RIGEN: BEYOND ACTION FOLLOWING WITH RECIPROCAL INTERACTION IN GENERATIVE GAMEPLAY

Anonymous authors
Paper under double-blind review

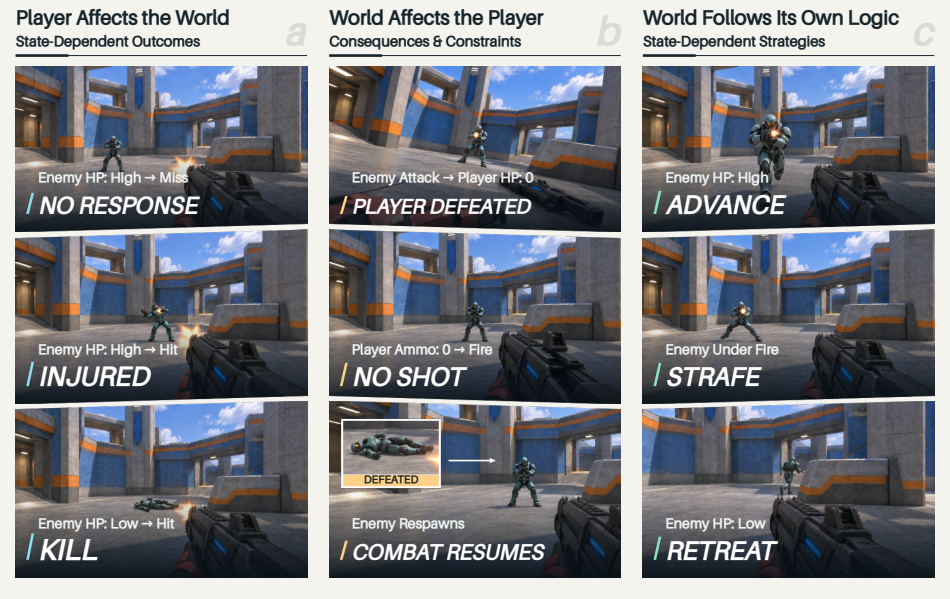

Figure 1: **Reciprocal interaction in RIGen.** Players affect the generated world, whose rules, persistent state, and characters in turn shape gameplay. (a) Attacks have state-dependent consequences. (b) Enemy actions, resource constraints, and respawn rules shape the player's survival and continued play. (c) Opponents adapt their behavior to their health and current combat conditions.

ABSTRACT

Generative gameplay requires more than immediate responses to player actions. Actions should have persistent consequences, while the generated world should act according to its own state and rules, shaping the player's situation and subsequent choices. Video generation captures local dynamics, but limited history, partially observed game state, and the need for contextual behavior decisions make such interactions difficult to sustain through audiovisual prediction alone. We introduce RIGen (Reciprocally Interactive Generation), a framework for reciprocal interaction in generative gameplay. RIGen combines persistent state and configurable rules, contextual interaction reasoning, and action-conditioned audiovisual generation. Together, these components make interaction consequences and actor behavior jointly determine generated content while maintaining consistency between game state and visible outcomes. We construct a Xonotic gameplay dataset to train an audiovisual generator for learning dynamics and interactions. Further post-training enables RIGen-Arena, an interactive generative first-person shooter (FPS) prototype. Experiments demonstrate more reliable combat outcomes and reciprocal interaction between the player and the generated world. We will release all models, datasets, code, and data creation pipelines as open source. Project page: `https://rigen-anonymous.netlify.app`.

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

Table 1: **Capabilities of generative gameplay systems.** Player→World denotes player actions changing the state of objects or characters in the world. World→Player denotes world events changing the player's gameplay state, such as health or available actions. NPC planning denotes explicit scene- and state-conditioned behavior reasoning. New scenes tests transfer to unseen scenes.

	Control		Gameplay			Output	
Method	Navigation	Actions	Player→World	World→Player	NPC Planning	Audio	New Scenes
GameNGen (Valevski et al., 2025)	✓	✓	✓	✓	–	–	–
SCOPE (Tong et al., 2026)	✓	✓	–	–	–	–	✓
LingBot-World 2.0 (Gao et al., 2026)	✓	✓	✓	–	–	–	✓
Incantation (Zhu et al., 2026b)	✓	✓	✓	✓	–	–	–
HelloWorld (Ouyang et al., 2026)	✓	✓	–	–	–	✓	✓
EchoWM (Zhang et al., 2026)	✓	–	–	–	–	✓	✓
RIGen (Ours)	✓	✓	✓	✓	✓	✓	✓

1 INTRODUCTION

Generative models offer a route to game experiences whose environments, characters, and events can be created during play (Bruce et al., 2024; Yu et al., 2025). Such experiences could be personalized to each player's preferences and choices. Realizing this potential requires reciprocal interaction: players should be able to change the world through their actions, while the world follows its own operating logic. Its characters act according to their situation and goals, and its rules and events shape the player's circumstances and subsequent choices.

Recent video-based world models have made progress toward this goal. DIAMOND (Alonso et al., 2024) and GameNGen (Valevski et al., 2025) demonstrate playable game simulation. LingBot-World (Robbyant Team et al., 2026) and SCOPE (Tong et al., 2026) provide responsive camera and movement control, while LingBot-World 2.0 (Gao et al., 2026) and Incantation (Zhu et al., 2026b) incorporate semantic guidance for character behavior and interaction events. Table 1 compares these systems across control, gameplay, and output capabilities. It highlights the remaining gap in combining reciprocal state changes with scene- and state-conditioned actor planning.

The prevailing approach builds on video models that capture rich local dynamics and generate continuous audiovisual feedback conditioned on player controls, making them a natural foundation for interactive games. However, sustaining reciprocal interaction through audiovisual prediction alone faces three challenges. **(1) Coordinating reciprocal interaction.** Responding to player actions does not by itself determine how the world should act upon the player. Actors must choose behavior based on current conditions and their own goals, and the consequences must update player state and constrain subsequent interactions. **(2) Limited history.** Events beyond the model's context window can still have consequences for current interactions (Xiao et al., 2025). **(3) Partially observed state.** Gameplay variables such as health and ammunition may not be recoverable from the visible scene, yet determine whether an action can be executed and what outcome it should have.

In sustained combat, for example, accumulated damage must determine whether the next hit is lethal, even when earlier attacks fall outside the video context. The opponent's remaining health and goals should shape whether it attacks or retreats. Damage dealt by the opponent must likewise affect the player's health and ability to continue fighting.

To address these challenges, we introduce RIGen, a framework for reciprocal interaction in generative gameplay (Figure 2). The *state and rules* layer maintains persistent gameplay variables and configurable rules, including strategic goals and constraints. The *interaction reasoning* layer interprets the scene and player actions to infer events and plan actor behavior under these conditions. The *audiovisual generation* layer realizes the resulting guidance alongside player controls. Together, these layers let player actions change the state that governs subsequent interactions. The world also acts according to its own rules and goals, affecting the player's state and available actions.

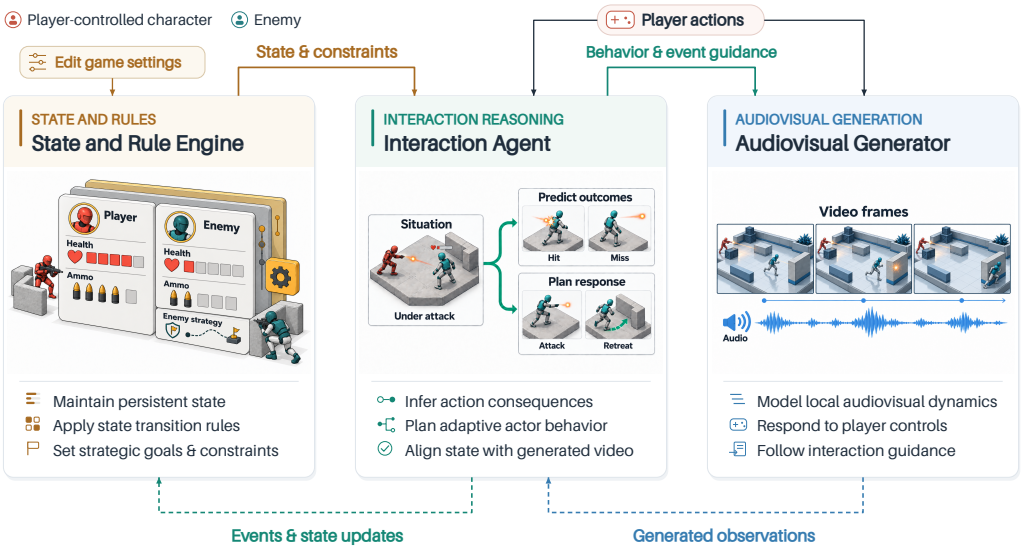

Figure 2: **Three coupled responsibilities in RIGen.** State and rules maintain persistent gameplay state, apply transition rules, and expose editable game settings and strategic constraints. Interaction reasoning infers action consequences and plans behavior. Audiovisual generation realizes local dynamics under player controls and guidance, returning observations for subsequent decisions.

We instantiate RIGen in first-person shooter (FPS) environments. Our experiments assess combat outcomes, while qualitative demonstrations and RIGen-Arena illustrate state-dependent interactions and autonomous enemy behavior that affects the player (Figure 1).

Our contributions are threefold. **(1) A system for reciprocal generative gameplay.** We integrate interaction prediction, executable state updates, and audiovisual generation into a closed loop in which player actions change world state and world behavior affects the player. Model comparisons and controlled ablations demonstrate more reliable realization of state-dependent interaction outcomes. **(2) FPS gameplay dataset.** We construct a dataset covering basic dynamics, controlled interactions, and sustained combat, pairing audiovisual observations with player controls and interaction annotations, with a reproducible data creation pipeline. **(3) Interactive gameplay prototype.** We build RIGen-Arena, integrating few-step audiovisual generation with closed-loop interaction to support continued play under persistent game rules.

