GAMMA-WORLD: GENERATIVE MULTI-AGENT WORLD MODELING BEYOND TWO PLAYERS

Anonymous authors
Paper under double-blind review



Figure 1: We propose γ-World, a novel generative multi-agent world model from virtual games to real-world environments.

ABSTRACT

World models for interactive video generation have largely focused on single-agent settings, where future observations are rolled out from a single action stream, user input, or controllable viewpoint. However, many simulated worlds are inherently populated: multiple players, robots, or embodied agents act simultaneously within a shared, evolving environment. Scaling world models to such settings requires a principled multi-agent design: agents should remain independently controllable, permutation-symmetric, and support efficient inference while maintaining consistency across time and perspectives. In this paper, we present γ-World, a generative multi-agent world model for interactive simulation. γ-World introduces *Simplex Rotary Agent Encoding*, a parameter-free extension of 3D RoPE that represents agents as vertices of a regular simplex in rotary angle space. This gives each agent a distinct phase while making all agents permutation-equivalent, enabling scalable agent identity without learned per-slot identities or a fixed agent ordering. To support efficient cross-agent interaction, we further propose *Sparse Hub Attention*, where learnable hub tokens mediate communication across agents, reducing cross-agent attention cost from quadratic to linear in the number of agents. Finally, we use a bidirectional multi-agent teacher to guide a block-causal student with distillation, after which the final causal model can use KV caching for streaming, achieving real-time action-responsive rollouts at 24 FPS. Experiments in multiplayer virtual environments show that γ-World improves video fidelity, action controllability, and inter-agent consistency over slot-based and dense-attention baselines, while generalizing from two to four players without additional training.

1 INTRODUCTION

The worlds we wish to simulate are populated, not solitary. Players cooperate and compete in the same game, robot arms coordinate around shared objects, and embodied agents act under mutual physical and visual constraints. Controllable multi-agent world modeling is therefore a necessary step toward multiplayer game generation (Bruce et al., 2024; Savva et al., 2026), interactive simulation (Ren et al., 2025; Ali et al., 2025), and embodied AI (Fung et al., 2025; Feng et al., 2026).

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.

Despite this need, most video world models (Alonso et al., 2024; Valevski et al., 2024; He et al., 2025; Xiao et al., 2025; HunyuanWorld, 2025; Sun et al., 2025; Yu et al., 2025) remain single-agent simulators: they roll out future observations conditioned on one action stream, one user input, or one controllable viewpoint. Moving from single-agent to multi-agent simulation raises a new consistency requirement on top of standard video generation: generated observations must be consistent not only across time, but also across agent perspectives, since all agents share and act upon the same evolving world.

A concurrent effort in this direction is Solaris (Savva et al., 2026), which builds a multiplayer Minecraft world model by combining a dense joint attention block over all agent tokens with a learned per-player ID embedding. While effective in the two-player setting, this design has two structural limitations. First, dense joint attention couples every agent token to every other agent token, so its cost grows quadratically with the number of agents, and becomes increasingly restrictive for real-time rollouts beyond two players. Second, agents in a shared world are intrinsically exchangeable: two agents with identical capabilities should not be treated differently simply because they occupy different slots. A learned per-slot ID embedding violates this symmetry and ties the model to a fixed player roster that cannot be extended without retraining. This leaves open the question of whether a video world model can represent multiple agents in a way that is individually controllable, permutation-symmetric, and scalable beyond two players.

We address this question with γ-World, a generative multi-agent world model for interactive simulation that scales beyond two players. γ-World extends standard 3D RoPE with an explicit agent axis through *Simplex Rotary Agent Encoding*. Rather than assigning agents scalar indices or learned identity vectors, we place them at the vertices of a regular simplex in rotary angle space. Geometrically, this puts all agents at equal pairwise distances, so every pair is permutation-equivalent while each agent retains a distinct rotary phase. The encoding is parameter-free, does not rely on a fixed learned slot identity, and can be instantiated for different numbers of agents without changing the video transformer architecture.

To complement this symmetric agent representation with efficient cross-agent interaction, we introduce *Sparse Hub Attention*. Within each causal block, agent tokens attend to their own stream and to a small set of learnable hub tokens. The hub tokens aggregate information across agents and broadcast it back, providing a shared communication pathway without dense pairwise attention. This hub-mediated topology preserves a shared communication pathway among agents without dense pairwise interaction, reducing the dominant cross-agent cost from quadratic to linear in the number of agents and substantially relieving the scaling pressure of going beyond two players. To make the model deployable as a real-time interactive simulator, we further distill a bidirectional multi-agent teacher into a block-causal student with KV caching, enabling 24-FPS streaming autoregressive rollouts that respond to newly issued actions.

We evaluate γ-World on multiplayer virtual environments with two and four players spanning movement, mining, combat, and building scenarios. Across these settings, γ-World improves video fidelity, action controllability, and inter-agent consistency over slot-based and dense-attention baselines. Ablations validate the contributions of simplex rotary encoding and sparse hub communication. Scaling studies further show that, benefiting from the permutation-symmetric simplex agent encoding, the model generalizes from two to four players without additional training.

In summary, our main contributions are:

- We identify permutation symmetry as a fundamental property of multi-agent world models and propose *Simplex Rotary Agent Encoding*, a parameter-free rotary identity encoding that preserves permutation symmetry while maintaining distinct agent identities.
- We propose *Sparse Hub Attention*, a cross-agent communication mechanism that reduces cross-agent attention cost from quadratic to linear in the number of agents.
- We demonstrate effective multi-agent simulation in multiplayer environments with two and four players, with ablations validating the proposed agent encoding and efficiency design.

2 Related Work

Video generation. Diffusion-based generative models (Ho et al., 2020; Lipman et al., 2023; Song et al., 2021) have become a leading approach for video generation. Built upon latent diffusion models (Rombach et al., 2022), recent methods (Chen et al., 2024b; Guo et al., 2024; Yang et al.,

2025) perform generative modeling in the latent space, enabling more efficient synthesis of high-quality videos. In parallel, autoregressive video generation approaches (Chen et al., 2024a; Henschel et al., 2025; Kim et al., 2024) have attracted increasing attention for their potential to extend generation to arbitrary-length sequences, thereby offering a promising basis for building world models. Driven by scalable architectures (Peebles & Xie, 2023) and increasingly sophisticated data curation pipelines, large-scale video generation systems (Deepmind, 2025; OpenAI, 2024; Wan et al., 2025; Kuaishou, 2024; Gao et al., 2025; Kong et al., 2024) trained on large-scale datasets have further exhibited emergent zero-shot abilities to understand, model, and manipulate the visual world (Wiedemer et al., 2025), bringing physical world simulation closer to reality.

