluated actions as the most frequent in-distribution actions per entity, since these dominate each entity's training data and therefore provide the strongest possible supervision for the Action-Index interface, ruling out long-tail artifacts as an explanation for any subsequent NL vs. Action-Index gap.

Table 7: **Axis 1 full score distributions.** Each cell reports the percentage of trials rated **2** (full execution), **1** (partial), and **0** (absent), with ACA(≥ 1) defined as the sum of the full and partial percentages.

	NL (ours)		**Action-Index**	
Action	$2\,/\,1\,/\,0\,(\%)$	ACA(≥ 1)	$2\,/\,1\,/\,0\,(\%)$	ACA(≥ 1)
Margit · Double light blade throw	$60\,/\,35\,/\,\ 5$	**95**	$45\,/\,40\,/\,15$	85
Margit · Staff slam	$70\,/\,10\,/\,20$	**80**	$50\,/\,30\,/\,20$	**80**
Knight · Shield block	$95\,/\,\ 5\,/\,\ 0$	**100**	$65\,/\,35\,/\,\ 0$	**100**
Knight · Overhead slash	$100\,/\,\ 0\,/\,\ 0$	**100**	$75\,/\,10\,/\,15$	85
Knight · Tail of the crucible	$70\,/\,30\,/\,\ 0$	**100**	$50\,/\,45\,/\,\ 5$	95
Mean (5 actions)		**95**		89

A.12 Axis 2: Full Cross-Entity Distributions and Per-Tier Mechanism

We extend the main-body Axis 2 summary (Table 1) with the per-trial $2/1/0$ score distributions in Table 8. We evaluate both interfaces on the same five cross-entity action pairs under identical annotation (20 trials per pair, 5 starting frames $\times$ 4 seeds, three blinded annotators with median rating, prompt-injection protocol of Appendix A.6). In the rest of this subsection we first define the three tiers used to organize the pairs, and then we decompose the cross-entity NL vs. Action-Index gap by tier.

Definition of the three tiers. We grade each cross-entity action pair (source action, target entity) by the relationship between the source action and the *target entity's* native action repertoire, since

this relationship determines what the Action-Index interface can possibly fall back on when an out-of-context index is injected. We adopt three tiers of increasing visual overlap:

- **Tier I (no overlap):** The source action is entirely absent from the target entity's native repertoire, with no morphologically related fallback animation available. For example, the "double light blade throw" is exclusive to Margit and has no counterpart in the Crucible Knight's repertoire. This is the most stringent regime for the Action-Index interface, since its index-to-embedding map has no nearest neighbor to recover.
- **Tier II (motion shared, visual style distinct):** The target entity possesses an action that shares the underlying motion class (e.g., a tail swipe) with the source action, but the two animations differ in a visually salient feature such as luminosity or trajectory shape. For instance, both Margit and the Knight execute a tail swipe, yet only the Knight's variant emits a luminous energy trail. Action-Index can fall back on the shared motion class, but it cannot produce the entity-specific visual feature.
- **Tier III (same action label, animation alignment varies):** Both entities possess an action carrying the same lexical label (slash, overhead, horizontal), yet the underlying animations may be more or less tightly aligned in timing and amplitude. This is in principle the easiest regime for Action-Index, since its embedding can in principle map onto a visually adjacent animation.

We choose this stratification so that the three tiers progressively give the Action-Index interface more and more chance to succeed via nearest-neighbor fallback. If the cross-entity NL advantage persists across all three tiers, then the gap cannot be attributed to a single failure mode of the Action-Index interface.

Table 8: **Axis 2 full score distributions.** Each cell reports the percentage of trials rated **2** (full) **/ 1** (partial) **/ 0** (absent), with ACA(≥ 1) defined as the sum of the full and partial percentages. We grade tiers by the degree of visual overlap between the source action and the target entity's native repertoire.

		NL (ours)		**Action-Index**	
Action prompt (source entity)	Target entity	2 / 1 / 0 (%)	ACA(≥ 1)	2 / 1 / 0 (%)	ACA(≥ 1)
Tier **I** *— action absent from the target entity's native repertoire*					
Double light blade throw (Margit)	Knight	65 / 15 / 20	**80**	0 / 15 / 85	15
Tier **II** *— target has a morphologically similar but visually distinct action*					
Tail of the crucible (Knight)	Margit	20 / 70 / 10	**90**	0 / 60 / 40	60
Tier **III** *— both entities share a same-named action; Action-Index may fall back to it*					
Heavy overhead slash (Margit)	Knight	95 / 5 / 0	**100**	60 / 15 / 25	75
Diagonal slash (Knight)	Margit	45 / 50 / 5	**95**	5 / 20 / 75	25
Horizontal slash (Margit)	Knight	70 / 10 / 20	**80**	0 / 40 / 60	40
Mean (5 pairs)			**89**		43

Per-tier decomposition of the gap. We now expand on the cross-entity result of Section 4.2 by tier. We argue that the three tiers of Table 8 together rule out every single-confounder explanation of the cross-entity gap.

