Gamma-World: Generative Multi-Agent World Modeling Beyond Two Players

Fangfu Liu[1,2*] **Kai He**[1,3,4*] **Tianchang Shen**[1] **Tianshi Cao**[1] **Sanja Fidler**[1,3,4] **Yueqi Duan**[2] **Jun Gao**[1]
Igor Gilitschenski[3,4†] **Zian Wang**[1†] **Xuanchi Ren**[1†]
[1]NVIDIA [2]Tsinghua University [3]University of Toronto [4]Vector Institute

Project page: research.nvidia.com/gamma-world

Figure 1: We propose γ-World, a novel generative multi-agent world model from virtual games to real-world environments. More results and video demos are available on our project page.

Abstract

World models for interactive video generation have largely focused on single-agent settings, where future observations are generated from a single control signal. However, many generated environments require multi-agent interaction: multiple players, robots, or embodied agents act simultaneously within a shared space. 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 avoid dense all-to-all attention across agents, we further propose *Sparse Hub Attention*, where learnable hub tokens mediate token-interaction across agents, reducing cross-agent attention cost from quadratic to linear in the number of agents. For real-time rollout, we distill a full-context diffusion teacher into a causal student that generates temporal blocks sequentially with KV caching, enabling action-responsive generation 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.

*Equal contribution; order interchangeable.
†Joint advising.

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 [5, 47], interactive simulation [2, 42], and embodied AI [11, 13].

Despite this need, most video world models [3, 19, 24, 50, 51, 55, 63] 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 [47], 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 [21, 37, 48] have become a leading approach for video generation. Built upon latent diffusion models [43], recent methods [7, 18, 58] perform generative modeling in the latent space, enabling more efficient synthesis of high-quality videos. In parallel, autoregressive video generation approaches [6, 20, 26] 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 [41] and increasingly sophisticated data curation pipelines, large-scale video generation systems [8, 15, 27, 28, 40, 52] trained on large-scale datasets have further exhibited emergent zero-shot abilities to understand, model, and manipulate the visual world [54], 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 [4, 33, 57]. 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 [14, 31, 32, 39, 56, 59], video games [3, 10, 19, 51, 55], autonomous driving [1, 22], and physical simulation [29, 64], suggesting their potential as general-purpose simulators for dynamic visual environments. Beyond open-loop video rollout, recent efforts [24, 50, 63] 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 for video diffusion models is to accelerate sampling through distillation [12, 17, 44], which enables few-step inference and thus supports real-time synthesis. Several works [25, 34, 35, 36, 45, 46] 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 [38, 60, 61] leverages variational score distillation [53] to obtain strong few-step generation performance. Beyond these general acceleration techniques, CausVid [62] 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 [23] further improves rollout training to reduce the exposure bias accumulated during long-horizon generation.

3. Method

We consider the problem of synchronized action-conditioned multi-agent video generation. Formally, the goal is to learn a function γ-World$(\{o^p_{1:t}\}^P_{p=1}, \{a^p_{1:t}\}^P_{p=1})$, that takes a sequence of past observations and actions for P agents as input and produces the next observation for each agent $\{o^p_{t+1}\}^P_{p=1}$. Because we focus on