Video world models. With the development of video diffusion transformers and the rapid advances in text-to-video and image-to-video generation, a growing body of world-modeling research has begun to repurpose video diffusion models as visual simulators (Yang et al., 2023; Bar et al., 2025; Liang et al., 2024). Rather than learning compact abstract states for downstream planning and decision making, these methods directly model future visual observations through generative video prediction. We therefore refer to this family of approaches as video world models, highlighting their central reliance on video generation for simulating the evolution of the visual world. Recent video world models have been increasingly studied in embodied scenarios, where agents must anticipate the visual consequences of actions and interactions. Representative applications span robotics (Mereu et al., 2025; Yang et al., 2023; Li et al., 2025b; 2026b; Ye et al., 2026; Gao et al., 2026), video games (Alonso et al., 2024; He et al., 2025; Valevski et al., 2024; Xiao et al., 2025), autonomous driving (Agarwal et al., 2025; Hu et al., 2023), and physical simulation (Li et al., 2025a; Yuan et al., 2026), suggesting their potential as general-purpose simulators for dynamic visual environments. Beyond open-loop video rollout, recent efforts (HunyuanWorld, 2025; Sun et al., 2025; Yu et al., 2025) further move toward interactive and temporally consistent world modeling, where the simulated environment can react to user or agent actions while preserving coherent scene structure, object dynamics, and long-range temporal consistency.

Video diffusion model distillation. Real-time generation is a crucial requirement for building world models. A common strategy is to accelerate sampling through diffusion distillation (Salimans & Ho, 2022), which enables few-step inference and thus supports real-time synthesis. Complementary approaches learn one- or few-step generators directly, including MeanFlow (Geng et al., 2025) and Shortcut Models (Frans et al., 2024). Several works (Sauer et al., 2024a;b; Kang et al., 2024; Lin et al., 2025a; 2024; 2025b) introduce adversarial training objectives to reduce the number of denoising steps, but such methods are often prone to optimization instability and model collapse. Another line of approaches (Yin et al., 2024b;a; Lu et al., 2025) leverages variational score distillation (Wang et al., 2023) to obtain strong few-step generation performance. Beyond these general acceleration techniques, CausVid (Yin et al., 2025) tackles real-time autoregressive generation by transferring knowledge from a bidirectional diffusion teacher to a causal student model. Building on this causal generation setup, Self-Forcing (Huang et al., 2025) further improves rollout training to reduce the exposure bias accumulated during long-horizon generation.

3 METHOD

3.1 PRELIMINARIES

Latent video diffusion. We build on DiT-based latent video diffusion transformers (Peebles & Xie, 2023; Wan et al., 2025). Let $\mathbf{z}_0 \in \mathbb{R}^{T\times H\times W\times C_z}$ denote a clean video latent with T temporal positions, $H \times W$ spatial positions, and C_z channels per token. We adopt the flow-matching objective (Lipman et al., 2023): given $\mathbf{z}_0$, a noise sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and a noise level $\sigma \in [0, 1]$, we form the linear interpolant

$$\mathbf{z}_\sigma = (1-\sigma)\,\mathbf{z}_0 + \sigma\,\boldsymbol{\epsilon}, \tag{1}$$

and train a velocity field v_θ to regress its time derivative $\boldsymbol{\epsilon} - \mathbf{z}_0$:

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{\mathbf{z}_0,\,\boldsymbol{\epsilon},\,\sigma}\left[\left\|v_\theta(\mathbf{z}_\sigma, \sigma, \mathcal{C}) - (\boldsymbol{\epsilon} - \mathbf{z}_0)\right\|_2^2\right], \tag{2}$$

where $\mathcal{C}$ denotes conditioning signals such as initial observations and agent actions.

Causal autoregressive video generation. Full-context bidirectional attention improves global consistency but prevents streaming rollout. Self-Forcing (Huang et al., 2025) combines Diffusion Forcing (Chen et al., 2024a) with block-causal attention for autoregressive generation.

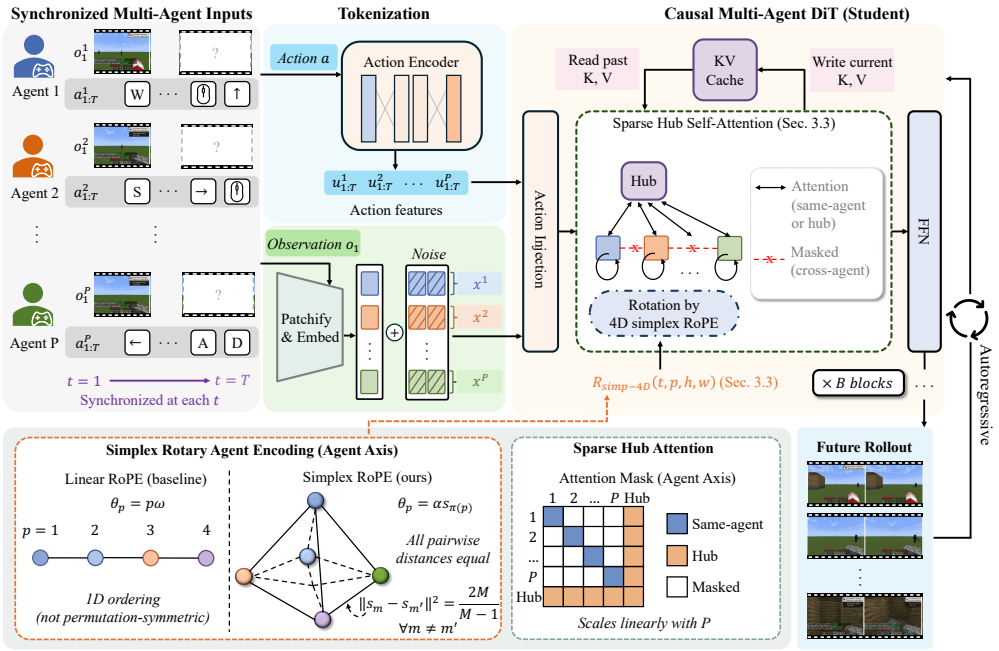


Figure 2: **Method overview.** γ-World takes synchronized observations and actions from multiple agents as input, tokenizes each agent stream with shared visual and action encoders, and generates future multi-agent rollouts with a causal multi-agent DiT. We formulate the input with an explicit synchronized agent axis (§3.2), encode exchangeable agent identity using Simplex Rotary Agent Encoding (§3.3), and route cross-agent information through Sparse Hub Attention (§3.3). During streaming inference, the causal student uses KV caches for past visual tokens and hub states to preserve block-causal generation while scaling efficiently with the number of agents (§3.4).

The latent sequence is partitioned into temporal blocks of n frames, an independent noise level $\sigma_b \sim \mathcal{U}(0, 1)$ is sampled for each block, and each query attends only to keys from the same or earlier blocks. Attention remains bidirectional within each block, matching streaming inference where the model repeatedly denoises the current block conditioned on previously generated blocks.

Rotary Position Embedding (RoPE). Video diffusion transformers commonly use 3D RoPE (Su et al., 2024) to inject relative position into self-attention by rotating query and key features along factorized temporal and spatial axes. The rotary head dimensions are split into temporal, height, and width bands of sizes d_t, d_h, d_w, with $d_t + d_h + d_w = d_{\text{rope}}$. For a token at coordinate (t, h, w), the 3D RoPE operator is

$$\mathbf{R}_{\text{3D}}(t, h, w) = \text{diag}(\mathbf{R}_t(t),\, \mathbf{R}_h(h),\, \mathbf{R}_w(w))\,, \tag{3}$$

where each $\mathbf{R}_x(x)$ is a block-diagonal of 2D rotations whose angles follow the standard RoPE frequency schedule (Su et al., 2024).