2 Related Work

Video generation and interactive worlds. Video and audiovisual foundation models provide appearance and dynamics priors (Kong et al., 2024; Wan Team et al., 2025; Ruan et al., 2023; Polyak et al., 2024; HaCohen et al., 2026; Su et al., 2026). Controllable visual simulation (Kim et al., 2020; Menapace et al., 2021; 2022) has developed into generative game environments (Bruce et al., 2024; Valevski et al., 2025; Alonso et al., 2024; Kanervisto et al., 2025; Decart et al., 2024; Guo et al., 2025) and interactive content creation (Che et al., 2025; Yu et al., 2025; Li et al., 2025b; Zhang et al., 2025; Ye et al., 2025). Recent work broadens scene coverage and streaming control (He et al., 2025a; Tong et al., 2026; Robbyant Team et al., 2026; Zhu et al., 2026a; Huang et al., 2026a), adds multiplayer or audiovisual interaction (Hu et al., 2026; Zhang et al., 2026), and improves long-horizon consistency through memory (Xiao et al., 2025; Wu et al., 2026; Sun et al., 2025; Wang et al., 2026b; AlayaWorld Team et al., 2026). We focus on persistent gameplay state and rule-governed interaction consequences.

Actor control and semantic interaction. Reactive opponents and language-guided character control extend player-driven generation (Agarwal et al., 2026; Wang et al., 2026a; Cao et al., 2026; Ouyang et al., 2026; Tang et al., 2025). LingBot-World 2.0 (Gao et al., 2026) uses pilot and director

agents for character control and scene events. Incantation (Zhu et al., 2026b) tracks damage after generation, leaving hit/miss resolution to the video model and risking delayed death responses during fast-paced combat. RIGen couples behavior planning with executable gameplay rules, predicts interaction consequences before generation, and reconciles state with committed video.

Explicit state and generative rendering. Prior systems combine structured state or game logic with generative rendering (Li et al., 2025a; Cheng et al., 2025; Chen et al., 2025; Li et al., 2026; Lin et al., 2026b; Cai et al., 2026), while planning-based video methods organize content before synthesis (Lin et al., 2024; Lian et al., 2024). Geometric rendering interfaces provide explicit spatial control (Huang et al., 2026c; Lin et al., 2026a; Meng et al., 2026). Programmable World Model (Huang et al., 2026b) compiles state-augmented 3D bounding boxes into rendering controls, while Code World Model (Chen et al., 2026) compiles executable state into proxy videos. RIGen instead connects gameplay state to generation through player actions and local interaction guidance, without constructing explicit spatial trajectories. The PixVerse Game Engine technical blog (PixVerse Research, 2026) describes a separation of game mechanics, agent orchestration, and generative expression. We provide a concrete implementation for continuous player control, connecting interaction prediction, executable state updates, and video feedback, with corresponding generator training and controlled ablations.

3 METHOD

3.1 OVERVIEW

RIGen couples three layers to align player actions, persistent state, and generated audiovisual outcomes (Figure 3). The *State and Rule Engine* maintains persistent state, applies transition rules, and specifies strategic constraints (Section 3.2). The *Interaction Agent* reasons about action consequences and plans actor behavior (Section 3.3). The *Audiovisual Generator* models local dynamics under player controls and generation guidance (Section 3.4).

At each chunk boundary, the agent uses recent video, player controls, and persistent state to predict interaction events and plan actor behavior. The rule engine evaluates these events, and the updated state and planned behavior jointly guide the next audiovisual chunk. After generation, the agent checks which events occurred, and the rule engine uses this evidence to determine the state for the next decision.

3.2 PERSISTENT STATE AND RULE-BASED UPDATES

We represent gameplay as an event-driven state machine with numerical variables, following standard state-based game AI (Millington & Funge, 2009). The engine maintains player and enemy health and life/death state, player ammunition, and opponent strategy modes, together with configurable rule parameters such as the player's per-shot damage and transition conditions. Writing s for the complete game state and e for an interaction event record, the engine maps s to its successor s' through

$$s' = \mathrm{Rules}(s, e). \tag{1}$$

For example, firing requires a living player with available ammunition and consumes a round regardless of whether the shot hits. A hit reduces the target's health by the configured damage, for either the player or an enemy. Reaching zero health triggers death. If an opponent survives with low health, a configured rule can switch its strategy from engagement to retreat. The updated strategy guides the Interaction Agent's next decision, such as choosing a retreat direction suited to the current scene (Section 3.3).

3.3 EVENT REASONING, BEHAVIOR PLANNING, AND VISUAL FEEDBACK

Before generating chunk o_k, we perform *reconciliation* through verification and state correction. The agent verifies o_{k-1} using sampled verification frames $\mathcal{I}_{k-1}$, player controls u_{k-1}, and local guidance p_{k-1}. The rule engine then corrects the planned state s_k^{plan} using feedback f_{k-1}:

$$\begin{aligned} f_{k-1} &= \mathrm{Verify}(\mathcal{I}_{k-1}, u_{k-1}, p_{k-1}), \\ s_k &= \mathrm{Correct}(s_k^{\mathrm{plan}}, f_{k-1}). \end{aligned} \tag{2}$$

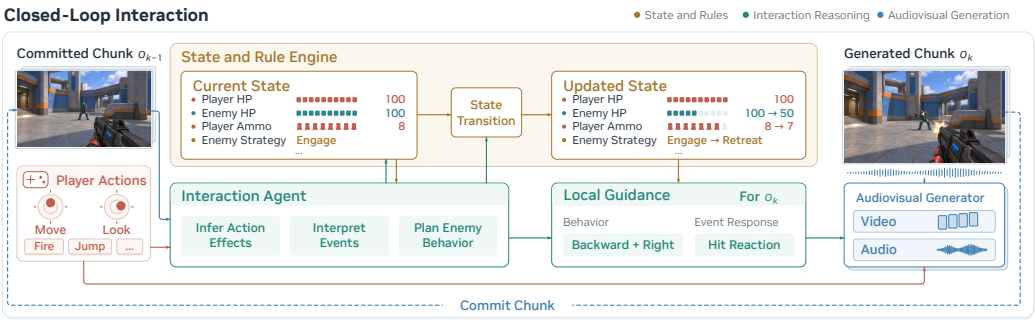

Figure 3: **Closed-loop interaction in RIGen.** The interaction agent uses committed observations, player actions, and persistent state to infer events and behavior. The updated state and interaction proposals produce local guidance for the next audiovisual chunk, whose observations inform subsequent decisions.

Using this updated state, feedback, and current player controls u_k, the agent proposes events e_k and behavior b_k:

$$(e_k, b_k) = \text{Agent}(o_{k-1}, s_k, u_k, f_{k-1}). \tag{3}$$

The rule engine computes a provisional successor s_{k+1}^{plan} and local interaction guidance p_k without overwriting s_k:

$$\begin{aligned} s_{k+1}^{\text{plan}} &= \text{Rules}(s_k, e_k), \\ p_k &= \text{Prompt}(s_{k+1}^{\text{plan}}, e_k, b_k). \end{aligned} \tag{4}$$

The generator realizes this guidance as the next audiovisual chunk o_k (Section 3.4). At the next boundary, verification of o_k reconciles s_{k+1}^{plan} with committed visual outcomes to obtain s_{k+1}. These observations take precedence over conflicting predictions, and the corrected state conditions the next decision. Implementation scope and verification details are given in the appendix.

3.4 Audiovisual Generation

The global prompt g describes the scene and character identity shared across chunks. Local guidance p_k specifies the current chunk's actor behavior and observable interaction outcomes, derived from the agent's proposals and rule-based state updates (Section 3.3). Together, they form text condition c_k. Given audiovisual context $\mathcal{H}_k$ and player controls u_k, the generator produces the next chunk:

$$c_k = (g, p_k), \qquad o_k \sim G_\theta(\mathcal{H}_k, u_k, c_k). \tag{5}$$

The context $\mathcal{H}_k$ contains a first-frame anchor, a character reference, and recent generated chunks with their controls and text conditions. The generator denoises the target under these conditions, and the resulting o_k provides observations for the next interaction decision.

We instantiate G_θ with the LTX-2.3 (HaCohen et al., 2026; Lightricks, 2026) audiovisual backbone, using coupled video and audio transformer streams (Figure 4). Continuous movement/camera controls modulate video features through FiLM (Perez et al., 2018), while button inputs enter through causal cross-attention. Audiovisual attention connects the two streams and is causal across chunks and joint within each chunk. The anchor and reference provide visual context without text or action updates.

Chunk-wise prompt routing. In both video and audio streams, each chunk attends only to the global prompt and its corresponding local guidance through text cross-attention (Figure 4). Historical chunks retain their original text conditions, while the target receives the current guidance. The same routing is used in bidirectional and causal training.

Training and continued generation. Following the bidirectional-to-autoregressive adaptation used in CausVid (Yin et al., 2025), we first train the model bidirectionally to follow player controls and chunk-level guidance, then adapt it to autoregressive next-chunk prediction conditioned

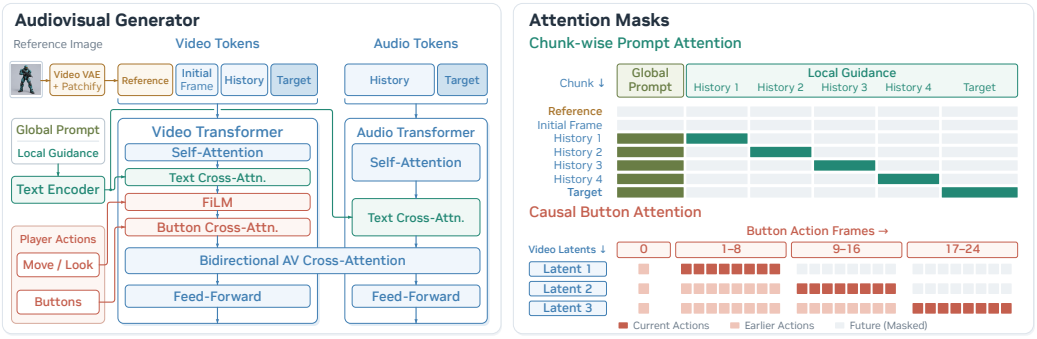

Figure 4: **Audiovisual generation with player actions and interaction guidance.** *Left:* Global prompts and local guidance condition both Transformers, while player actions condition the video stream. *Right:* Each chunk attends to the global prompt and its own local guidance. Each video latent aligns with eight action frames and attends to current and earlier actions, with future actions masked.

on clean audiovisual history. Sliding-window training retains a window of recent chunks alongside the first-frame anchor and character reference, keeping $\mathcal{H}_k$ bounded during continued generation. Training objectives and implementation details are provided in the appendix.