Tier I (NL$= 80\%$, Action-Index$= 15\%$). We observe the largest single-pair gap of $+65$ pp here. The Crucible Knight is never paired with the "double light blade throw" index during training, so this tier is also the most stringent point of the prompt-injection protocol: the cross-entity action has zero training co-occurrence with the target entity, and the un-conditioned warm-up therefore anchors the model in an arbitrary Knight pose with no preparatory frames for the requested throw, which places visual coherence and semantic compliance maximally in tension. Under this regime, the Action-Index baseline reaches only 15% ACA, with all successes registering as partial rather than full execution (0% rating-2, 15% rating-1). We interpret this score-distribution pattern as the intended diagnostic signature of the protocol and as consistent with an embedding-leakage account: the Margit-trained "double light blade" embedding does carry enough action-specific visual gradient that, when injected into Knight context, the backbone occasionally synthesizes partial luminous-blade content; however, the Knight's vocabulary contains no nearest-neighbor action that visually resembles a thrown pair

of blades, so the embedding has no usable fallback, and the partial visual content fails to crystallize into a full execution. NL, in contrast, reaches 80% on the same prompt, with 65% rating-2 full executions and 15% rating-1 partials. The reason is that "double light blade throw" decomposes into subword units that the text encoder has seen in many training contexts (light, blade, throw), and the compositional meaning therefore transfers to the Knight without requiring any (Knight, light-blade) training co-occurrence. The language interface thereby largely resolves the visual-coherence-versus-semantic-compliance tension that the prompt-injection protocol creates by construction. We further note that, when the same student is evaluated under the trajectory-conditioned protocol used for the system-level metrics in Table 2, where ground-truth captions accompany the visual context from frame zero and the visual–semantic tension is removed, ACA reaches 90.4–93.2%, which is the regime in which the model is actually deployed during real-time play.

Tier II (NL$= 90\%$, Action-Index$= 60\%$). We observe a $+30$ pp gap on this tier. The source action is the Crucible Knight's luminous energy tail, while Margit possesses a structurally analogous tail-swipe action with a dark, opaque visual style. We attribute the relatively high non-zero Action-Index ACA (60%) to nearest-neighbor fallback: the injected index activates Margit's existing tail motion. However, Action-Index cannot encode the luminous visual feature, since this feature is specific to the Knight's animation. NL reaches 90%, with 70% of clips judged as partial rather than full execution. We attribute this to the fact that the prompt "Tail of the crucible" encodes both the motion and the luminous visual characteristic through its pretrained semantics, which suppresses Margit's native dark-tail prior. The high partial-execution rate suggests, however, that the luminous quality is rendered less crisply on Margit than on its native Knight animation in Tier III shared-action cases.

Tier III (Action-Index$\in [25\%, 75\%]$). We observe that, when the source action has a same-named counterpart in the target entity's native repertoire (slash, overhead, horizontal), the Action-Index nearest-neighbor fallback is in principle the strongest, because the index-to-embedding map should land on a visually adjacent animation. Empirically, however, the fallback is sharply contingent on *animation alignment* rather than on the shared label. We illustrate this through three pairs of increasing animation mismatch. *Heavy overhead slash* is the most aligned pair, since Margit's and the Knight's overhead slashes share both timing and amplitude; Action-Index therefore reaches 75% ACA, the highest cross-entity Action-Index number in the table. *Horizontal slash* carries an identical lexical label, but the two underlying animations differ in sweep duration; Action-Index drops to 40%. *Diagonal slash* is the most extreme case: the Knight's variant is a short jab while Margit's is a wide arcing swing, so the shared index lands on visually mismatched footage, and Action-Index falls to 25%. We highlight that this Tier III value (25%) is even *below* the Tier II Action-Index number (60%), where at least the underlying motion class is shared. The variation across Tier III is therefore governed by animation-level visual specificity rather than by the lexical fact that the two entities share an action name. NL, in contrast, stays above 80% on all three pairs, because language conditions on motion semantics independently of which animation index the target entity happens to own. We further note the residual NL $\geq$ Action-Index gap on the easiest possible Action-Index case (Heavy overhead slash, $+25$ pp), which indicates that language provides richer steering even on unambiguous shared actions, rather than merely compensating for missing vocabulary entries.