3.2 Problem Formulation

We consider P synchronized agents acting in a shared environment over T latent frames. At each time t, agent p receives an observation o_t^p and executes an action a_t^p. The observations from all agents are synchronized: $\{o_t^p\}_{p=1}^P$ correspond to different perspectives of the same underlying world state at time t, and the joint action $\{a_t^p\}_{p=1}^P$ drives the subsequent world evolution.

After visual tokenization, a clean multi-agent latent is represented as $\mathbf{Z}_0 \in \mathbb{R}^{P\times T\times H\times W\times C_z}$, which extends the single-agent latent $\mathbf{z}_0$ from §3.1 with an explicit agent axis indexed by $p \in \{1, \ldots, P\}$. The model is conditioned on initial observations and per-agent action sequences, and predicts future latent observations for all agents jointly. This formulation encourages generated rollouts to remain consistent across both time and agent perspectives. At inference, the first observations $\{o_1^p\}_{p=1}^P$ are encoded as context, while future latent tokens are initialized from noise and denoised block by block under the per-agent action sequences.

3.3 Model Design

Action conditioning. Each agent is controlled by its own action sequence $\mathbf{a}^p_{1:T}$. We use a shared action encoder f_a across all agents to map each action to a hidden action feature $\mathbf{u}^p_t = f_a(a^p_t) \in \mathbb{R}^D$, where D is the DiT hidden dimension. For discrete or multi-component controls, we embed each action component and combine the embeddings with a lightweight MLP.

Let $\mathbf{x}_{\ell,p,t,h,w} \in \mathbb{R}^D$ denote the transformer state at layer ℓ for agent p, frame t, and spatial position (h, w). At transformer block ℓ, the action feature is projected to a layer-specific action bias

$$\boldsymbol{\beta}^p_{\ell,t} = g_\ell(\mathbf{u}^p_t) \in \mathbb{R}^D, \tag{4}$$

and broadcast to all spatial tokens of the corresponding agent and frame:

$$\mathbf{x}_{\ell,p,t,h,w} \leftarrow \mathbf{x}_{\ell,p,t,h,w} + \boldsymbol{\beta}^p_{\ell,t}. \tag{5}$$

The biased tokens are then passed to self-attention. The action encoder and projection layers are shared across agents, so the same action has the same representation regardless of agent identity.

Simplex Rotary Agent Encoding. To distinguish agents without imposing an arbitrary slot ordering, we extend the 3D RoPE coordinate from §3.1 with an explicit agent axis p. We partition the rotary head dimension as $d_{\text{rope}} = d_t + d_p + d_h + d_w$, which yields the 4D rotary operator

$$\mathbf{R}_{\text{4D}}(t, p, h, w) = \text{diag}(\mathbf{R}_t(t),\, \mathbf{R}_p(p),\, \mathbf{R}_h(h),\, \mathbf{R}_w(w))\,. \tag{6}$$

A natural instantiation of $\mathbf{R}_p$ is to assign each agent a scalar phase $\boldsymbol{\theta}_p = p\,\boldsymbol{\omega}$ for some frequency vector $\boldsymbol{\omega}$. However, this places exchangeable agents on a one-dimensional line: different pairs receive different rotary distances depending on $|p - q|$, and the indexing convention can make certain slots structurally special. Learned per-slot embeddings have a different failure mode: they tie agent identity to a fixed roster and break permutation symmetry. We instead seek an encoding that distinguishes agents while making all agent identities *equidistant and exchangeable*.

To achieve this, we represent agents as vertices of a regular simplex in rotary angle space. Let V denote a fixed simplex pool size chosen at training time, corresponding to the maximum number of supported agent identities, with $V \le d_p/2 + 1$. We construct V simplex vertices in the $d_p/2$-dimensional agent-angle space:

$$\mathbf{s}_v = \sqrt{\tfrac{V}{V-1}}\,\mathbf{Q}\big(\mathbf{e}_v - \tfrac{1}{V}\mathbf{1}\big) \in \mathbb{R}^{d_p/2}, \qquad v = 1, \ldots, V, \tag{7}$$

where $\mathbf{e}_v \in \mathbb{R}^V$ is the v-th one-hot vector, $\mathbf{1} \in \mathbb{R}^V$ is the all-ones vector, and $\mathbf{Q}$ is a linear isometry from the zero-mean subspace of $\mathbb{R}^V$ into $\mathbb{R}^{d_p/2}$.

The resulting vertices have unit norm and equal pairwise distance:

$$\|\mathbf{s}_v\|_2 = 1, \qquad \|\mathbf{s}_v - \mathbf{s}_{v'}\|_2^2 = \tfrac{2V}{V-1} \quad \text{for all } v \neq v', \tag{8}$$

with the proof given in Appendix C.

For a batch with $P \le V$ active agents, we sample an injective assignment $\pi : \{1, \ldots, P\} \to \{1, \ldots, V\}$ and assign agent p to simplex vertex $\mathbf{s}_{\pi(p)}$. The agent-band rotation angles are

$$\boldsymbol{\theta}_p = \alpha\,\mathbf{s}_{\pi(p)}, \tag{9}$$

where $\alpha > 0$ controls the strength of agent separation. Replacing $\mathbf{R}_p$ with the corresponding simplex rotation $\mathbf{R}_{\text{simp}}(\pi(p))$ gives

$$\mathbf{R}_{\text{simp-4D}}(t, p, h, w) = \text{diag}(\mathbf{R}_t(t), \mathbf{R}_{\text{simp}}(\pi(p)), \mathbf{R}_h(h), \mathbf{R}_w(w))\,. \tag{10}$$

This encoding is parameter-free, permutation-symmetric, and does not introduce learned per-slot identities. The simplex pool provides a direct mechanism for agent-count scaling. During training, active agents are randomly assigned to distinct vertices from the fixed pool, discouraging slot-specific overfitting. At inference, additional agents can be activated by selecting additional unused vertices from the same pool, without changing the transformer architecture or introducing new learned identities. For pretrained video DiTs that lack an explicit agent band, we follow ReRoPE (Li et al., 2026a) and allocate d_p dimensions from the low-frequency end of the temporal band, leaving the high-frequency temporal and spatial bands intact.

Sparse Hub Attention. A direct way to model cross-agent interaction is dense joint attention over all agent tokens, as in Solaris (Savva et al., 2026). With $L = HW$ tokens per frame and P agents, dense cross-agent attention over a block of n frames costs $\mathcal{O}(P^2n^2L^2)$. This becomes expensive as the number of agents grows, and it is often unnecessary: in many shared-world settings, agents influence one another primarily through a compact evolving environment state rather than through dense token-level pairwise exchange at every layer.

We therefore introduce *Sparse Hub Attention* (SHA), a hub-mediated attention topology that adds a small set of learnable hub tokens as a compact shared communication state. Agent tokens attend only to tokens from the same agent stream and to the hub tokens. Hub tokens attend to all agents and to other hub tokens. Direct attention between distinct agent streams is masked, so cross-agent information flows through a two-hop path: agent $\rightarrow$ hub $\rightarrow$ agent.