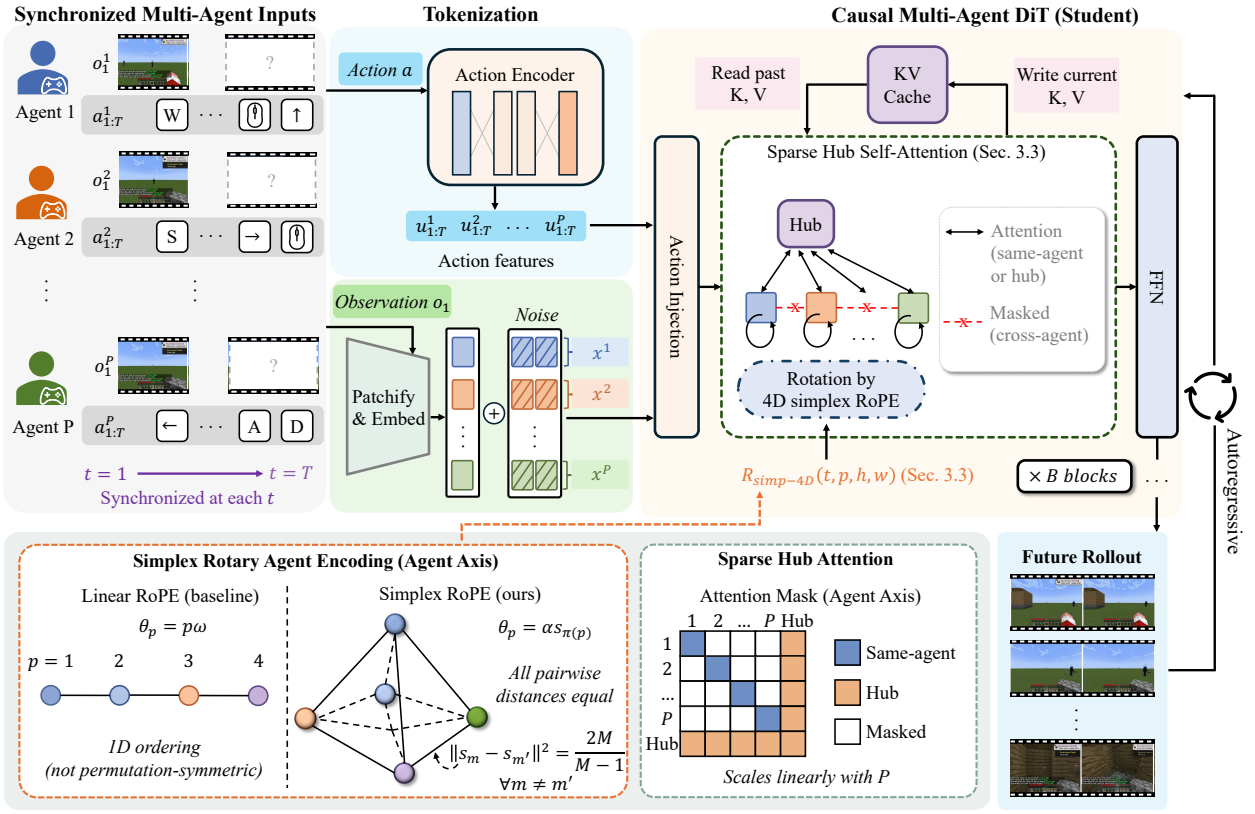

Figure 2: **Method overview.** γ-World takes synchronized observations and actions from multiple agents as input, tokenizes each agent stream with shared visual and action encoders, and generates future multi-agent rollouts with a causal multi-agent DiT. We formulate the input with an explicit synchronized agent axis, encode exchangeable agent identity using Simplex Rotary Agent Encoding (§3.2), and route cross-agent information through Sparse Hub Attention (§3.2). 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 (§3.3).

the multi-agent setting, it is important to note that $\{o_t^p\}_{p=1}^P$ correspond to different perspectives of the same underlying world state at time t.

Our proposed model builds on transformer-based latent video diffusion models adapted for autoregressive generation, in which rotary position embeddings encode spatial and temporal locations (§3.1). We modify this position embedding to account for agent identities and propose a multi-agent aware attention masking mechanism to reduce computational cost (§3.2). Finally, the full model is trained in two steps: first, a bidirectional teacher model is trained, and then it is distilled into a causal student model that supports the streaming setting (§3.3). An overview of the method is provided in figure 2.

3.1. Preliminaries

Latent video diffusion. We build on DiT-based latent video diffusion transformers [41, 52]. 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 [37]: 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. Standard video diffusion transformers are typically bidirectional: during denoising, tokens can attend to the full latent video, which improves global consistency but prevents streaming rollout. To enable autoregressive generation, Self-Forcing [23] combines Diffusion Forcing [6] with block-causal attention. 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 [49] 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 $d_t + d_h + d_w = d_{\mathrm{rope}}$. For a token at coordinate (t, h, w), the 3D RoPE operator is

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

where each $\mathbf{R}_x(x)$ is a block-diagonal of 2D rotations whose angles follow the standard RoPE frequency schedule [49].

3.2. Model Design

Our model starts by performing visual tokenization, after which 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.

Action conditioning. Each agent is controlled by its own action sequence $\mathbf{a}_{1:T}^p$. We use a shared action encoder f_a across all agents to map each action to a hidden action feature $\mathbf{u}_t^p = f_a(a_t^p) \in \mathbb{R}^D$, 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

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

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

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

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 $d_{\mathrm{rope}} = d_t + d_p + d_h + d_w$, which yields the 4D rotary operator

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

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:

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

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 B.