Summary. The three tiers together rule out every single-confounder explanation of the cross-entity gap. The NL advantage is not merely that "Action-Index has no embedding for the action" (Tier **I**, $+65$ pp), nor merely that "Action-Index picks the wrong visual style" (Tier **II**, $+30$ pp); rather, it persists even when Action-Index has both the right embedding and the right visual style (Tier **III**, Heavy overhead slash, $+25$ pp). We therefore conclude that the gap tracks the structural property that NL possesses and Action-Index lacks, namely the compositional decomposition of action prompts into entity-independent semantics, rather than any single failure mode of the index-bound interface.

VLM pairwise corroboration. To provide an automated sanity check complementary to the human annotation, we run a vision-language model (VLM) judge on the same five cross-entity pairs. For each pair we uniformly sample 10 frames from the action burst segment of each video and present all 20 frames to `gemini-3.1-flash-lite-preview` (accessed via API), randomising which video is labelled A and which is B. The model is prompted with the action name, a concise visual definition, and asked to (i) score each video on a 0–3 action-fidelity scale and (ii) declare a winner (*A*, *B*, or *tie*). We repeat this for n=20 matched NL/Action-Index pairs per action and report the fraction of trials won by each interface (Table 9).

Table 9: **VLM pairwise judge (10 frames per video, n=20 pairs, `gemini-3.1-flash-lite-preview`).** Each trial presents both videos with randomised A/B assignment; the model chooses which clip better executes the named action. NL win% and ID win% are the fractions of trials won by each interface; ties account for the remainder. Δ = NL win% − ID win%.

Tier	Action	NL win%	ID win%	Δ
I	Double light blade	85	15	+70 pp
II	Tail of the crucible	50	50	≈0 pp
III	Heavy overhead slash	60	40	+20 pp
III	Diagonal slash	55	45	+10 pp
III	Horizontal slash	60	35	+25 pp

The VLM results largely corroborate the human annotations. On Tier I, the model assigns NL a +70 pp advantage, consistent with the +65 pp human gap and confirming the most unambiguous transfer case. On Tier III, NL leads by +10–+25 pp across all three pairs, matching the human-observed pattern that Action-Index can partially fall back on a same-named animation but cannot match the steering precision of language. The sole divergence is Tier II (*Tail of the crucible*): the VLM produces chance-level agreement (50%/50%), because the luminous energy tail is an out-of-distribution visual element on Margit—the VLM has no game-domain prior to distinguish it from her native golden-blade effects, and therefore cannot reliably judge which clip is correct. This reflects an inherent limitation of general-purpose VLM judges on domain-specific cross-entity transfers: when the target action is visually OOD for the target entity, only human annotators with game context can evaluate it reliably.

A.13 OOV Probe Set

We evaluate four out-of-vocabulary (OOV) probes referenced in Section 4.2, each obtained by editing exactly one content word of a top-3-by-frequency in-vocabulary base prompt. We run each probe for 10 trials (5 starting frames × 2 seeds) under the prompt-injection protocol of Appendix A.6, and we have two annotators rate each clip on the same ordinal scale used elsewhere; ACA(≥ 1) reports the sum of correct and partial ratings. By construction, none of the four probes corresponds to an index in the 47-way joint vocabulary, so the Action-Index interface returns no output for any of them and its coverage is structurally **0**%, regardless of model capacity.

Table 10: **OOV probe results (NL).** Per-probe ACA on the four-probe evaluation set, with aggregate ACA of **90**% over 40 trials. Action-Index coverage on each probe is structurally **0**%.

Base action (entity)	OOV prompt (edit type)	Trials	ACA(≥ 1)
Overhead slash (Knight)	`Downward slash.` (synonym)	10	100
Tail of the crucible (Knight)	`Crucible tail.` (abbreviation)	10	100
Shield block (Knight)	`Shield guard.` (synonym)	10	90
Double light blade throw (Margit)	`Dual light blade throw.` (synonym)	10	70
Mean (4 probes)		40	**90**

Excluded probe candidates. We exclude two probes from the original candidate set based on setup-side issues that are upstream of the language interface itself, namely cases in which the target action is under-specified by the prompt or insufficiently distinguishable from neighboring classes by annotators. We summarize the two excluded candidates below. *`Aerial slam.`* (paraphrase of *Jump and mid-air slam*) is confounded with other aerial-attack classes that share its semantic surface form. We observe that the rendered behavior disagrees with the intended target action across all 10 trials, which indicates that the probe under-specifies the target rather than that the interface fails to act on the prompt. *`Staff swing.`* (intended as a synonym of *Staff upswing*) is overly generic in Margit's repertoire, where it overlaps with *Staff slam* and *Charged staff thrust*; annotators report sustained ambiguity in distinguishing the intended action from these alternatives. We therefore exclude both candidates from the quantitative aggregates rather than scoring them as "NL failures," since the

failure mode is upstream of the language interface itself. We release the per-trial ratings for both excluded probes alongside the codebase for transparency.