We organize the sequence as PTL agent tokens followed by TK hub tokens, with K hub tokens per latent frame. The hub tokens are drawn from a learnable matrix $\mathbf{H} \in \mathbb{R}^{K\times D}$, broadcast across frames, and removed from the output, so they act purely as internal communication states. Let $\rho(i) \in \{1, \dots, P, \mathrm{hub}\}$ denote the identity of token i. The hub-and-spoke topology is defined by

$$\mathcal{M}_{\mathrm{hub}}(i, j) = \mathbf{1}[\rho(i) = \rho(j) \ \vee \ \rho(i) = \mathrm{hub} \ \vee \ \rho(j) = \mathrm{hub}]\,. \tag{11}$$

For causal autoregressive generation, we compose this topology with the block-causal mask (§3.1):

$$\mathcal{M}(i, j) = \mathbf{1}[b(j) \le b(i)] \cdot \mathcal{M}_{\mathrm{hub}}(i, j), \tag{12}$$

where $b(i)$ is the temporal block index of token i. The first factor enforces block-level causality, while the second enforces hub-mediated cross-agent communication. Hub tokens reuse the temporal RoPE phase of their associated frame and use identity rotations on the agent, height, and width bands, keeping them temporally aligned while remaining neutral to agent identity and spatial position.

Sparse Hub Attention preserves a shared interaction pathway while reducing the per-block attention cost from $\mathcal{O}(P^2n^2L^2)$ to $\mathcal{O}(P\,nL\,(nL + nK)) + \mathcal{O}(nK\,(P\,nL + nK))$, which is linear in P for fixed block size n, spatial length L, and number of hub tokens K.

3.4 Model Training and Inference

Our deployment target is a conditional few-step causal generator for real-time multi-agent rollout. A bidirectional diffusion model provides strong visual quality and cross-agent consistency, but cannot be used directly for online generation because it attends to future frames. A causal model supports KV-cached streaming, but training only on ground-truth histories creates a train–test mismatch during autoregressive rollout.

Inspired by Diffusion Forcing (Chen et al., 2024a) and Self-Forcing (Huang et al., 2025), we therefore use a three-stage recipe tailored to interactive multi-agent simulation: we first train a high-quality bidirectional teacher, then train a block-causal multi-step student with Sparse Hub Attention and Diffusion Forcing, and finally distill the causal student into a conditional few-step generator for low-latency streaming inference.

Bidirectional teacher. We first train a bidirectional teacher for high-quality conditional denoising. The teacher processes the full multi-agent sequence in one forward pass with dense bidirectional attention and a single shared noise level across all agent-time slots. It is conditioned on the same signals used at inference time, including first-frame observations and per-agent action sequences. Because the teacher is used only during training, it can exploit full temporal and cross-agent visibility to model local dynamics, agent interactions, and cross-perspective consistency. This teacher provides the high-quality conditional multi-agent distribution used in the final distillation stage.

Causal student. We separately train a causal autoregressive generator using the Diffusion Forcing formulation. The student combines block-causal attention from §3.1 with the Sparse Hub Attention mask $\mathcal{M}$ from §3.3. Each temporal block receives an independently sampled noise level, and each query attends only to the current or previous blocks. In addition to this causal constraint, agents communicate through shared hub tokens rather than dense pairwise attention.

Unlike post-training recipes (Huang et al., 2025; Yin et al., 2025) that use causal training mainly as a short warm-up before few-step distillation, we train the causal student as a full multi-step diffusion model. Before any few-step compression, the causal student can already produce reasonable autoregressive rollouts with multi-step sampling, providing a stable starting point for distillation.

Table 1: Comparison with Solaris across multi-agent evaluation protocols. FID and FVD are lower better (↓). Best per column in **bold**.

Method	Memory		Grounding		Movement		Building		Consistency	
	FVD ↓	FID ↓	FVD ↓	FID ↓	FVD ↓	FID ↓	FVD ↓	FID ↓	FVD ↓	FID ↓
Frame concat (Enigma-team, 2025)	450.6	69.8	528.3	63.2	556.9	65.0	551.8	87.3	576.0	123.2
Solaris (Savva et al., 2026)	333.8	51.7	301.9	36.1	311.1	36.3	448.6	71.0	443.1	94.8
γ-World (Ours)	**184.1**	**24.8**	**199.3**	**24.0**	**191.5**	**21.2**	**264.5**	**32.1**	**280.0**	**46.9**

Table 2: Architecture design. All metrics are averaged over the test scenarios in the game environment.

Setting	**Composition**	**Agent Encoding**	**Interaction**	FVD ↓	FID ↓	LPIPS ↓	PSNR ↑	SSIM ↑
Spatial Concat	Spatial concat	None	Full	312.4	38.7	0.326	24.8	0.782
Sequence Concat	Sequence concat	None	Full	285.6	35.2	0.298	25.6	0.798
View Embedding	Sequence concat	View emb.	Full	256.3	32.4	0.281	26.4	0.815
Simplex Encoding	Sequence concat	Simplex	Full	228.5	**29.6**	**0.265**	27.5	0.830
γ-World (Full)	Sequence concat	Simplex	Sparse Hub	**223.4**	30.2	0.269	**27.7**	**0.836**

Conditional Self-Forcing distillation. Finally, inspired by Self-Forcing (Huang et al., 2025), we distill the multi-step causal student into a few-step generator for real-time inference. The student is initialized from the trained multi-step causal model, while the bidirectional teacher provides the high-quality conditional distribution-matching signal. We train the few-step student under autoregressive self-rollout: generated blocks are written to the KV cache and reused as history for subsequent blocks, matching the inference-time rollout distribution. We apply distribution matching distillation (DMD) (Yin et al., 2024b) with rollout-aware training, encouraging the few-step student to preserve quality not only within each block, but also over its own generated histories.

Our setting requires conditional distillation. Interactive world models must preserve the initial observation and respond to actions, rather than merely produce plausible videos. We therefore provide the same conditioning package $\mathcal{C}$, including first-frame observations and per-agent actions, to both teacher and student during distillation. This aligns the conditional rollout distribution and prevents the few-step model from drifting away from the specified initial state or action controls.

KV-cached streaming inference. At inference time, the distilled few-step student generates one temporal block at a time, conditioned on the initial observations and the latest per-agent action block, and streams the resulting rollout at 24 FPS. To preserve the Sparse Hub Attention topology during streaming, we maintain separate KV caches for each agent stream and a shared KV cache for hub tokens. When generating a new block, each agent reads keys and values from its own past blocks and from the hub cache, while the hub reads cached states from all agents and previous hub tokens. Thus, cross-agent information still flows only through the hub, even when using cached histories.

4 Experiments

Experimental setup. Implementation details are provided in Appendix E. For virtual-game environments, we construct synchronized multi-agent Minecraft trajectories with a data-generation pipeline inspired by SolarisEngine (Savva et al., 2026), using controllable episode scripts, coordinated bots, and aligned visual-action recording. The game dataset includes two-agent episodes as the main setting and extends the same collection pipeline to more agents, such as four-agent scenes, enabling evaluation of both pairwise interaction and scalable multi-agent generation. We evaluate generation quality with FVD and FID, and measure perceptual and pixel-level quality with LPIPS, PSNR, and SSIM. To evaluate scalability, we measure DiT latency, self-attention latency, and self-attention FLOPs as the number of agents changes.