4 Xonotic Interaction Dataset

We modify the open-source engine of Xonotic (Xonotic Team, 2023) to construct an audiovisual gameplay dataset with controlled capture and synchronized annotations. Three complementary routes target different aspects of gameplay. *Basic Dynamics* captures how player controls affect local visual dynamics. *Interaction* covers the relationships between player actions, opponent behavior, and combat outcomes in controlled encounters, including longer sequences of successive encounters. *Arena* captures sustained combat between independently controlled bots to support continued interactive play. The collection contains 95,731 source recordings and 176.8 hours of encoded video (Table 2).

Each recording provides synchronized first-person video, game audio, player controls, and engine-grounded camera poses, together with scene and character descriptions. The interaction subsets pair chunk-level descriptions of opponent behavior, visibility, and combat outcomes with engine state and event annotations. Actor reference images are bound to the corresponding identities where used. Table 2 compares our dataset with existing gameplay datasets. Our collection jointly targets local dynamics, controlled combat outcomes, and sustained interaction. The appendix details the capture routes, alignment, filtering, and test set construction.

5 Post-training for RIGen-Arena

RIGen-Arena is an interactive one-on-one FPS prototype in which the player fights an opponent, with either side respawning after defeat to continue the match (Figure 5).

We initialize post-training for RIGen-Arena from the bidirectional model trained on the Interaction subset. We first fine-tune it bidirectionally on the Arena subset to adapt to sustained combat. We then train the adapted model to generate chunks autoregressively using a sliding window of recent history together with the first-frame anchor and character reference (Section 3.4).

We adopt the two-stage distillation recipe of Causal-rCM (Zheng et al., 2026a). Teacher-forcing consistency distillation first produces a few-step generator. Distribution matching then trains this generator on its own autoregressive rollouts.

We additionally train a video refiner to recover high-frequency details in dynamic gameplay. Training pairs spatially degraded latents with the original encodings of the same arena clips. We first train the refiner bidirectionally with paired flow matching (Lipman et al., 2023), then fine-tune it for

Table 2: **Dataset composition and annotation coverage.** (a) Collection by route. (b) Annotation coverage across datasets. GT denotes recorded positions and orientations. Mixed combines GT and estimated poses. Local text describes temporally localized behavior, state labels record gameplay variables, and controlled outcomes cover misses, nonlethal hits, and kills.

(a) Our collection

Route	Coverage	Recordings	Hours
Basic Dynamics	Controls and local dynamics	50,140	69.64
Interaction	Controlled interactions	42,000	47.25
Interaction-Long	Actor sequences and outcomes	1,791	29.87
Arena	Sustained one-on-one combat	1,800	30.03

(b) Comparison with related datasets

Dataset	Audio	Action Labels	Pose Labels	Local Text	State Labels	Controlled Outcomes
CS:GO (Pearce & Zhu, 2022)	–	Inputs	GT	–	✓	–
Gaming-500-hours (Markov AI, 2026)	–	Inputs	–	–	–	–
GF-Minecraft (Yu et al., 2025)	–	Inputs	–	–	–	–
PLAICraft (He et al., 2025b)	✓	Inputs	–	–	–	–
CrossFPS (Tong et al., 2026)	–	Inputs	–	–	–	–
SANA-WM (Zhu et al., 2026a)	–	–	Mixed	–	–	–
ABot-World-0 (Jiang et al., 2026)	–	Inputs	Mixed	–	–	–
SolarWM (Huang et al., 2026a)	–	–	Mixed	–	–	–
Incantation (Zhu et al., 2026b)	–	Animation IDs	–	✓	–	–
WildWorld (Li et al., 2026)	–	Animation IDs	GT	✓	✓	–
Ours	✓	Inputs	GT	✓	✓	✓

causal, chunk-wise refinement with a bounded history window. At inference, it refines each generated chunk without access to future chunks. We integrate the generator and refiner with the state and rule engine and interaction agent to form RIGen-Arena. Training objectives, sampling schedules, and implementation details are provided in the appendix.

6 Experiments

This section evaluates motion adherence, combat outcomes, and visual quality through quantitative comparisons and ablations. RIGen-Arena (Figure 5) and the qualitative results (Figures 6 and 7) illustrate state-dependent interactions, continued gameplay, and the world's effects on the player.

6.1 Quantitative Evaluation

Setup. We compare RIGen's checkpoint trained on Interaction-Long with sliding-window autoregressive training against SCOPE (Tong et al., 2026) and LingBot-World 2.0 (Gao et al., 2026) on 100 Xonotic combat scenarios. Test maps and enemy appearances are excluded from all task-specific training. Enemy motion follows source annotations. Both baselines receive prescribed combat outcomes as text guidance. InternVL3-8B (Zhu et al., 2025) is our Interaction Agent for event reasoning and visual state correction.

Ablation Variants. *Motion guidance only* retains player inputs and source motion, without interaction guidance or an online agent. *Semantic prompting* provides state-update rules as text for the agent to interpret and iteratively update its textual state, without an executable rule engine or independent verification. *Post-hoc event updates* moves interaction reasoning and rule-based state updates from before generation to after it, so the resulting guidance conditions the next chunk rather than the current one. *Without reconciliation* disables visual correction of planned state. *RIGen (w/ Reference guidance)* freezes event guidance derived from source hit annotations and health/damage rules, without an online agent.



Figure 5: **A gameplay session in RIGen-Arena.** Opponent defeat, respawn and seeking, and player defeat. The HUD is rendered from rule-engine state and overlaid on the video, not generated by the video model.

Table 3: Model comparisons and ablations on 100 Xonotic scenarios with 50 survival and 50 death targets. RIGen (w/ Reference guidance) is a privileged offline control.

Method / Variant	Visual Quality & Dynamics					Interaction Quality			
	MS ↑	AQ ↑	SC ↑	BC ↑	DD	EB ↑	FSA ↑	FSA-S ↑	FSA-D ↑
SCOPE (Tong et al., 2026)	94.82	36.20	61.15	81.71	87.00	35.71	4.00	6.00	2.00
LingBot-World 2.0 (Gao et al., 2026)	96.50	**53.86**	79.99	92.82	**100.00**	55.32	29.00	58.00	0.00
RIGen (Ours)	**99.18**	50.93	**90.97**	**94.17**	89.00	**78.91**	**81.00**	**90.00**	**72.00**
Motion guidance only	99.17	50.90	90.85	94.11	88.00	76.59	39.00	78.00	0.00
Semantic prompting	99.13	50.84	90.92	94.02	88.00	75.49	35.00	70.00	0.00
Post-hoc event updates	99.15	50.67	90.84	94.01	88.00	77.74	64.00	86.00	42.00
w/o reconciliation	99.18	50.88	90.96	94.15	89.00	78.02	76.00	88.00	64.00
RIGen (w/ Reference guidance)	99.18	50.96	90.99	94.14	89.00	81.99	89.00	90.00	88.00

Metrics. Qwen3.8-27B (Qwen Team, 2026) evaluates two interaction metrics offline. *Enemy Behavior* (EB) measures adherence to prescribed enemy motion within specified time windows. *Final-state Agreement* (FSA) measures agreement between the final visible enemy state and its survival/death target. FSA-S is the percentage of the 50 survival-target cases in which the enemy is visibly alive at the end. FSA-D is the percentage of the 50 death-target cases in which the enemy is visibly dead at the end. Overall FSA averages these two rates. Incorrect or unjudgeable outcomes count as failures, so FSA can fall below 50%. We also report Motion Smoothness (MS), Aesthetic Quality (AQ), Subject Consistency (SC), Background Consistency (BC), and Dynamic Degree (DD), following VBench (Huang et al., 2024). Scores are on a 0–100 scale. Higher is better except for DD, which measures motion prevalence rather than quality.

Results. Table 3 shows that RIGen achieves 81% FSA, compared with 29% for LingBot-World 2.0 and 4% for SCOPE, and higher EB than both baselines. With the generator fixed, FSA falls to 39% with motion-only guidance, 35% with semantic prompting, 64% with post-hoc updates, and 76% without reconciliation. These results support predicting interaction consequences before generation, executing state updates, and reconciling them with visual feedback. Visual metrics remain nearly unchanged across ablations. Lower AQ than LingBot-World 2.0 may reflect our smaller training dataset. The appendix provides further details and analysis.

6.2 Qualitative Evaluation

Combat responses. Figure 6 compares visible combat responses from the same initial image and attack requests. In this example, RIGen produces a hit reaction followed by a fall and a sustained downed state, while the baseline opponents remain active under repeated fire despite receiving text guidance specifying the correct combat outcomes.

State-dependent interactions. Figure 7 varies initial enemy health, player ammunition, and state-dependent opponent strategy from a shared starting scene under the same player input requests. The low-health opponent falls, depleted ammunition suppresses the third shot, and an injured opponent switches from hold to retreat, moving left and crouching at cover. These examples illustrate how explicit state and strategy affect visible gameplay.