A.14 Failure Case Analysis

We identify two residual failure modes of the deployed system, and we discuss each below together with its mitigation.

(1) Rapid motion blur (rare, $< 3\%$ of frames). We observe that, during high-speed boss attacks with extreme camera shake, the 2-step student occasionally produces visual artifacts. We mitigate this by increasing the guidance scale to 4.5 for frames classified as "Boss performing leaping attack."

(2) Hit-detection false positives ($< 5\%$ of windows). We observe that the VLM-based hit detector occasionally misclassifies non-damaging contact, for instance a weapon-on-shield parry, as a hit event. The rule engine tolerates such sporadic false positives because the hit-counter threshold provides natural error buffering, and the persistent state therefore remains stable over long episodes.

A.15 Long-Horizon Stability

We probe whether the Self-Forcing student on Elden Ring, with its $1.75\,\mathrm{s}$ training context and RoPE-decoupled KV-cache sliding window, maintains generation quality far beyond its training horizon. To this end, we render 5 independent rollouts of $\sim$118 minutes each on Margit under the trajectory-conditioned protocol of Appendix A.6, and we evaluate FVD on 200-second sliding windows starting at $t = 30\,\mathrm{min}$. Within each window, we extract 15 uniformly spaced 10-second clips per video ($5 \times 15 = 75$ clips per window) and compare them against the held-out 10-second test set using I3D features with *intersect* = *False*. We choose a window length that matches the test set's per-clip duration so that the statistics are computed on directly comparable distributions.

We report the resulting FVD trajectory in Table 11. Across the full 30-to-118-minute interval, which corresponds to 88 minutes of monitored generation per video and $27 \times 75 = 2{,}025$ evaluated clips in total, FVD stays in $[162.4, 171.3]$ with mean 166.0 and standard deviation 2.3. We observe no monotonic degradation trend: the largest value (171.3) occurs early in the monitored interval (the 33–37 min window), and the latest window (117–118.3 min) reads 164.5, which is indistinguishable from the global mean. For reference, the Stage-1 teacher's 50-step FVD on the same test split is 206.2 (Table 2), so the long-horizon student stays roughly 40 FVD points below the teacher even after running for two orders of magnitude longer than its training context.

Table 11: Long-horizon FVD on 5 independent $\sim$118-minute Margit rollouts. Each row reports a 200-second sliding window with 15 uniformly sampled clips per video (75 clips per window, 2,025 clips in total). Evaluation follows the same I3D protocol as Table 2 on the held-out 10-second test set.

Window (min)	FVD	Window (min)	FVD	Window (min)	FVD
30.0–33.3	170.3	60.0–63.3	168.6	90.0–93.3	164.7
33.3–36.7	171.3	63.3–66.7	165.1	93.3–96.7	162.4
36.7–40.0	167.4	66.7–70.0	164.3	96.7–100.0	163.5
40.0–43.3	168.5	70.0–73.3	164.6	100.0–103.3	165.2
43.3–46.7	167.0	73.3–76.7	166.4	103.3–106.7	164.9
46.7–50.0	165.9	76.7–80.0	167.7	106.7–110.0	170.3
50.0–53.3	165.8	80.0–83.3	164.1	110.0–113.3	164.4
53.3–56.7	166.4	83.3–86.7	163.7	113.3–116.7	165.4
56.7–60.0	166.3	86.7–90.0	164.6	116.7–118.3	164.5

Aggregate over 27 windows: mean 166.0, std 2.3, range $[162.4, 171.3]$.

We attribute this stability to the bounded RoPE-decoupled KV-cache sliding window of Section 3.2: by holding the rotary positional state inside the model's training distribution while allowing the sliding cache to attend over recent visual history, the generator avoids the positional drift that typically destabilizes long autoregressive video rollouts. This long-horizon stability also serves as the empirical foundation for the closed-loop playable system described below, which presumes that the underlying student remains visually well-behaved across multi-minute episodes.