4.1 Quantitative Results

Comparison with multi-agent baselines. We compare γ-World against two representative baselines for multi-agent world modeling. The first is a frame-concatenation baseline, following the Multiverse-style design (Enigma-team, 2025), which merges multiple agent views into a single visual stream. The second is Solaris (Savva et al., 2026), a multiplayer Minecraft world model that explicitly trains on synchronized player trajectories. As shown in Table 1, our method achieves the best overall performance across the evaluation categories, demonstrating the effectiveness of our framework, including order-free agent identity encoding and sparse hub-based interaction. Compared with frame concatenation, our model avoids compressing multiple agents into a single undifferentiated visual stream, leading to more coherent interaction modeling and better preservation of each agent's

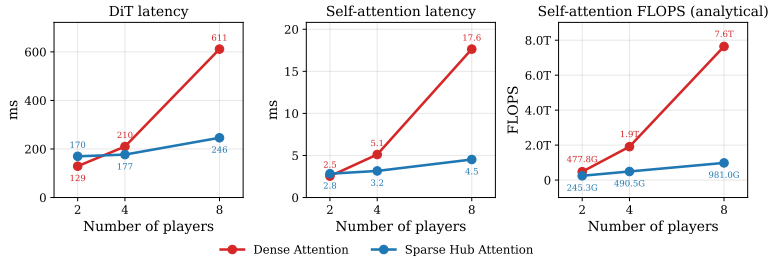


Figure 3: Efficiency comparison between dense cross-agent attention and Sparse Hub Attention across 2, 4, and 8 agents.

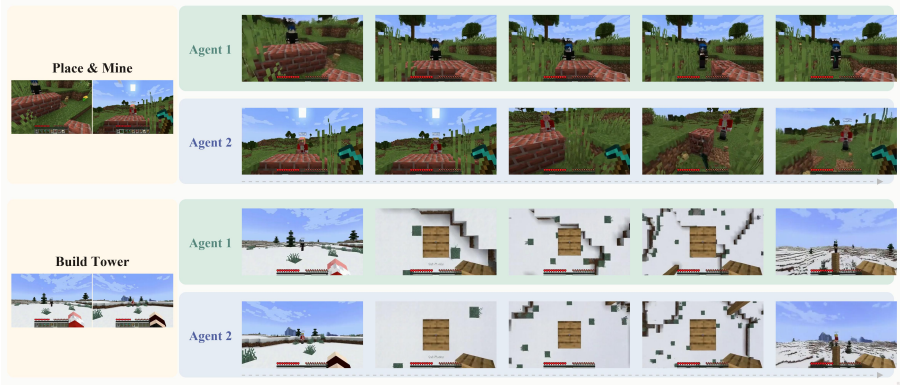


Figure 4: Qualitative examples of two-agent interaction. Each row shows a different task.

viewpoint. Compared with Solaris, our design further strengthens agent-level correspondence and information exchange, yielding more reliable results in scenarios that require memory, grounding, building, and cross-view consistency. These results indicate multi-agent world modeling benefits from treating agents as distinct but coupled entities rather than simply aggregating their visual observations.

Architecture ablations. In Table 2, we ablate the main architectural choices in γ-World: input organization, agent identity encoding, and cross-agent interaction. For input organization, *Spatial Concat* merges agent views into a larger canvas, while *Sequence Concat* preserves each agent as a separate stream. Although these designs can behave similarly in the two-agent setting, spatial concatenation becomes increasingly expensive as the number of agents grows because it increases the effective spatial resolution. Sequence concatenation keeps the per-agent spatial resolution fixed and is therefore more compatible with variable agent counts. Given sequence concatenation, we compare learned *View Embedding* against our *Simplex Rotary Agent Encoding*. Simplex Rotary Agent Encoding improves over learned view embeddings by assigning distinct agent identities without imposing a privileged slot order, matching the exchangeability of agents in a shared world. Finally, we replace dense cross-agent attention with *Sparse Hub Attention* to test whether agents can communicate through a compact shared state. As shown in Table 2, the full design, combining sequence-based multi-agent tokenization, Simplex Rotary Agent Encoding, and Sparse Hub Attention, achieves strong visual quality and consistency across FVD, FID, LPIPS, PSNR, and SSIM. These ablations support our core design principle: agents should be represented as distinct but exchangeable entities and coupled through an efficient shared interaction pathway.

Efficiency of Sparse Hub Attention. We further evaluate the efficiency of *Sparse Hub Attention* by comparing it with full cross-agent attention as the number of agents increases. This comparison directly tests the motivation in Sec. 3.3: dense all-to-all interaction grows quadratically with the number of agents, whereas *Sparse Hub Attention* routes cross-agent information through a compact shared state. We report DiT latency, self-attention latency, and self-attention FLOPs in Figure 3. Latency is averaged over 3 full rollouts to 24 latent frames with full KV cache, and FLOPs is computed analytically from the average token sequence length over the 24-latent rollout. *Sparse Hub Attention* substantially reduces computation time and FLOPs as the number of agents grows, while dense attention quickly becomes expensive at larger agent counts. These results show that *Sparse Hub Attention* provides a practical scaling path for multi-agent world models beyond two players.

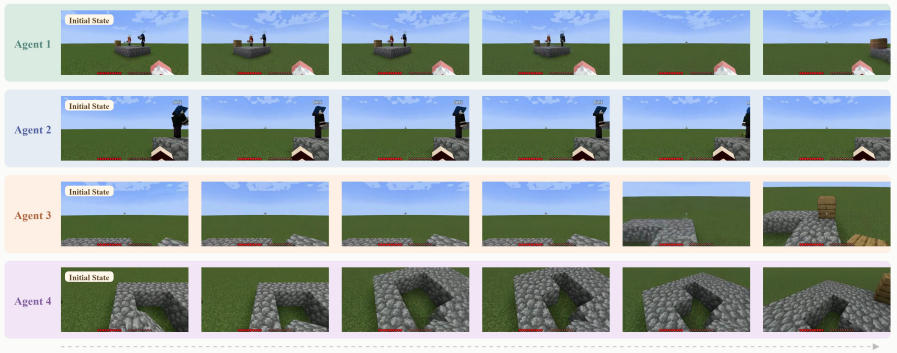


Figure 5: Qualitative example of scaling beyond two players. The first frame in each row shows the initial state for one agent, followed by synchronized rollouts across four agents.



Figure 6: Qualitative examples of real-world robotic coordination.

4.2 Qualitative Results

We complement the quantitative evaluation with qualitative examples that visualize how the model preserves shared-world consistency across multiple agent viewpoints.

Two-agent interaction. Figure 4 shows two-agent rollouts generated by γ-World. The paired streams remain synchronized: actions taken by one agent are reflected in the other agent's observation when the agents interact in the shared environment. The model also maintains object and agent grounding when agents temporarily move out of each other's field of view, suggesting that it tracks a shared latent world state rather than generating independent single-agent videos.

Scaling beyond two players. Figure 5 shows zero-shot four-agent rollouts from a model trained only on two-agent data. The same model generates synchronized visual streams for multiple players without changing the architecture. This scaling behavior is enabled by Simplex Rotary Agent Encoding, which avoids fixed learned slot identities, and Sparse Hub Attention, which provides a shared communication pathway without dense pairwise attention. The examples indicate that γ-World learns coupled multi-agent dynamics rather than independently rolling out each view.