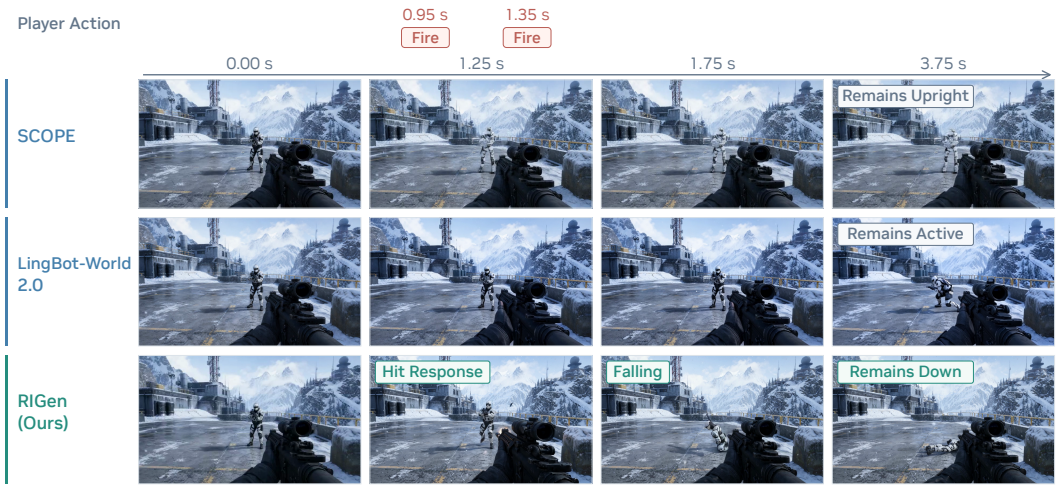

Figure 6: **Qualitative combat responses.** Red markers denote fire commands. Despite receiving the prescribed outcomes as text guidance, SCOPE and LingBot-World 2.0 keep opponents active under repeated fire. RIGen produces a hit reaction, a fall, and a sustained downed state.



Figure 7: **State-dependent interaction in RIGen.** The reference has enemy HP 200, eight rounds, and a hold strategy. All branches use 50 damage per shot. Red markers denote fire. Lower health, depleted ammunition, and injury alter survival, firing, and retreat behavior, respectively.

7 Conclusion

We presented RIGen, a framework for reciprocal interaction in generative gameplay. Player actions change persistent world state, while world rules and actor decisions affect the player. Quantitative comparisons demonstrate more reliable combat outcomes, and qualitative results illustrate state-dependent interactions and autonomous enemy behavior.

Limitations and future work. RIGen is not yet real-time, supports only single-opponent FPS gameplay, and relies on VLM perception and reasoning for interaction decisions and state verification. Future work will optimize generation and agent inference, extend to multiple opponents and broader action spaces and game genres, and improve visual grounding and reasoning reliability.

AI USE STATEMENT

We used generative AI tools to assist with drafting and polishing the manuscript and with research execution, specifically code implementation, debugging, experiment scripting, and development of the dataset construction pipelines. The authors reviewed and revised the AI-assisted text and checked the AI-assisted code and experimental outputs for correctness. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

We construct our dataset through scripted and bot-controlled gameplay in Xonotic, without collecting human-player data or conducting human-subject experiments. The official Xonotic distribution is licensed under GPL-3.0-or-later, with its DarkPlaces engine under GPL-2.0-or-later.[1] Code and assets remain subject to their applicable upstream licenses. Our release will document the licenses for the dataset and accompanying code and preserve required attribution and notices. We follow the applicable usage terms of pretrained models. The proposed method is intended for research purposes. Deployment should account for potential misuse of generated content, including misleading or deceptive media.

REPRODUCIBILITY STATEMENT

The main text describes the proposed method and experimental setup. Additional implementation details, hyperparameter settings, and evaluation protocols are provided in the appendix to support reproducibility. The datasets and baseline methods used in our experiments are identified and referenced in the paper.

REFERENCES

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. URL https://arxiv.org/abs/2603.00825.

AlayaWorld Team, Kaipeng Zhang, Chuanhao Li, Yifan Zhan, Yongtao Ge, Yuanyang Yin, Jiaming Tan, Kang He, Liaoyuan Fan, Mingliang Zhai, et al. AlayaWorld: Interactive Long-Horizon World Modeling - Full Technical Report (v1.1). *arXiv preprint arXiv:2608.13492*, 2026. URL https://arxiv.org/abs/2608.13492. AlayaWorld v1.1.

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*, volume 37, pp. 58757–58791, 2024. doi: 10.52202/079017-1873. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6bdde0373d53d4a501249547084bed43-Abstract-Conference.html.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative Interactive Environments. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623. PMLR, 2024. URL https://proceedings.mlr.press/v235/bruce24a.html.

Ziqi Cai, Siqi Yang, Yimu Wang, Zixian Gao, Yunheng Liu, Shuchen Weng, Erwin Wu, Kaipeng Zhang, and Boxin Shi. MASS: Multiplayer World Models with Authoritative Shared State. *arXiv preprint arXiv:2608.06257*, 2026. URL https://arxiv.org/abs/2608.06257.

Jin Cao, Zian Meng, and Kaipeng Zhang. ShadowDancer: Teaching Video World Models Any Action by Learning Unified Dynamics Representations from a Video and Its Shadow. *arXiv preprint arXiv:2607.28362*, 2026. URL https://arxiv.org/abs/2607.28362.

[1] https://github.com/xonotic/xonotic/blob/master/COPYING

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*, pp. 37546–37593, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5d2e24df9cfaad3189833b819c40b392-Abstract-Conference.html.

Jingye Chen, Yuzhong Zhao, Yupan Huang, Lei Cui, Li Dong, Tengchao Lv, Qifeng Chen, and Furu Wei. Model as a Game: On Numerical and Spatial Consistency for Generative Games. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 1958–1967, 2025. URL https://openaccess.thecvf.com/content/ICCV2025W/HiGen/html/Chen_Model_as_a_Game_On_Numerical_and_Spatial_Consistency_for_ICCVW_2025_paper.html.

Yiwen Chen, Guosheng Lin, and Chi Zhang. Code World Model: Coding Agent as World Brain. *arXiv preprint arXiv:2608.25927*, 2026. URL https://arxiv.org/abs/2608.25927.

Junhao Cheng, Yuying Ge, Yixiao Ge, Jing Liao, and Ying Shan. AnimeGamer: Infinite Anime Life Simulation with Next Game State Prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10875–10885, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Cheng_AnimeGamer_Infinite_Anime_Life_Simulation_with_Next_Game_State_Prediction_ICCV_2025_paper.html.

Decart, Julian Quevedo, Quinn McIntyre, Spruce Campbell, Xinlei Chen, and Robert Wachen. Oasis: A Universe in a Transformer. Technical report, 2024. URL https://oasis-model.github.io/.

Zelin Gao, Qiuyu Wang, Jiapeng Zhu, Jingye Chen, Zichen Liu, Qingyan Bai, Jiahao Wang, Yufeng Yuan, Hanlin Wang, Yichong Lu, et al. Infinite Worlds with Versatile Interactions. *arXiv preprint arXiv:2607.07534*, 2026. URL https://arxiv.org/abs/2607.07534. LingBot-World 2.0 (LingBot-World-Infinity).

Google DeepMind. Gemini 3.1 Flash-Lite. Official model documentation, 2026. URL https://ai.google.dev/gemini-api/docs/models/gemini-3.1-flash-lite.

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. URL https://arxiv.org/abs/2504.08388.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, et al. LTX-2: Efficient Joint Audio-Visual Foundation Model. *arXiv preprint arXiv:2601.03233*, 2026. URL https://arxiv.org/abs/2601.03233.

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

Yingchen He, Christian D. Weilbach, Martyna E. Wojciechowska, Yuxuan Zhang, and Frank Wood. PLAICraft: Large-Scale Time-Aligned Vision-Speech-Action Dataset for Embodied AI. *arXiv preprint arXiv:2505.12707*, 2025b. URL https://arxiv.org/abs/2505.12707.

Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. *arXiv preprint arXiv:2207.12598*, 2022. URL https://arxiv.org/abs/2207.12598.

Anthony Hu, Václav Volhejn, Adrien Ramanana Rahary, Chris Mulder, Aditya Makkar, Alyx Liao, Amélie Royer, Manu Orsini, Adam Jelley, Eloi Alonso, et al. Multiplayer Interactive World Models with Representation Autoencoders. *arXiv preprint arXiv:2607.05352*, 2026. URL https://arxiv.org/abs/2607.05352.

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

Junchao Huang, Guian Fang, Shengju Qian, Xianghao Kong, Zhuoran Zhao, Wei Huang, Yihua Du, Zixin Zhang, Justin Cui, Yuchao Gu, et al. SolarWM: Open Data and Scalable Training for Long-Horizon Video World Models. *arXiv preprint arXiv:2609.02886*, 2026a. URL https://arxiv.org/abs/2609.02886.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self Forcing: Bridging the Train-Test Gap in Autoregressive Video Diffusion. In *Advances in Neural Information Processing Systems*, volume 38, pp. 167283–167308, 2025. doi: 10.52202/085713-5576. URL https://papers.neurips.cc/paper_files/paper/2025/hash/f4823f831af67a3ef15e41a85434422a-Abstract-Conference.html.

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

Zheng-Hui Huang, Zhixiang Wang, Jiaming Tan, Ruihan Yu, Yidan Zhang, Bo Zheng, Yu-Lun Liu, Yung-Yu Chuang, and Kaipeng Zhang. Generative World Renderer. *arXiv preprint arXiv:2604.02329*, 2026c. URL https://arxiv.org/abs/2604.02329.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. VBench: Comprehensive Benchmark Suite for Video Generative Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Huang_VBench_Comprehensive_Benchmark_Suite_for_Video_Generative_Models_CVPR_2024_paper.html.