A.16 Extension: Persistent Entity State via an Observer–Tracker–Policy Loop

The two architectural patterns in the main paper (Sections 3.1 and 3.2) together deliver a real-time, language-controllable video generator. However, per-frame controllability alone does not yet deliver *long-horizon interactivity*. Any sufficiently long interactive video episode carries *entity-level discrete state* that evolves on a substantially slower timescale than the generator's attention context, including task progress in embodied manipulation, phase state in narrative video, fuel or route state in driving, and damage and phase state in adversarial combat. Such state is not always recoverable from pixels within the current context window, is discrete rather than pixel-continuous, and must remain consistent across hundreds or thousands of frames. Standard video diffusion backbones provide no mechanism to maintain it.

We describe in this appendix an extension that addresses the gap. We present it as an extension rather than as a core contribution for three reasons: (i) the interface claim and its supporting empirical evidence stand independently of it; (ii) the instantiation is hand-specified per domain; and (iii) the loop described here is the first domain-specific instantiation rather than a fully general implementation, while a world-agnostic learned version is left as future work.

A.16.1 The Memory Gap is Structural

The generator attends over a bounded context of $K_s + K_r + K_n = 9$ latent tokens, spanning $\sim$1.75 s of generated video through the recent window. Episode-level entity state, in contrast, evolves over minutes. The resulting $\sim 100\times$ temporal gap cannot be closed by simply scaling the backbone: doubling the context still leaves a $\sim 50\times$ gap, and discrete state transitions are not a quantity that video pretraining optimizes for. Empirically, we observe that a model trained on 45 hours of Elden Ring combat data (30 h Margit and 15 h Crucible Knight), which contains frequent player deaths and multiple boss-death sequences, never spontaneously triggers a terminal state (Section A.16.4). The model faithfully renders every individual event, yet it never accumulates them into a state transition. We therefore conclude that this behavior is not a training-data artifact but a direct consequence of the memory-horizon mismatch.

A.16.2 A Three-Part Loop

We close the gap with an additive module that makes the asymmetry explicit: the neural backbone renders *what the world looks like*, while an external module maintains *what the world is in*. We decompose this external module into three roles, which we describe below.

(1) Observer: structured event extraction from the generated video stream. We use an event extractor that operates on the rendered video itself and emits discrete, structured signals at every action window. We adopt a VLM rather than a classical detector because the extracted events are typically semantic ("did entity E take damage?", "did task T complete?", "did two entities collide?") rather than low-level visual. In our instantiation, we use Qwen3-VL-2B-Instruct, which we fine-tune to emit a binary damage event per entity per 0.25 s window (Section A.16.5). We emphasize that the Observer also compensates for passive-response events that we deliberately exclude from the Stage 1 prompt vocabulary: the generator is never conditioned to produce them, but the Observer reads them back off the rendered pixels.

(2) Tracker: unbounded-horizon accumulation of entity state. We use a lightweight external state machine to aggregate the Observer's event stream into structured entity state that persists across the full episode. Unlike the bounded diffusion context, the Tracker has an unbounded temporal horizon: it integrates events from the first frame to the current frame at negligible cost. The Tracker's internal representation is typed and discrete (integer counters, categorical phase labels, or structured records), and therefore matches the actual semantics of episode-level state far more naturally than any pixel-latent representation could. In our instantiation, the Tracker maintains per-entity integer HP counters that it updates from the Observer's damage stream.

(3) Policy: state-conditioned reinjection into the language interface. When the Tracker's state crosses a relevant threshold or transition condition, the Policy advances the episode to the corresponding phase and selects the next action prompt to inject into the generator. We design the Policy to be deliberately minimalist: its role is not to perform complex reasoning but to expose the Tracker's state back to the generator *through the same language interface used for control*. This is what makes the loop architecturally free, since the generator sees only per-frame language prompts, regardless of

whether these prompts describe user-issued actions or state-triggered transitions, and no modification to the generator is required. In our instantiation, the Policy maps HP thresholds to phase labels (normal combat, stagger, execution, terminal) and selects the corresponding prompt template.

A.16.3 Related Work on External Memory for Generative World Models

Video diffusion models operate over a bounded context window, which makes them intrinsically ill-suited to tasks that require persistent state across hundreds of frames. This limitation has motivated a family of approaches that couple neural generators with external memory. In NLP, retrieval-augmented [25] and memory-augmented [42] architectures delegate long-term dependencies to an explicit memory store that the neural model writes to and reads from. In interactive world modeling, NeSyS [49] constrains LLM-based simulators with executable rules to reduce hallucination, BlendRL [34] interleaves symbolic and neural policies within a single RL agent on Atari, MultiGen [29] maintains an external memory for editable diffusion game engines, and LiveWorld [12] persists entity evolution while entities are out of view. In generative modeling more broadly, symbolic state has been injected into diffusion processes through score manipulation [32], interleaved symbolic optimization [10], logic-guided vector fields [3], and physical-consistency constraints [26]. Classical neuro-symbolic game AI also couples symbolic priors with neural policies [14, 35, 41], although it operates at the policy level rather than at the generator level. Relative to these prior approaches, our loop does not modify the generator's score, policy, or attention. It is a purely additive module that bridges the diffusion backbone's ~ 1.75 s recent context with the horizons on which entity-level discrete state actually evolves.