Real-world robotics applications. Beyond games, we also conceptually evaluate γ-World on real-world robot coordination tasks. We use the RealOmin-Open Dataset (Gen Robot, 2025), treating the left and right robot arms as two interacting agents. As shown in Figure 6, the model generates future frames that preserve the coordinated motion of multiple robotic agents and the spatial layout of the scene. These examples indicate that the same generative multi-agent formulation can extend from virtual games to real-world environments.

5 Discussion

We presented γ-World, a generative multi-agent world model for interactive simulation beyond two players. γ-World combines Simplex Rotary Agent Encoding for distinct yet permutation-symmetric agent identities with Sparse Hub Attention for efficient hub-mediated cross-agent communication. Together with conditional teacher–student distillation and KV-cached streaming inference, these components enable real-time action-responsive rollouts that remain consistent across time and agent perspectives. Experiments in multiplayer virtual environments show that γ-World improves fidelity, controllability, and inter-agent consistency over baselines. Ablations validate the design choices, while scaling experiments show that the model generalizes from two-player training to four-player simulation without additional training and achieves lower inference cost as the number of agents increases. We further demonstrate the same formulation on real-world robotic coordination by treating the left and right robot arms as interacting agents.

AI USE STATEMENT

In this work, we used generative AI tools for translating portions of the manuscript and assisting with the implementation of our methods. We have not used generative AI tools for any other tasks requiring disclosure under the ICLR policy. Additionally, we used generative AI tools for information retrieval and discovery of related work, drafting portions of the manuscript, polishing the writing for clarity and readability, and correcting grammatical errors. We have reviewed all AI-assisted work. Specifically, the authors reviewed all AI-assisted text and translations for accuracy and fidelity to the intended meaning, verified references against the original sources, and reviewed all AI-assisted code for correctness. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.

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

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *CVPR*, pp. 15791–15801, 2025.

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

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024a.

Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *CVPR*, pp. 7310–7320, 2024b.

Google Deepmind. Veo3 video model, 2025. `https://deepmind.google/models/veo/`.

Enigma-team. Introducing Multiverse: The first AI multiplayer world model. Enigma Blog, 2025.

Zhaohan Feng, Ruiqi Xue, Lei Yuan, Yang Yu, Ning Ding, Meiqin Liu, Bingzhao Gao, Jian Sun, Xinhu Zheng, and Gang Wang. Multi-agent embodied ai: Advances and future directions. *Science China Information Sciences*, 69(5):151202, 2026.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. *arXiv preprint arXiv:2410.12557*, 2024.

Pascale Fung, Yoram Bachrach, Asli Celikyilmaz, Kamalika Chaudhuri, Delong Chen, Willy Chung, Emmanuel Dupoux, Hongyu Gong, Hervé Jégou, Alessandro Lazaric, et al. Embodied ai agents: Modeling the world. *arXiv preprint arXiv:2506.22355*, 2025.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, Liang Li, Xiaojie Li, et al. Seedance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113*, 2025.

Gen Robot. 10kh-realomin-opendata, 2025.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *ICLR*, 2024.

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

Roberto Henschel, Levon Khachatryan, Hayk Poghosyan, Daniil Hayrapetyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Streamingt2v: Consistent, dynamic, and extendable long video generation from text. In *CVPR*, pp. 2568–2577, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

Team HunyuanWorld. Hy-world 1.5: A systematic framework for interactive world modeling with real-time latency and geometric consistency. *arXiv preprint*, 2025.

Minguk Kang, Richard Zhang, Connelly Barnes, Sylvain Paris, Suha Kwak, Jaesik Park, Eli Shechtman, Jun-Yan Zhu, and Taesung Park. Distilling diffusion models into conditional gans. In *ECCV*, pp. 428–447, 2024.

Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. Fifo-diffusion: Generating infinite videos from text without training. *Advances in Neural Information Processing Systems*, 37: 89834–89868, 2024.

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.

Kuaishou. Kling video model, 2024. `https://klingai.com/global/`.

Chenyu Li, Oscar Michel, Xichen Pan, Sainan Liu, Mike Roberts, and Saining Xie. Pisa experiments: Exploring physics post-training for video diffusion models by watching stuff drop. *arXiv preprint arXiv:2503.09595*, 2025a.

Chunyang Li, Yuanbo Yang, Jiahao Shao, Hongyu Zhou, Katja Schwarz, and Yiyi Liao. Rerope: Repurposing rope for relative camera control. *arXiv preprint arXiv:2602.08068*, 2026a.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026b.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025b.

Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. *arXiv preprint arXiv:2406.16862*, 2024.

Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024.

Shanchuan Lin, Xin Xia, Yuxi Ren, Ceyuan Yang, Xuefeng Xiao, and Lu Jiang. Diffusion adversarial post-training for one-step video generation. In *ICML*, 2025a.

Shanchuan Lin, Ceyuan Yang, Hao He, Jianwen Jiang, Yuxi Ren, Xin Xia, Yang Zhao, Xuefeng Xiao, and Lu Jiang. Autoregressive adversarial post-training for real-time interactive video generation. *arXiv preprint arXiv:2506.09350*, 2025b.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023.

Yanzuo Lu, Yuxi Ren, Xin Xia, Shanchuan Lin, Xing Wang, Xuefeng Xiao, Andy J Ma, Xiaohua Xie, and Jian-Huang Lai. Adversarial distribution matching for diffusion distillation towards efficient image and video synthesis. In *ICCV*, pp. 16818–16829, 2025.

Riccardo Mereu, Aidan Scannell, Yuxin Hou, Yi Zhao, Aditya Jitta, Antonio Dominguez, Luigi Acerbi, Amos Storkey, and Paul Chang. Generative world modelling for humanoids: 1x world model challenge technical report. *arXiv preprint arXiv:2510.07092*, 2025.

OpenAI. Sora video model, 2024. `https://sora.chatgpt.com/`.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pp. 4195–4205, 2023.

Xuanchi Ren, Yifan Lu, Tianshi Cao, Ruiyuan Gao, Shengyu Huang, Amirmojtaba Sabour, Tianchang Shen, Tobias Pfaff, Jay Zhangjie Wu, Runjian Chen, et al. Cosmos-drive-dreams: Scalable synthetic driving data generation with world foundation models. *arXiv preprint arXiv:2506.09042*, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In *SIGGRAPH Asia*, pp. 1–11, 2024a.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *ECCV*, pp. 87–103, 2024b.

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.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

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`.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837*, 2024.

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

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *NeurIPS*, 36:8406–8441, 2023.

Thaddäus Wiedemer, Yuxuan Li, Paul Vicol, Shixiang Shane Gu, Nick Matarese, Kevin Swersky, Been Kim, Priyank Jaini, and Robert Geirhos. Video models are zero-shot learners and reasoners. *arXiv preprint arXiv:2509.20328*, 2025.

Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. Worldmem: Long-term consistent world simulation with memory. *arXiv preprint arXiv:2504.12369*, 2025.

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

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *ICLR*, 2025.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *NeurIPS*, 37: 47455–47487, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, pp. 6613–6623, 2024b.