Fan Jiang, Zhaoxu Sun, Mengchao Wang, Ziyu Zhu, Chiyu Wang, Yunpeng Zhang, Wenlin Liu, Yun Wang, Xue Zheng, Rui Sun, et al. ABot-World-0: Infinite Interactive World Rollout on a Single Desktop GPU. *arXiv preprint arXiv:2607.19191*, 2026. URL https://arxiv.org/abs/2607.19191.

Anssi Kanervisto, Dave Bignell, Linda Yilin Wen, Martin Grayson, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Tabish Rashid, Tim Pearce, Yuhan Cao, et al. World and Human Action Models towards gameplay ideation. *Nature*, 638(8051):656–663, 2025. doi: 10.1038/s41586-025-08600-3. URL https://www.nature.com/articles/s41586-025-08600-3. WHAM / Muse.

Seung Wook Kim, Yuhao Zhou, Jonah Philion, Antonio Torralba, and Sanja Fidler. Learning to Simulate Dynamic Environments With GameGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1231–1240, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Kim_Learning_to_Simulate_Dynamic_Environments_With_GameGAN_CVPR_2020_paper.html.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. HunyuanVideo: A Systematic Framework For Large Video Generative Models. *arXiv preprint arXiv:2412.03603*, 2024. URL https://arxiv.org/abs/2412.03603.

Jialu Li, Yuanzhen Li, Neal Wadhwa, Yael Pritch, David E. Jacobs, Michael Rubinstein, Mohit Bansal, and Nataniel Ruiz. Unbounded: A Generative Infinite Game of Character Life Simulation. In *International Conference on Learning Representations*, pp. 87465–87491, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/da2411768acb844b255bb6770e5a71c7-Abstract-Conference.html.

Jiaqi Li, Junshu Tang, Zhiyong Xu, Longhuang Wu, Yuan Zhou, Shuai Shao, Tianbao Yu, Zhiguo Cao, and Qinglin Lu. Hunyuan-GameCraft: High-dynamic Interactive Game Video Generation

with Hybrid History Condition. *arXiv preprint arXiv:2506.17201*, 2025b. URL https://arxiv.org/abs/2506.17201.

Zhen Li, Zian Meng, Shuwei Shi, Wenshuo Peng, Yuwei Wu, Bo Zheng, Chuanhao Li, and Kaipeng Zhang. WildWorld: A Large-Scale Dataset for Dynamic World Modeling with Actions and Explicit State toward Generative ARPG. *arXiv preprint arXiv:2603.23497*, 2026. URL https://arxiv.org/abs/2603.23497.

Long Lian, Baifeng Shi, Adam Yala, Trevor Darrell, and Boyi Li. LLM-grounded Video Diffusion Models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/d9f8b5abc8e0926539ecbb492af7b2f1-Abstract-Conference.html.

Lightricks. LTX-2.3. Official model release, 2026. URL https://huggingface.co/Lightricks/LTX-2.3. 22B dev backbone, spatial upscaler x2 v1.1, and distilled LoRA v1.1.

Guixu Lin, Zheng-Hui Huang, Siqi Yang, Ming-Hsuan Yang, Kaipeng Zhang, and Zhixiang Wang. Generative World Renderer at the Speed of Play. *arXiv preprint arXiv:2607.18703*, 2026a. URL https://arxiv.org/abs/2607.18703.

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

Zijun Lin, Zeqing Wang, Cheston Tan, Bihan Wen, and Yeying Jin. StatePlay: State-Aware Game World Models for Mechanics-Consistent Generation. *arXiv preprint arXiv:2607.26754*, 2026b. URL https://arxiv.org/abs/2607.26754.

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*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Cheng Lu and Yang Song. Simplifying, Stabilizing and Scaling Continuous-time Consistency Models. In *International Conference on Learning Representations*, pp. 50611–50649, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/7e9c2053258b1bdd32ff2654802cd594-Abstract-Conference.html.

Markov AI. Gaming-500-hours: Gaming Dataset (gaming-1). Hugging Face dataset, 2026. URL https://huggingface.co/datasets/markov-ai/gaming-500-hours.

Willi Menapace, Stephane Lathuiliere, Sergey Tulyakov, Aliaksandr Siarohin, and Elisa Ricci. Playable Video Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10061–10070, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Menapace_Playable_Video_Generation_CVPR_2021_paper.html.

Willi Menapace, Stéphane Lathuilière, Aliaksandr Siarohin, Christian Theobalt, Sergey Tulyakov, Vladislav Golyanik, and Elisa Ricci. Playable Environments: Video Manipulation in Space and Time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3584–3593, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Menapace_Playable_Environments_Video_Manipulation_in_Space_and_Time_CVPR_2022_paper.html.

Zian Meng, Zhen Li, Chuanhao Li, Qiang Li, and Kaipeng Zhang. Marionette: Predicting World States, Rendering Geometry, Painting Appearance. *arXiv preprint arXiv:2608.14530*, 2026. URL https://arxiv.org/abs/2608.14530.

Ian Millington and John Funge. *Artificial Intelligence for Games*. Morgan Kaufmann, 2 edition, 2009. URL https://shop.elsevier.com/books/artificial-intelligence-for-games/millington/978-0-12-374731-0.

Liangyang Ouyang, Ruicong Liu, Xuangeng Chu, Kaipeng Zhang, and Yoichi Sato. HelloWorld: Enabling Socially Interactive Characters in Video World Models. *arXiv preprint arXiv:2608.05070*, 2026. URL https://arxiv.org/abs/2608.05070.

Tim Pearce and Jun Zhu. Counter-Strike Deathmatch with Large-Scale Behavioural Cloning. In *2022 IEEE Conference on Games (CoG)*, pp. 104–111. IEEE, 2022. doi: 10.1109/CoG51982.2022.9893617. URL https://doi.org/10.1109/CoG51982.2022.9893617.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual Reasoning with a General Conditioning Layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 3942–3951, 2018. doi: 10.1609/aaai.v32i1.11671. URL https://ojs.aaai.org/index.php/AAAI/article/view/11671.

PixVerse Research. PixVerse Game Engine Deep Dive: Real-Time Interactive Gaming. Technical blog, 2026. URL https://pixverse.ai/en/blog/pixverse-game-engine-deep-dive.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie Gen: A Cast of Media Foundation Models. *arXiv preprint arXiv:2410.13720*, 2024. URL https://arxiv.org/abs/2410.13720.

Qwen Team. Qwen3.8-27B. Official model release, 2026. URL https://huggingface.co/Qwen/Qwen3.8-27B.

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. URL https://arxiv.org/abs/2601.20540. LingBot-World.

Ludan Ruan, Yiyang Ma, Huan Yang, Huiguo He, Bei Liu, Jianlong Fu, Nicholas Jing Yuan, Qin Jin, and Baining Guo. MM-Diffusion: Learning Multi-Modal Diffusion Models for Joint Audio and Video Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10219–10228, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Ruan_MM-Diffusion_Learning_Multi-Modal_Diffusion_Models_for_Joint_Audio_and_Video_CVPR_2023_paper.html.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency Models. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

Yaofeng Su, Yuming Li, Zeyue Xue, Jie Huang, Siming Fu, Haoran Li, Haoyang Huang, and Nan Duan. OmniForcing: Unleashing Real-Time Joint Audio-Visual Generation. In *European Conference on Computer Vision*, pp. 566–584. Springer, 2026. doi: 10.1007/978-3-032-37271-0_31. URL https://link.springer.com/chapter/10.1007/978-3-032-37271-0_31.

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. URL https://arxiv.org/abs/2512.14614.

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

Zizhao Tong, Yeying Jin, Hongfeng Lai, Zeqing Wang, Zhaohu Xing, Kexu Cheng, Haoran Xu, Zhao Pu, Shangwen Zhu, Ruili Feng, et al. SCOPE: Simulating Cross-game Operations in Playable Environments for FPS World Models. *arXiv preprint arXiv:2605.23345*, 2026. URL https://arxiv.org/abs/2605.23345.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion Models Are Real-Time Game Engines. In *International Conference on Learning Representations*, pp. 73754–73776, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/b71ecea210f7159f31e46631fe5c838f-Abstract-Conference.html.

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. URL https://arxiv.org/abs/2503.20314.

Zeqing Wang, Danze Chen, Zhaohu Xing, Zizhao Tong, Yinhan Zhang, Xingyi Yang, and Yeying Jin. ReactiveGWM: Steering NPC in Reactive Game World Models. *arXiv preprint arXiv:2605.15256*, 2026a. URL https://arxiv.org/abs/2605.15256.

Zile Wang, Zexiang Liu, Jiaxing Li, Kaichen Huang, Baixin Xu, Fei Kang, Mengyin An, Peiyu Wang, Biao Jiang, Yichen Wei, et al. Matrix-Game 3.0: Real-Time and Streaming Interactive World Model with Long-Horizon Memory. *arXiv preprint arXiv:2604.08995*, 2026b. URL https://arxiv.org/abs/2604.08995.

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. URL https://arxiv.org/abs/2602.02393.

Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. WorldMem: Long-term Consistent World Simulation with Memory. In *Advances in Neural Information Processing Systems*, volume 38, pp. 49632–49652, 2025. doi: 10.52202/085713-1659. URL https://papers.neurips.cc/paper_files/paper/2025/hash/470629a47e2d65ce0606c40055df5d26-Abstract-Conference.html.

Xonotic Team. Xonotic 0.8.6 Release. Official release announcement, 2023. URL https://xonotic.org/posts/2023/xonotic-0-8-6-release/.

Deheng Ye, Fangyun Zhou, Jiacheng Lv, Jianqi Ma, Jun Zhang, Junyan Lv, Junyou Li, Minwen Deng, Mingyu Yang, Qiang Fu, et al. Yan: Foundational Interactive Video Generation. *arXiv preprint arXiv:2508.08601*, 2025. URL https://arxiv.org/abs/2508.08601.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved Distribution Matching Distillation for Fast Image Synthesis. In *Advances in Neural Information Processing Systems*, volume 37, pp. 47455–47487, 2024a. doi: 10.52202/079017-1505. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/54dcf25318f9de5a7a01f0a4125c541e-Abstract-Conference.html.