A.16.4 Episode-Level State Accumulation

Protocol. We run independent 10-minute test episodes and inject attack prompts at regular intervals after a calibration period. A correct system must trigger a terminal state (an entity-death sequence or execution animation) once the number of accumulated damage events crosses the per-entity threshold. We run the protocol twice under identical prompting, once with the Observer–Tracker–Policy loop attached and once without.

Result. Without the loop, no episode terminates: the neural backbone faithfully renders every individual damage animation, yet it never spontaneously transitions to a terminal sequence, since no mechanism internal to the generator can accumulate the integer count required to cross the threshold across the ~ 1.75 s local context. With the loop attached, the Tracker issues a reliable termination signal whenever its state crosses the threshold, and the Policy injects the corresponding terminal prompt back into the generator. This result directly confirms the structural prediction of Section A.16.1: the memory gap is not a training-data artifact but a backbone-horizon mismatch that cannot be closed by scaling the generator alone.

A.16.5 Observer: VLM Event Extraction

The Observer serves as the bridge between the neural backbone's ~ 1.75 s local context and the episode-level state maintained by the Tracker. It operates on the generated video stream itself and emits a binary damage event per entity at every action window. In our instantiation the event is *Taking Hit* / *No Hit*, and we deploy two Observers, one per entity, using Qwen3-VL-2B-Instruct [4] as the backbone. We fine-tune on $4{,}477$ damage-event windows and $16{,}309$ non-event windows of 0.25 s each. To inject domain-specific visual priors such as blood splatters, attack contact, and stagger animations, we encode them as auxiliary instructions in the VLM prompt. To capture event-specific spatio-temporal dynamics while preserving pretrained knowledge, we update only the vision–language connector and the cross-attention modules; training completes in 8 hours on a single H100.

We evaluate both Observers on a held-out test set with binary damage-event labels (Table 12) and achieve over 90% on every per-class Precision/Recall/F1 metric. We further conduct a complementary user study on distilled-generator video, and we obtain comparable numbers, which confirms robustness under the distribution shift from real to generated footage and hence under closed-loop deployment, namely the regime in which the Observer actually operates when the full loop is running.

Table 12: Observer evaluation (damage-event instantiation) on the held-out test set vs. a user study on generated video. The user-study setting matches the distribution that the Observer actually sees inside the closed loop.

Detector	Evaluation	Class	Precision (%)	Recall (%)	F1 Score (%)
Boss	Test Set	No hit	97.08	90.01	93.41
		Taking hit	93.50	98.15	95.77
	User Study	No hit	87.18	82.93	85.00
		Taking hit	92.39	94.44	93.40
Player	Test Set	No hit	96.01	97.13	96.57
		Taking hit	97.02	95.86	96.44
	User Study	No hit	97.42	92.92	95.12
		Taking hit	85.80	94.56	89.97

A.16.6 Scope and Limits of This Extension

We deliberately hand-specify the loop per domain: each domain requires its own event schema (what the Observer extracts), its own state schema (what the Tracker maintains), and its own transition table (what the Policy emits). Our current instantiation is tailored for combat, with binary damage events, integer HP counters, and HP-threshold phase transitions. Extending the loop to a new domain therefore requires re-annotating the Observer and rewriting the Tracker schema. A world-agnostic, learned version of the loop, in which all three schemas are inferred directly from event-labeled video, is an obvious next step but lies outside the scope of this paper.

A.17 Real-Time Inference: Streaming Pipeline

Interactive deployment requires that the per-chunk generation latency not exceed the chunk's playback duration. With $C{=}2$ new latent frames per chunk and $4\times$ temporal compression at 16 fps, each chunk represents 500 ms of video; the DiT's KV-cache sliding window spans 7 latent frames ($\sim$1.75 s) of attended context. On a single H100, this constraint cannot be met by a fully sequential DiT–VAE schedule; it requires overlapping the two stages in time. We describe the pipeline redesign and quantify its effect.