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 *CVPR*, pp. 22963–22974, 2025.

Jiwen Yu, Jianhong Bai, Yiran Qin, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Context as memory: Scene-consistent interactive long video generation with memory retrieval. *arXiv preprint arXiv:2506.03141*, 2025.

Jianhao Yuan, Xiaofeng Zhang, Felix Friedrich, Nicolas Beltran-Velez, Melissa Hall, Reyhane Askari-Hemmat, Xiaochuang Han, Nicolas Ballas, Michal Drozdzal, and Adriana Romero-Soriano. Inference-time physics alignment of video generative models with latent world models. *arXiv preprint arXiv:2601.10553*, 2026.

A Limitations and Future Work

Our current evaluation focuses primarily on gaming environments and robotics examples; broader validation in more complex, heterogeneous, and long-horizon settings remains future work. The simplex pool supports agent-count scaling within a fixed rotary agent band, but very large populations may require larger bands or hierarchical agent grouping. Finally, because γ-World does not explicitly enforce 3D geometry or physical constraints, long rollouts may still accumulate inconsistencies.

B More Visualizations

We include extended 24-second multi-agent rollouts, qualitative comparisons against baselines, and real-world robotic coordination videos in the supplementary video, offering a more intuitive demonstration of our method.

C Proof of Simplex Equidistance

We provide the derivation for the equidistance property of Simplex Rotary Agent Encoding. Let V denote the simplex pool size, matching the notation of the main paper. We first construct the simplex vertices in $\mathbb{R}^V$ and then embed them into the $(d_p/2)$-dimensional agent angle space, where d_p is the agent rotary band size and $V \leq d_p/2 + 1$.

Let $\mathbf{e}_p \in \mathbb{R}^V$ denote the one-hot vector of agent p, and let $\mathbf{1} \in \mathbb{R}^V$ denote the all-one vector. We define the centered vector

$$\bar{\mathbf{s}}_p = \mathbf{e}_p - \frac{1}{V}\mathbf{1}. \tag{13}$$

These vectors lie in the zero-mean subspace because

$$\mathbf{1}^\top \bar{\mathbf{s}}_p = 0. \tag{14}$$

For any agent p, the squared norm of $\bar{\mathbf{s}}_p$ is

$$\|\bar{\mathbf{s}}_p\|_2^2 = \left(\mathbf{e}_p - \frac{1}{V}\mathbf{1}\right)^\top \left(\mathbf{e}_p - \frac{1}{V}\mathbf{1}\right) \tag{15}$$

$$= 1 - \frac{2}{V} + \frac{1}{V} \tag{16}$$

$$= \frac{V-1}{V}. \tag{17}$$

For two distinct agents $p \neq q$, their inner product is

$$\bar{\mathbf{s}}_p^\top \bar{\mathbf{s}}_q = \left(\mathbf{e}_p - \frac{1}{V}\mathbf{1}\right)^\top \left(\mathbf{e}_q - \frac{1}{V}\mathbf{1}\right) \tag{18}$$

$$= 0 - \frac{1}{V} - \frac{1}{V} + \frac{1}{V} \tag{19}$$

$$= -\frac{1}{V}. \tag{20}$$

After normalization, we obtain

$$\mathbf{s}_p = \sqrt{\frac{V}{V-1}}\bar{\mathbf{s}}_p. \tag{21}$$

Therefore,

$$\|\mathbf{s}_p\|_2^2 = 1, \tag{22}$$

$$\mathbf{s}_p^\top \mathbf{s}_q = -\frac{1}{V-1}, \qquad p \neq q. \tag{23}$$

The squared distance between any two distinct simplex vertices ($V \geq 2$) is then

$$\|\mathbf{s}_p - \mathbf{s}_q\|_2^2 = \|\mathbf{s}_p\|_2^2 + \|\mathbf{s}_q\|_2^2 - 2\mathbf{s}_p^\top \mathbf{s}_q \tag{24}$$

$$= 2 + \frac{2}{V-1} \tag{25}$$

$$= \frac{2V}{V-1}, \qquad p \neq q. \tag{26}$$

Thus, all distinct agent pairs are exactly equidistant in the simplex angle space.

In Simplex Rotary Agent Encoding, the agent angle is defined as

$$\boldsymbol{\theta}_p = \alpha \mathbf{s}_p, \tag{27}$$

where α is a scalar scale factor. Therefore, for any $p \neq q$,

$$\|\boldsymbol{\theta}_p - \boldsymbol{\theta}_q\|_2^2 = \alpha^2 \|\mathbf{s}_p - \mathbf{s}_q\|_2^2 \tag{28}$$

$$= \alpha^2 \frac{2V}{V-1}. \tag{29}$$

Hence, all different-agent pairs receive the same amount of separation in the underlying agent angle space. We further analyze the induced distance in the complex RoPE space. The complex rotary representation of agent p is

$$\boldsymbol{\Phi}_p = \exp(i\boldsymbol{\theta}_p). \tag{30}$$

For two agents p and q, the squared distance between their complex rotations is

$$\|\boldsymbol{\Phi}_p - \boldsymbol{\Phi}_q\|_2^2 = \|\exp(i\boldsymbol{\theta}_p) - \exp(i\boldsymbol{\theta}_q)\|_2^2 \tag{31}$$

$$= \sum_{r=1}^{d_p/2} \left| e^{i\theta_p^r} - e^{i\theta_q^r} \right|^2 \tag{32}$$

$$= \sum_{r=1}^{d_p/2} 2\left(1 - \cos(\theta_p^r - \theta_q^r)\right). \tag{33}$$

For sufficiently small α such that the coordinate-wise angle differences $|\theta_p^r - \theta_q^r|$ are small for all r, we use the approximation $1 - \cos x \approx x^2/2$. Thus,

$$\|\boldsymbol{\Phi}_p - \boldsymbol{\Phi}_q\|_2^2 \approx \sum_{r=1}^{d_p/2} (\theta_p^r - \theta_q^r)^2 \tag{34}$$

$$= \|\boldsymbol{\theta}_p - \boldsymbol{\theta}_q\|_2^2 \tag{35}$$

$$= \alpha^2 \frac{2V}{V-1}, \qquad p \neq q. \tag{36}$$

This shows that Simplex Rotary Agent Encoding guarantees exact equidistance in the agent angle space and yields approximately equal pairwise separation in the complex RoPE space. In the common implementation where the centered one-hot simplex is embedded with zero padding and $d_p/2 \geq V$, the complex-space distance is also identical across distinct agent pairs because all pairwise difference vectors share the same non-zero coordinate pattern up to permutation.

D More Architecture

D.1 Action Design

We use explicit per-agent action traces as additional conditioning signals. For both domains, actions are synchronized with the video frames and provided separately for different agents. The action specification is domain-specific: the game domain uses player control commands, while the robot domain uses continuous end-effector state.

Game action format. For Minecraft-style game videos, each agent action at a frame contains 25 fields: 23 discrete player controls and 2 continuous camera controls. The discrete controls cover

Table 3: Game action format. Each frame stores a 25-field action vector for each agent.