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*, pp. 6613–6623, 2024b. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yin_One-step_Diffusion_with_Distribution_Matching_Distillation_CVPR_2024_paper.html.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From Slow Bidirectional to Fast Autoregressive Video Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22963–22974, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yin_From_Slow_Bidirectional_to_Fast_Autoregressive_Video_Diffusion_Models_CVPR_2025_paper.html. CausVid.

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

Songchun Zhang, Yaowei Li, Junhao Zhuang, Weiyang Jin, Haoyu Wang, Xin Lu, Yilang Sun, Shiyi Zhang, Haoran Li, Xiaoxiao Ma, et al. EchoWM: Open and Enterable Omnimodal World Models. *arXiv preprint arXiv:2608.23189*, 2026. URL https://arxiv.org/abs/2608.23189.

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. URL https://arxiv.org/abs/2506.18701.

Kaiwen Zheng, Guande He, Min Zhao, Jintao Zhang, Huayu Chen, Jianfei Chen, Chen-Hsuan Lin, Ming-Yu Liu, Jun Zhu, and Qianli Ma. Causal-rCM: A Unified Teacher-Forcing and Self-Forcing Open Recipe for Autoregressive Diffusion Distillation in Streaming Video Generation and Interactive World Models. *arXiv preprint arXiv:2606.25473*, 2026a. URL https://arxiv.org/abs/2606.25473.

Kaiwen Zheng, Yuji Wang, Qianli Ma, Huayu Chen, Jintao Zhang, Yogesh Balaji, Jianfei Chen, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Large Scale Diffusion Distillation via Score-Regularized Continuous-Time Consistency. In *International Conference on Learning Representations*, pp. 2582–2603, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/0534abc9e6db91683d82186ef0d68202-Abstract-Conference.html.

Haoyi Zhu, Haozhe Liu, Yuyang Zhao, Tian Ye, Junsong Chen, Jincheng Yu, Tong He, Song Han, and Enze Xie. SANA-WM: Efficient Minute-Scale World Modeling with Hybrid Linear Diffusion Transformer. *arXiv preprint arXiv:2605.15178*, 2026a. URL https://arxiv.org/abs/2605.15178.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. InternVL3: Exploring Advanced Training and Test-Time Recipes for Open-Source Multimodal Models. *arXiv preprint arXiv:2504.10479*, 2025. URL https://arxiv.org/abs/2504.10479.

Shangwen Zhu, Qianyu Peng, Zhao Pu, Zhilei Shu, Xiangrui Ke, Zhaohu Xing, Zizhao Tong, Zeqing Wang, Xinyu Cui, Zian Zheng, et al. Incantation: Natural Language as the Action Interface for Multi-Entity Video World Models. *arXiv preprint arXiv:2605.18601*, 2026b. URL https://arxiv.org/abs/2605.18601.

APPENDIX CONTENTS

A Implementation Details **18**

A.1 Audiovisual Generator and Action Conditioning 18

A.2 Chunk-wise Prompt Routing and Guidance Construction 18

A.3 Training Objectives and Distillation 18

A.4 State Updates and Visual Verification 20

A.5 Design Rationale 21

B Dataset Construction **22**

B.1 Shared Capture and Alignment 22

B.2 Basic Dynamics 22

B.3 Interaction 22

B.4 Arena: Sustained Combat 23

B.5 Quality Control and Test Set Construction 23

C Evaluation Details **23**

C.1 Evaluation Protocol 23

C.2 Metrics 24

C.3 Ablations and Reference Comparison 24

C.4 Runtime 25

A Implementation Details

A.1 Audiovisual Generator and Action Conditioning

We build on the LTX (HaCohen et al., 2026) audiovisual architecture, using the LTX-2.3 22B dev backbone (Lightricks, 2026). Separate video and audio autoencoders produce latent sequences processed by coupled transformer streams. Each stream uses self-attention and text cross-attention, with cross-modal attention connecting video and audio. At 20 video frames per second, each generated chunk contains two video latent steps, corresponding to 16 new frames, and 20 audio latent steps, corresponding to 0.8 seconds. Generation is causal between chunks. Tokens within a chunk are denoised jointly.

The action interface provides four continuous movement/camera channels and supports fire, jump, and reload inputs. For each generated video latent step, the eight aligned raw-frame vectors are concatenated into a 32-dimensional continuous-action vector J_t and broadcast over spatial locations. After projection, per-layer networks produce feature scale and shift through FiLM (Perez et al., 2018)

$$h'_{\ell,t} = \big(1 + \gamma_\ell(J_t)\big) \odot h_{\ell,t} + \beta_\ell(J_t). \tag{6}$$

Buttons remain at the raw-frame rate and enter through an additional cross-attention residual. Each generated video latent attends to its eight aligned action frames and earlier retained actions, with future actions masked. Temporal rotary embeddings (Su et al., 2024) align these queries and keys. Continuous scale/shift outputs and button-attention output projections are initialized to zero. The action branches act on the video stream, while cross-modal attention connects the resulting dynamics to audio.

Writing z_j for the audiovisual latents of generated chunk j, the causal model uses a compact context

$$[\text{first-frame anchor} \mid z_{k-4:k-1} \mid T_k \mid r]. \tag{7}$$

Here r denotes the encoded character reference. At steady state, four previous chunks are clean context and only the target chunk T_k is denoised. Tokens in each generated chunk attend to tokens in that chunk and earlier chunks. Reference-image tokens attend only to one another and are readable directly by the target video chunk. Direct cross-modal attention excludes the reference. The anchor and reference receive no action or text updates. As the window advances, old history is removed and temporal positions are rebased locally. The same layout supports zero to three history chunks during startup. An optional cache reuses clean-context keys and values within a chunk's denoising steps. It is rebuilt when the window advances.

A.2 Chunk-wise Prompt Routing and Guidance Construction

The generator's prompt does not encode numerical health, hit counters, or internal generation identifiers. High-frequency player controls remain in the action tensors, while text specifies the actor behavior and consequences those controls should realize.

Each chunk has its own text condition $c_j = (g, p_j)$. The global description covers the environment, weapon, and character appearance. Local guidance describes the current actor behavior, visibility, and observable interaction outcomes. Each chunk's combined text condition is encoded independently, and the resulting context tokens are concatenated. The text attention mask routes both video and audio queries to the corresponding prompt, with padding excluded and anchor/reference queries receiving no text-attention update in the causal model. In bidirectional adaptation, the initial frame instead shares the first generated chunk's prompt. Historical chunks use their corresponding prompts, while the target directly attends only to its current prompt. Historical context remains available through audiovisual attention, which is causal across chunks during autoregressive training and generation.

At inference, a deterministic text renderer converts the controller's structured predictions and rule-updated state into local guidance.

A.3 Training Objectives and Distillation

Generator training optimizes the audiovisual backbone and action branches while freezing the text encoder and autoencoders. The Interaction Agent uses InternVL3-8B (Zhu et al., 2025) without

task-specific fine-tuning, while the rule engine and prompt compiler are programmed. The loader aligns video, audio, player controls, reference images, and chunk text conditions at shared temporal boundaries.

Training hyperparameters. We train on 16 NVIDIA GB300 GPUs with a global batch size of 16 and BF16 mixed precision. Table 4 lists the generator learning rates. Consistency distillation and distribution matching use AdamW with $\beta = (0, 0.999)$, weight decay 0.01, and 100 warmup steps. The learned score model uses a learning rate of 10^{-5}. Both refiner stages use 8-bit AdamW with a learning rate of 10^{-5}, $\beta = (0.9, 0.999)$, weight decay 0.01, and 100 warmup steps. The domain LoRA has rank 384 and scaling parameter $\alpha = 384$.

Table 4: Generator training learning rates.

Training stage	Backbone	Action branches
Basic Dynamics adaptation	10^{-6}	10^{-5}
Interaction adaptation	5×10^{-7}	10^{-7}
Arena bidirectional adaptation	5×10^{-7}	10^{-7}
Sliding-window autoregressive training	10^{-6}	10^{-5}
Consistency distillation	10^{-6}	10^{-6}
Distribution matching	10^{-5}	10^{-5}

Action and prompt adaptation. Basic Dynamics data teach control-conditioned local dynamics, while the Interaction data add chunk-local behavior and outcome guidance. Arena post-training starts from the Interaction-trained bidirectional model and adapts it to sustained combat before causal training and distillation. Bidirectional adaptation uses chunk-specific text conditions and a linear-path flow-matching objective (Lipman et al., 2023; Liu et al., 2023). For modality $m \in \{v, a\}$, a clean latent sequence z^m is perturbed as $z^m_\sigma = (1-\sigma)z^m + \sigma\epsilon^m$, with Gaussian noise ϵ^m. The model predicts the flow velocity $\epsilon^m - z^m$ and is trained by minimizing

$$\mathcal{L}_{\text{bi}} = \sum_{m \in \{v,a\}} \lambda_m \mathbb{E}\left[\| v^m_\theta(z_\sigma, \sigma; u, c) - (\epsilon^m - z^m) \|^2_{\Omega_m} \right], \tag{8}$$

where c denotes the sequence of chunk text conditions and the norm averages supervised tokens, excluding the conditioning video frame and ineligible audio samples. The text attention mask restricts each generated chunk to its own text condition, while audiovisual self- and cross-attention remain bidirectional at this stage.