A.17.1 Sequential Baseline

In the unmodified schedule, all N latent frames are produced by the DiT before VAE decoding begins. DiT and VAE thus occupy non-overlapping GPU time slots. With $C{=}2$ and the Wan backend, the per-chunk averages are 501 ms (DiT), 432 ms (VAE), and 37 ms (write), so the sequential pipeline processes one chunk every $\sim$970 ms—a $1.94\times$ real-time ratio for the 500 ms of video each chunk represents.

A.17.2 Chunk-Based Streaming

We replace the original sequential schedule with a *chunk-based producer–consumer pipeline*. Instead of generating the full latent sequence before decoding, the DiT generates latent windows incrementally and submits them to the VAE as soon as they become available. Each yielded window contains a fixed number of newly generated latent frames, optionally augmented with a small overlap of preceding latents to preserve temporal continuity across chunk boundaries.

In this mode, the DiT and VAE run on separate CUDA streams. After the DiT finishes assembling a latent window for a chunk, the host clones that window into a dedicated contiguous buffer and records a CUDA event on the DiT stream. The VAE stream waits on this event before consuming the corresponding latent buffer. This enables asynchronous GPU-side pipelining: latent generation for later chunks can overlap with VAE decoding of earlier chunks whenever hardware resources permit. The host does not impose a per-chunk global synchronization barrier, although it may occasionally wait on the oldest pending decode event to preserve ordered video emission.

Read-after-write hazard. The implementation does not expose the generator's transient latent window directly to the VAE. Instead, each yielded latent window is materialized as a separate cloned buffer before being submitted to the decode stream. This avoids lifetime and aliasing hazards while the producer continues advancing to subsequent chunks. In other words, the VAE always consumes an immutable per-chunk latent snapshot rather than a tensor that may later be modified or discarded by the generation path.

Because each in-flight decode job owns its own latent copy, the extra memory overhead is not constant in video length but is bounded by the chunk window size and the number of queued in-flight jobs. In practice, this overhead scales with the latent window size multiplied by the queue depth limit.

Bounded in-flight queue. When the VAE is substantially faster than the DiT (e.g., with the TaeHV backend, where VAE averages 9 ms against DiT's $\sim$362 ms), the decode queue drains near-instantly and presents no buildup risk; the bound instead guards against worst-case transient spikes or future configurations where the VAE cost approaches or exceeds the DiT cost. To prevent unbounded accumulation of pending decode jobs, we maintain a bounded FIFO queue of in-flight chunks. The producer is allowed to submit new work only while the number of pending jobs remains below a fixed queue depth Q. Once the queue is full, the host dispatch loop temporarily stops submitting additional chunks until earlier decode jobs complete and are drained.

Each queued job carries its chunk index and completion event. Completed chunks are written in submission order, ensuring temporal consistency even if decode completion times vary slightly across chunks. In practice, when the VAE is much faster than the DiT, the queue remains nearly empty and the bound is rarely exercised; its primary role is to cap memory usage and provide robustness under less favorable speed ratios or transient execution spikes.

A.17.3 Temporal Consistency

The Wan VAE decoder is temporally convolutional: decoded pixel values at chunk boundaries depend on latents outside the chunk window. Decoding isolated chunks therefore introduces inter-chunk discontinuities. We mitigate this by prepending L latent frames from the preceding chunk to each VAE input. The corresponding L decoded frames are discarded after decoding; only the C non-overlapping frames are retained. This is the pixel-domain analogue of the latent-domain KV-cache sliding in Section 3.2: both use a bounded overlap window to preserve causal context without unbounded history accumulation.

A.17.4 Timing Analysis

Each chunk of C=2 latent frames decodes to $C \times 4 = 8$ pixel frames, representing 500 ms of video at 16 FPS. We report per-chunk averages to expose stage-level costs independently of clip length; pipeline throughput is the measured active compute per chunk, directly comparable to the 500 ms budget.

Table 13: **Per-chunk pipeline timing (single H100, 2-step distilled student,** 480×832**,** C=2**,** `kv_window`**=7,** `local_rope_cap`**=12).** Each chunk produces 8 pixel frames (500 ms of 16 FPS video). DiT and VAE columns are per-chunk averages; *throughput* is measured active compute per chunk (accounting for stream overlap); *Eff. FPS* and RT ratio are derived from throughput. Values below $1.0\times$ indicate active-compute throughput exceeds real time.

Overlap L	VAE	DiT avg (ms)	VAE avg (ms)	Throughput (ms/chunk)	Eff. FPS	RT ratio
3	Wan	501	432	789	10.1	$1.58\times$
1	Wan	504	236	596	13.4	$1.19\times$
3	TaeHV	363	9	409	19.6	$0.82\times$
1	TaeHV	361	9	406	**19.7**	$\mathbf{0.81}\times$

Three findings emerge from Table 13.