Index	Field	Description
0	`inventory`	Open inventory
1	`ESC`	Exit or cancel current menu
2–10	`hotbar.1`–`hotbar.9`	Select hotbar slot 1–9
11–14	`forward`, `back`, `left`, `right`	Locomotion controls
15–17	`jump`, `sneak`, `sprint`	Movement modifiers
18	`swapHands`	Swap held item between hands
19–22	`attack`, `use`, `pickItem`, `drop`	Interaction and item manipulation
23–24	`cameraX`, `cameraY`	Horizontal yaw and vertical pitch motion

Table 4: Robot action format. Each frame stores a 10-field continuous action vector for each robot.

Index	Field	Description
0–2	`pos_x`, `pos_y`, `pos_z`	End-effector position
3–8	`rot_6d_0`–`rot_6d_5`	End-effector orientation
9	`gripper`	Gripper opening value

inventory interaction, hotbar selection, movement, item manipulation, and mouse-button actions. The continuous camera controls describe horizontal and vertical view motion. Table 3 summarizes the game action fields.

Robot action format. For robot videos, each agent action at a frame contains 10 continuous fields describing the robot end-effector and gripper. The action contains the 3D end-effector position, a 6D orientation field, and the gripper opening value. The same format is used for the left and right robots, producing one temporally aligned action sequence per robot. Table 4 summarizes the robot action fields.

E Additional Implementation Details

Architecture. Both bidirectional teacher model and causal student model are based on Cosmos-Predict2.5-2B (Ali et al., 2025), with hidden dimension $D = 2048$, 28 transformer blocks, 16 attention heads (head dimension 128), MLP ratio 4, and AdaLN-LoRA of rank 256. Player identity is injected via our Simplex Rotary Agent Encoding (Sec. 3.3), partitioning the head dimension as $(64, 32, 16, 16)$ across the (t, p, h, w) axes and assigning each agent a vertex of a regular simplex as its rotation phase. We use a simplex pool of size 4 over 2 active runtime slots: at every training step we randomly sample 2 of the 4 vertices and additionally permute the agent slot order, forcing the model to disambiguate players exclusively through the simplex marker and allowing the same checkpoint to serve up to 4 players at inference. Per-player actions (a 23-dimensional keyboard one-hot and a 2-dimensional camera vector) are encoded by a single shared action encoder: two MLP branches that lift each modality to 128 dimensions, a fusion MLP that maps the concatenated 256-dimensional vector through a $4\times$-stride 1D temporal convolution, and a final projection to the model dimension $D = 2048$. The encoder output is added as a per-block bias to the self-attention input of every transformer layer. The causal student additionally enables (i) *Sparse Hub Attention*: each player attends only to its own past tokens and to $K = 8$ learnable global hub tokens per latent frame; and (ii) *local windowed attention*: each query attends to the most recent 24 latent frames per view, bounding the inference KV cache independently of generation length. The bidirectional teacher uses standard dense self-attention.

Stages 1 and 2 – Bidirectional teacher and causal student pretraining. Both the teacher and the student are initialized from the publicly released Cosmos-Predict2.5-2B TI2V checkpoint and trained on 2-agent gameplay at 320×480 per view. The teacher is first pretrained on 93-frame clips (latent length 24) for 10,000 iterations and then fine-tuned on 189-frame clips (latent length 48) for 6,000 iterations, while the causal student is pretrained on 93-frame clips for 15,000 iterations.

Table 5: Comparison of causal, bidirectional, and distilled variants of our model.

Variant	FVD ↓	FID ↓	LPIPS ↓	PSNR ↑	SSIM ↑
Bidirectional	227.3	31.0	0.272	27.7	0.828
Causal	266.4	34.4	0.277	26.2	0.805
Distilled	239.7	30.9	0.273	26.8	0.811

Table 6: Ablation on the number of hub tokens in Sparse Hub Attention. We vary the hub token count K and report generation quality, perceptual quality, and pixel-level quality.

Hub Tokens (K)	FVD ↓	FID ↓	LPIPS ↓	PSNR ↑	SSIM ↑
1	250.9	31.5	0.271	27.3	0.825
8	223.4	30.2	0.269	27.7	0.836
32	221.8	29.8	0.267	27.9	0.838
128	220.5	29.5	0.266	28.0	0.839

Both stages use AdamW with learning rate 3×10^{-5}, weight decay 10^{-3}, $(\beta_1, \beta_2) = (0.0, 0.999)$, 100-step linear warm-up, and gradient clipping at 0.1. The teacher is trained on 32 NVIDIA GB200s and the student on 32 NVIDIA GB200s. We do not apply classifier-free guidance (CFG) at inference, as we empirically observe that unguided sampling produces more accurate results.

Stage 3 – Self-Forcing distillation. The distillation model couples three networks: the student (trainable, initialized from Stage 2), a frozen *real score* (the Stage-1 teacher), and a trainable *fake score* also initialized from the Stage-1 teacher. We follow Self-Forcing (Huang et al., 2025) to optimize the student model with the DMD (Yin et al., 2024b) loss on 189-frame clips. At each generator step, each block is denoised with timesteps $\{1000, 750, 500, 250\}$ (warped by the flow shift 5.0). After each block, the model re-forwards the block under context-noise level 128 and writes the result into the per-layer KV cache before proceeding. Generator and critic are updated alternately at a $1{:}4$ ratio: the student is updated once every 5 iterations, while on the remaining iterations the fake critic is updated with a flow-velocity prediction loss; the real score is kept frozen throughout. Distillation runs for 400 iterations on 32 NVIDIA GB200s, using AdamW with learning rates 2×10^{-6} (student) and 4×10^{-7} (critic), weight decay 10^{-2}, $(\beta_1, \beta_2) = (0.0, 0.999)$.

Inference. At inference time, the student generates each per-view latent block autoregressively with the same 4-step denoising schedule and block size as in training. The KV cache uses the rolling local-attention window of 24 latent frames per view, decoupling the generated sequence length from cache memory.

F Additional Ablations

We compare three variants of our model: causal, bidirectional, and distilled. As shown in Table 5, we report FVD, FID, LPIPS, PSNR, and SSIM to evaluate perceptual quality and reconstruction fidelity.

We provide an additional ablation on the number of hub tokens K used in Sparse Hub Attention. In our design, hub tokens serve as the compact shared state through which agents exchange information, so K controls the capacity of the cross-agent communication bottleneck. When K is too small, the hub has limited capacity to summarize multi-agent state, which can hurt generation quality. Increasing K provides a richer shared state by allowing the hub to summarize multi-agent interactions with greater capacity. As shown in Table 6, we sweep $K \in \{1, 8, 32, 128\}$ and report FVD, FID, LPIPS, PSNR, and SSIM. This ablation helps identify a practical operating point where the hub has enough capacity for cross-agent interaction.

G More Experimental Results

G.1 Training Stage Comparison

We compare three training stages of our model: the bidirectional teacher, the causal student, and the distilled model. As shown in Table 5, the bidirectional teacher achieves the best overall performance due to its access to full temporal context during generation. The causal variant, while enabling streaming inference with KV caching, shows degraded performance because it can only attend to past frames. The distilled model recovers much of the teacher's quality while retaining the causal structure, demonstrating that knowledge distillation effectively transfers bidirectional modeling capacity into a streaming-compatible architecture.