Sliding-window autoregressive training. We initialize the causal model from the adapted checkpoint and train the layout in Equation 7. The anchor, reference, and available history chunks remain clean. Noise is applied only to the next chunk. For clean target latents z_k, Gaussian noise ϵ_k, and noise level $\sigma \in [0, 1]$, the noisy input and predicted clean estimate are

$$\begin{aligned} z_{k,\sigma} &= (1-\sigma)z_k + \sigma\epsilon_k, \\ \widehat{z}_k &= z_{k,\sigma} - \sigma v_\theta(z_{k,\sigma}, \sigma \mid \mathcal{H}_k, u_k, c_k). \end{aligned} \tag{9}$$

Here v_θ predicts flow velocity, and $\mathcal{H}_k$, u_k, and c_k follow Equation 5. Superscripts v and a denote video and audio. The target-chunk reconstruction objective is

$$\mathcal{L}_{\text{AR}} = \mathbb{E}[w(\sigma)\left(\text{MSE}(\widehat{z}^v_k, z^v_k) + \lambda_a \,\text{MSE}(\widehat{z}^a_k, z^a_k)\right)], \tag{10}$$

where $w(\sigma) = \exp[-2(\sigma - 0.5)^2]$ weights the noise level and $\lambda_a = 0.16$ weights the audio loss. Squared errors are averaged over target tokens only. This stage uses ground-truth history. Bidirectional and autoregressive adaptation share the chunk-wise prompt routing used at inference, but only autoregressive adaptation enforces temporal causality through audiovisual attention.

Consistency distillation and startup coverage. For few-step generation, the implementation follows the teacher-forcing consistency initialization in Causal-rCM (Zheng et al., 2026a). This recipe builds on Consistency Models (Song et al., 2023), sCM (Lu & Song, 2025), and rCM (Zheng et al.,

2026b). A frozen autoregressive teacher advances a noisy sample between nearby noise levels. At the advanced point, the current student produces a clean estimate without gradients, which serves as the consistency target for its prediction at the starting point. Video and audio both contribute to this loss. Checkpoint EMA is maintained separately.

Startup and continued generation use the same sliding-window context layout. The training recipe alternates an update with four history chunks and an update with zero, one, two, or three history chunks.

The final Arena recipe further distills text classifier-free guidance (Ho & Salimans, 2022) into the student. The teacher combines conditional and text-unconditional predictions while retaining player actions and visual context. The student uses a single conditional forward pass per denoising step.

Self-forcing distribution matching. After consistency distillation, we apply self-forcing distribution matching (Huang et al., 2025) as the second stage of the Causal-rCM (Zheng et al., 2026a) recipe. The objective follows DMD (Yin et al., 2024b) and DMD2 (Yin et al., 2024a), with bidirectional-teacher supervision of an autoregressive student as in CausVid (Yin et al., 2025). One student starts from the initial frame, generates four startup chunks, and continues for five additional chunks using the same reference, chunk-wise prompt routing, and sliding context. No ground-truth history is inserted into this rollout. Both the startup chunks and the later chunks receive distribution-matching supervision.

A frozen bidirectional score model and a learned score model for student samples evaluate the same perturbed rollout with aligned controls and semantic conditions. These score models receive no reference-image tokens, so their text conditions describe character appearance explicitly. The final Arena recipe applies text guidance to the frozen score model, while the learned score model and student remain conditional. Their normalized difference in denoised predictions defines a stop-gradient target. The generator then replays the recorded final denoising step of each chunk to optimize a masked video–audio regression loss against that target. Rollout generation runs without gradients, avoiding storage of its full computation graph.

Video refiner training. We construct temporally aligned training pairs from the same arena clip. Its original VAE encoding serves as the target. To construct the degraded input, we spatially downsample the RGB frames, encode them, and apply a frozen LTX-2.3 $2\times$ spatial upscaler v1.1 (Lightricks, 2026) to restore the target spatial size. We train a domain LoRA (Hu et al., 2022) on the refiner with truncated flow matching (Lipman et al., 2023) between the original target and a noise-perturbed upsampled latent. The refiner is first trained bidirectionally. Causal adaptation starts from this checkpoint, conditions on the first frame, character reference, and up to four clean history chunks, and applies loss only to the current chunk. Training uses ground-truth history, while deployment uses previously refined outputs. Both stages optimize only the domain LoRA, keeping the upsampler and backbone frozen. The official LTX-2.3 distilled LoRA v1.1 (Lightricks, 2026) is disabled for this training loss and enabled for few-step refinement at inference.

Inference settings. The distilled Arena generator uses four denoising steps with noise levels $(1, 0.9375, 0.8333, 0.625, 0)$. Refinement uses two denoising steps with noise levels $(0.725, 0.421875, 0)$, a classifier-free guidance scale of 1, and no spatiotemporal guidance.

A.4 State Updates and Visual Verification

Prediction and state commitment. At a chunk boundary, the controller retains the corrected state and uses predicted events to compute provisional outcomes and local guidance. Generation does not commit those predictions to the persistent ledger. After decoding, verification produces the feedback used to obtain the next retained state in Equation 2. The implementation recomputes this state from the saved pre-generation state and observed events, preventing predicted and verified damage from being counted twice. The first chunk uses the initial frame and configured state without a preceding verification step. The retained gameplay state is stored independently of the generator's sliding visual context. When earlier frames leave that context, their recorded interaction consequences remain available to subsequent rule updates and enter generation through the current guidance.

Observation selection and shot prediction. The online Arena implementation uses decoded RGB frames directly, without an observation refiner. Within each chunk, it selects the first frame of each uninterrupted firing interval and every fourth frame thereafter (5 Hz at 20 FPS). Before generation, the latest available frame is the preceding chunk's last frame, or the initial image at startup. For each selected firing time, a calibrated control-to-pixel mapping estimates the image displacement from the intervening controls. We translate the latest frame accordingly, take a 320×320 crop centered on the screen, and overlay a hollow green aiming ring. The Interaction Agent receives this projected crop and a separate enemy appearance reference, and predicts whether the enemy's body overlaps the marked aiming region.

Visual verification. After generation, we decode the corresponding actual firing frames, the chunk's final frame, and the frame three positions before it. Together with the pre-generation frame, these form the sampled observations $\mathcal{I}_{k-1}$. A firing-frame audit receives the selected firing frames in temporal order, with the same ring overlay, alongside the pre-generation and two late frames. Unlike prediction, verification uses full frames rather than center crops. The ring radius is 5% of the shorter image dimension and leaves its interior unobstructed. A separate state query places the final frame first and uses the pre-generation and earlier late frame only as temporal context. It classifies the enemy as alive, dead, or not visible in the final frame. Both queries receive an appearance reference for identity only. Full frames are padded to a 2:1 canvas and the reference to a square, without cropping their contents. Aim markers appear only in the VLM inputs. The requested outcome is the condition being checked, not visual evidence, and verification uses separate calls from prediction.

The rule engine specifies intended transitions, while committed visual outcomes ground the state used for subsequent interaction. When an observed outcome contradicts a prediction, the controller revises the affected state and derives subsequent guidance from that revision. Numerical attributes remain rule-maintained rather than directly estimated from images, and occlusion alone is not treated as death.

Action constraints and opponent behavior. Ammunition gates the executed fire commands and is consumed when a shot is permitted, independently of whether it hits. Before an enemy can attack, the Interaction Agent checks visual preconditions, including whether the enemy is visible and facing the player. When these conditions hold, attacks are sampled at a configured rate, and each attack hits with a preset probability. Successful hits subtract a fixed per-shot damage from player health. Reaching zero health disables player controls and adds the corresponding terminal outcome to guidance. For opponent behavior, the agent assesses movement feasibility from the current scene. The behavior policy combines this assessment with retained state and the configured strategy to select a movement description. Attack rules and strategy-transition conditions can be edited in the rule layer. Numerical state remains in the controller, while the generator receives observable behavior and outcome descriptions. World-side events thus combine visual assessment of the current scene with executable game rules that update player state and constrain subsequent controls.

A.5 Design Rationale

Reciprocal interaction concerns the causal effects of player actions on the world and of world events on the player. It does not require every game mechanic to be learned. Executable rules make damage, ammunition, and termination conditions explicit and configurable, while scene-conditioned behavior reasoning determines how actors act under those conditions. Their combination connects world-side decisions to persistent consequences for the player during continuous interaction.

Realizing these interactions also requires a generator that can express their consequences. Explicit state specifies intended outcomes, and text guidance communicates them, but the generator must learn to realize them as visible behavior. We therefore develop interaction data, train the conditioned generator, and connect event reasoning, executable updates, and visual feedback in an online loop. The contribution is the reciprocal generative gameplay enabled by this coordinated system, including its data and training procedure. Reconciliation supplies feedback within this loop by grounding subsequent decisions in committed observations.

B Dataset Construction

All routes share the capture and alignment pipeline below. Counts refer to accepted source recordings, excluding candidate jobs, prompt variants, and overlapping training windows.

B.1 Shared Capture and Alignment

Execution and recording. We specify the map, assets, initial conditions, actor policies, and player controls before capture. The engine records gameplay demos together with applied inputs, frame timing, state, and interaction events. Requested controls and realized events are retained separately, with damage labels taken from realized engine events. Offline replay renders the recorded trajectory and game audio under a fixed client configuration. Presentation settings, including HUD, crosshair, and effect suppression, are versioned per route. Renderer telemetry records camera position and orientation, transforms, and projection parameters, synchronized with simulation ticks.

Time and action interface. The capture pipelines use a 64 Hz simulation/control clock. Warm-up is removed using the recorded capture boundary before aligning model time zero. The model-facing stream uses 20 video frames per second, with four continuous movement/camera channels and fire, jump, and reload inputs under the shared action contract. Camera and movement values follow the recorded applied controls. Button resampling preserves short active runs that might otherwise disappear between sampled frames. In the long Interaction release, the action profiles explicitly remove 64 warm-up ticks and align row zero with the first video frame.