(1) The Wan VAE dominates per-chunk cost. At L=3, VAE decode averages 432 ms per chunk—55% of the 789 ms throughput. Reducing overlap to L=1 cuts the VAE cost by 45% to 236 ms by eliminating redundant boundary decodes, lowering throughput to 596 ms; the pipeline remains sub-real-time ($1.19\times$).

(2) TaeHV eliminates the VAE bottleneck. The TaeHV lightweight decoder averages 9 ms per chunk regardless of L, reducing the VAE share to $< 3\%$ of throughput. The pipeline becomes DiT-bound at $\approx$361–363 ms/chunk (DiT), and measured throughput reaches 406 ms/chunk—below the 500 ms budget, corresponding to 19.7 FPS effective output at 16 FPS target ($0.81\times$ real-time ratio).

(3) TaeHV reduces DiT cost. DiT average falls from $\sim$502 ms/chunk (Wan) to $\sim$362 ms/chunk (TaeHV), a 28% reduction. TaeHV operates in a lower-dimensional latent space, shortening the token sequence seen by the DiT attention layers and reducing the per-step cost of KV-cache construction and sliding.

Model loading ($\sim$68–70 s) and first-frame VAE encode ($\sim$0.3–1.6 s) are one-time startup costs that do not recur across chunks; in closed-loop deployment they are amortized over the full episode.

A.18 Licenses for External Assets

We list all external assets used in this work, together with their verified license terms.

Wan 2.2. We use the Wan 2.2 `TI2V-5B` variant as our Stage 1 backbone [39]. The model weights are released by Alibaba Group under the **Apache License 2.0**. The full license text is available at `https://huggingface.co/Wan-AI/Wan2.2-TI2V-5B/blob/main/LICENSE.txt`.

TAEHV. We use TAEHV (Tiny AutoEncoder for Hunyuan Video) [7] for real-time VAE decoding during streaming inference. TAEHV is released by Ollin Boer Bohan under the **MIT License** (Copyright © 2025 Ollin Boer Bohan). The full license text is available at `https://github.com/madebyollin/taehv/blob/main/LICENSE`.

Qwen3-VL-2B-Instruct. We fine-tune Qwen3-VL-2B-Instruct [5] with LoRA for action annotation during dataset construction. The model weights are released by Alibaba Cloud under the **Apache License 2.0**. The full license text is available at `https://github.com/QwenLM/Qwen3-VL/blob/main/LICENSE`.

Elden Ring. *Elden Ring* (© 2022 Bandai Namco Entertainment Inc. / © 2022 FromSoftware, Inc.) is a commercial video game. All in-game footage used in this work was self-recorded for non-commercial academic research purposes, in accordance with the BANDAI NAMCO Entertainment End User License Agreement (EULA, last updated April 1, 2018).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction (Section 1) clearly state the contributions. These claims are directly supported by the experimental results in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: Limitations are discussed in Section 5 and Appendix A.2.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper doesn't include theorems, lemmas, or formal proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Full architecture details, training hyperparameters, hardware configuration, data construction pipeline, and evaluation protocols are provided in Sections 3–4 and Appendices A.6–A.17. Full code, checkpoints, and the dataset will be released soon.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We include partial dataset and code in the supplementary material, and will release the full dataset, code and checkpoints soon.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training hardware, optimizer, learning rates, batch sizes, resolution curriculum, context window size, and evaluation protocols are fully specified in Sections 3 and 4.1, and in Appendix A.6.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Appendix A.7 clearly reports the statistical confidence of our evaluation protocols.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: GPU type, count, training step counts for each stage, and per-frame inference latency are reported in Sections 3 and 4.3.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The work uses commercially available game footage for purely academic research, involves no private personal data or human subjects, and raises no safety or fairness concerns specific to the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Section 5 and Appendix A.2 discuss positive applications and limitations. Although our experiments are restricted to game-world video, more capable interactive video generation systems could be misused for deceptive synthetic media, including deepfakes or disinformation; this risk motivates careful release practices and clear domain restrictions.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The model is trained exclusively on video game footage to generate game-world video, posing no meaningful misuse risk.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All used assets and explicit license terms are properly cited and mentioned in Appendix A.18.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We introduce new assets, each accompanied by dedicated documentation in the supplementary material submitted together with the paper.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects; all annotators are co-authors of this paper.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects; all annotators are co-authors of this paper.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this paper does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.