Native media differ across routes. Basic Dynamics stores 320 frames at 64 fps and 1280×720 resolution. Short and long Interaction sequences store 81 and 1,201 frames at 20 fps and 1280×720. Arena stores 1,201 frames at 20 fps and 1664×960. Representative files in all four subsets contain 48 kHz stereo game audio. The model loader resizes video and resamples audio according to the training configuration. The generator training configuration uses 832×480 video and 16 kHz stereo audio. The 81-frame layout comprises an initial frame and five 16-frame chunks. The 1,201-frame layout similarly supports 75 chunks.

Semantic labels and packaging. We use Gemini 3.1 Flash-Lite (Google DeepMind, 2026) (`gemini-3.1-flash-lite`) to separately annotate map style, weapon appearance and operation, and player and enemy appearance. Interaction releases add chunk-local motion, visibility, damage, and death descriptions derived from telemetry. Prompts distinguish no damage, nonlethal damage, a new death event, and an already-dead actor. Off-screen intervals are described using visibility records rather than invented visible motion. Player controls remain in action tensors. Enemy movement and jumping belong to semantic conditions. Where reference images are used, the manifest binds them to the appropriate actor and time interval.

B.2 Basic Dynamics

Basic Dynamics contains 50,140 five-second clips across 28 maps and nine weapons, including 39,539 without an enemy and 10,601 with a non-damaging enemy distractor. Movement, camera changes, firing, and local collision responses provide the principal supervision. The supplement emphasizes pure idle, idle-to-action, action-to-idle, interrupted action, smooth routes, and wall impacts.

Route validation and camera filtering remove unsuitable trajectories.

B.3 Interaction

Short interactions. The short Interaction subset contains 42,000 recordings across five maps, nine weapons, and 42 model–skin–color appearances. Capture varies distance, opponent movement, direction changes, stopping, jumping, visibility transitions, and firing. Firing times are sampled deterministically from the scenario seed. Weapon-specific schedules include isolated shots, repeated taps, and bounded bursts. Opponent motion and jump programs are specified separately from player controls.

Outcome labels follow the realized engine events, yielding 6,535 no-fire examples, 16,666 misses, 14,496 nonlethal-hit examples, and 4,303 kills. The visibility allocation is 37,800 continuously vis-

ible examples, 2,100 exit-and-hold examples, and 2,100 exit-and-return examples. The annotation identifies visibility transitions within each 0.8-second chunk. The underlying timestamped records also support constructing annotations over windows of arbitrary duration within each recording. This release describes view-frustum transitions, not geometric occlusion behind arbitrary cover. Crosshair and casing effects are disabled in its capture configuration.

Each recording has one 81-frame bidirectional view and one five-chunk autoregressive view.

Long interaction sequences. Interaction-Long contains 1,791 accepted recordings with blood-hit effects disabled during capture. Each episode contains four actor slots, arranged either as distinct actors or a controlled-return sequence, and zero, one, or two scheduled kills. The accepted collection contains 900 controlled-return and 891 distinct-actor episodes.

The maps are Bromine, Courtfun, Geoplanetary, and Stormkeep. The weapons are MachineGun, Rifle, and Shotgun. Scripted trajectories undergo collision and ground-support checks. Camera schedules use frozen calibration traces, with scan and aiming phases, rather than live feedback that tracks the current enemy.

The reference bank contains 42 appearances with five clean views each. Bindings expose the current actor reference, retain the outgoing reference through its despawn transition, and mask intervals without an actor. Each source episode provides fifteen 81-frame bidirectional windows. The autoregressive view keeps each full sequence and its 75 chunk prompts.

B.4 Arena: Sustained Combat

Arena contains 1,800 one-minute bot-versus-bot recordings in a compact arena, using MachineGun combat and a fixed opponent appearance. Independent bot controls produce ongoing movement and exchanges instead of prescribing a short outcome sequence. The capture configuration bounds locomotion and camera motion and removes crosshair and pickup clutter.

The release constructs the action stream from authoritative gameplay records. Chunk descriptions summarize opponent motion, attacks, visibility, nonlethal damage, kills, and player death from telemetry aligned with the rendered camera. Fully off-screen chunks suppress visible-motion claims. The manifest provides 75 prompts per episode and a fixed opponent reference binding. Arena trains RIGen-Arena.

B.5 Quality Control and Test Set Construction

Capture validation checks route geometry, camera behavior, and consistency between scheduled interactions and recorded events. Manifest validation checks media metadata, finite and aligned controls, complete prompt schedules, and reference bindings.

Test set construction. We reserve 100 Interaction recordings for quantitative evaluation. Their maps and enemy appearances are excluded from all task-specific training subsets, including Basic Dynamics, Interaction, Interaction-Long, and Arena, and are not used in subsequent adaptation or distillation.

C Evaluation Details

C.1 Evaluation Protocol

The 100 Xonotic scenarios comprise 50 survival targets and 50 death targets. All RIGen variants in Table 3 use the same Interaction-trained generator after sliding-window autoregressive training on Interaction-Long, without Arena post-training or distillation. SCOPE and LingBot-World 2.0 receive the prescribed combat outcomes directly as text guidance.

Enemy motion is supplied by source annotations in place of online behavior selection. The interaction agent handles hit, injury, and death reasoning and correction of predicted state. All variants use their own generated video history. Online interaction variants compute guidance from that history, while RIGen (w/ Reference guidance) uses frozen reference guidance.

Fixing enemy motion gives methods common behavior requirements and focuses the comparison on motion adherence and combat-outcome realization, separating behavior selection from its execution in generated video.

We combine controlled experiments, state-dependent branching examples, and continuous closed-loop gameplay. Controlled experiments compare the realization of specified interaction outcomes. Branch comparisons show how state and strategy affect generated behavior, while RIGen-Arena demonstrates the mutual effects of player actions and world events during continued play. Together, these complementary evaluations examine outcome realization, state dependence, and reciprocal interaction in the implemented system.

The external comparisons evaluate complete systems, including their data and training recipes. Learning to realize interaction outcomes is part of the proposed method. Specifying an outcome in text alone does not ensure that a generator can express it. We therefore assess the resulting system capability against external models and use fixed-generator ablations to examine the online interaction mechanisms separately.

Evaluating generated trajectories. Once a generated trajectory diverges from its source recording, source-engine hit and visibility labels cannot automatically serve as ground truth for the generated scene. Rule execution can be checked against configured transitions, while judging whether a generated attack hits requires independent evidence from the generated video itself. Autonomous behavior also admits multiple valid choices under the same state, calling for task-based criteria that accommodate alternative behaviors. Our benchmark uses prescribed motion and outcome targets with video-based assessment. Event-level visual correctness and autonomous decision quality require annotations or task criteria for the generated trajectories, rather than direct reuse of source-engine labels.

C.2 Metrics

Qwen3.8-27B (Qwen Team, 2026) judges complete generated videos offline.

Enemy Behavior. Enemy Behavior averages satisfaction of directional motion checks within predefined time windows, first within each video and then equally across the 100 scenarios. A check receives credit only when the judge reports it as satisfied and cites visual evidence within its time window. Lateral and depth motion are checked separately, relative to the environment rather than camera-induced image motion. For death-target cases, the prescribed lethal chunk and subsequent motion requirements are excluded for all methods using the same fixed windows.

Final-state Agreement. Final-state Agreement compares the enemy's visible state in the final 0.8 seconds with its preassigned survival or death target. The target is withheld from the judge. Unreadable outcomes and failures remain in the denominator and receive no agreement credit. Overall FSA uses all 100 cases. Survival-target and death-target success rates each use the corresponding 50 cases, so overall FSA is their mean. FSA tests whether the prescribed final combat outcome is actually realized. Locally plausible motion and high visual quality do not establish that an opponent survives or dies as required, making outcome agreement a distinct interaction criterion. EB measures motion adherence, while qualitative sequences illustrate intermediate combat responses.

Visual quality and dynamics. The five visual metrics use the official dimension-specific VBench pipelines on complete generated videos. Each reported score is the mean per-video normalized score multiplied by 100. Dynamic Degree is the percentage of videos meeting the official motion threshold.

The system also generates audio jointly with video.

C.3 Ablations and Reference Comparison

The ablation rows in Table 3 compare inference-time variants using the same video checkpoint, initial conditions, player inputs, and sampling settings.

1. **Full RIGen.** Predict interaction consequences before generation, apply executable rules, and guide the current chunk. Observe the generated chunk and reconcile predicted state with its visible outcome before the next decision.
2. **Motion guidance only.** Retain player inputs, scene, appearance, and source motion annotations. Remove hit, injury, and death guidance and the online interaction VLM.
3. **Semantic prompting.** The VLM maintains persistent textual state and selects the next chunk's combat outcome. Rules are supplied as text, without an executable rule engine or an independent verification-and-correction module. This variant compares agent-managed state updates with executable rules and separate visual reconciliation.
4. **Post-hoc event updates.** Generate first, observe events, apply rules, and pass consequence guidance to the following chunk. There is no pre-generation prediction of current-chunk consequences.
5. **Without reconciliation.** Retain pre-generation prediction and executable state updates, but do not correct the planned state from post-generation verification. The next reasoning step still sees the newly generated frames.
6. **RIGen (w/ Reference guidance).** Freeze reference event guidance in advance and remove online VLM calls. Death instructions are derived from source hit annotations combined with configured health and damage rules. They specify a counterfactual task answer, not ground-truth death in the source recording. The guidance is not revised from generated observations, but autoregressive video context uses this variant's own outputs.

C.4 RUNTIME

Table 5: Generation throughput at 832×480. Parameters count the generation backbone. Both RIGen settings include refinement.

Setting	Parameters	FPS
SCOPE	$\sim$5B	4.12
LingBot-World 2.0	14B	4.47
RIGen (Replay)	22B	4.41
RIGen (Online)	22B	3.93

Runtime is measured on a single NVIDIA GB300 GPU. In Table 5, Replay disables online VLM calls and reuses the same guidance, while Online runs the interaction loop. Both include refinement and remain below the target playback rate of 20 FPS.