For a batch with $P \le V$ active agents, we sample an injective assignment $\pi : \{1, \dots, P\} \to \{1, \dots, V\}$ 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

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

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 [30] 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 [47]. With $L = HW$ tokens per frame and P agents, dense cross-agent attention over a block of n frames costs $\mathcal{O}(P^2 n^2 L^2)$. 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 $\to$ hub $\to$ 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 $\rho(i) \in \{1, \ldots, P, \mathrm{hub}\}$ 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

$$\mathcal{O}(P\,nL\,(nL + nK)) + \mathcal{O}(nK\,(P\,nL + nK))\,, \tag{13}$$

which is linear in P for fixed block size n, spatial length L, and number of hub tokens K.

3.3. 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 [6] and Self-Forcing [23], 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 §3.2. 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 [23, 62] 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.

Conditional Self-Forcing distillation. Finally, inspired by Self-Forcing [23], 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

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 [9]	450.6	69.8	528.3	63.2	556.9	65.0	551.8	87.3	576.0	123.2
Solaris [47]	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**

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) [61] 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 D. For virtual-game environments, we construct synchronized multi-agent Minecraft trajectories with a data-generation pipeline inspired by SolarisEngine [47], 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,

Figure 3: Efficiency comparison between dense cross-agent attention and Sparse Hub Attention across 2, 4, and 8 agents. Sparse Hub Attention achieves significantly lower latency and FLOPs as the number of agents increases.

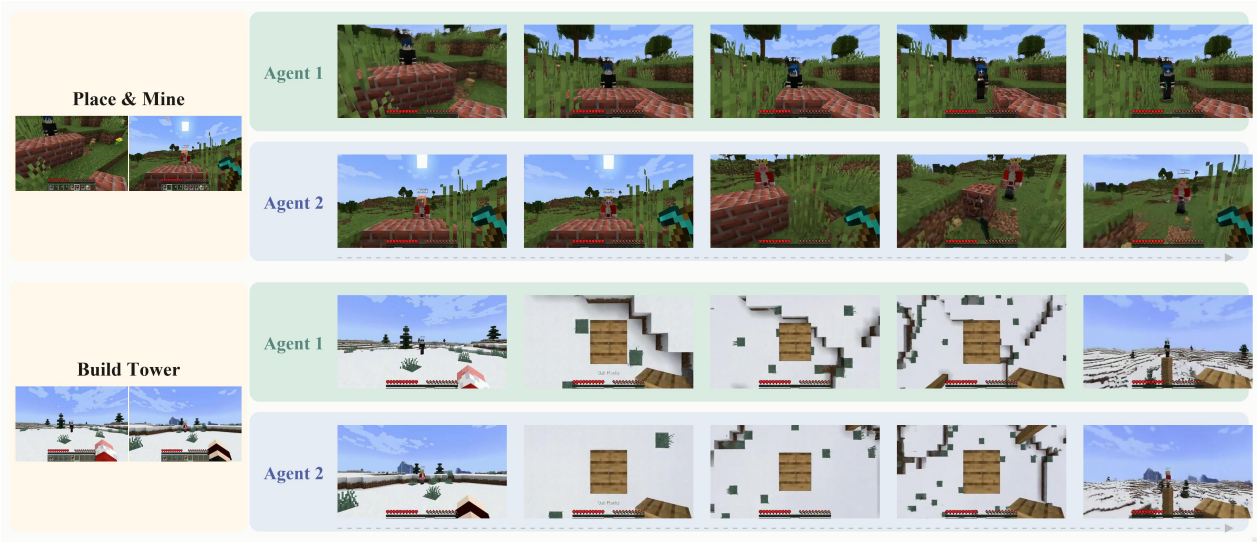

Figure 4: Qualitative examples of two-agent interaction. Each row shows a different task.

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 [9], which merges multiple agent views into a single visual stream. The second is Solaris [47], 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, which leads to more coherent interaction modeling and better preservation of each agent's 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 that 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. 3.2: 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.

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 virtual game environments, we also evaluate γ-World on real-world robotics coordination tasks. We use the RealOmin-Open Dataset [16], treating the left and right robot arms as two interacting agents—this lets the same multi-agent world-modeling framework that handles virtual players in Minecraft also capture coordinated bimanual manipulation in physical scenes. 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.

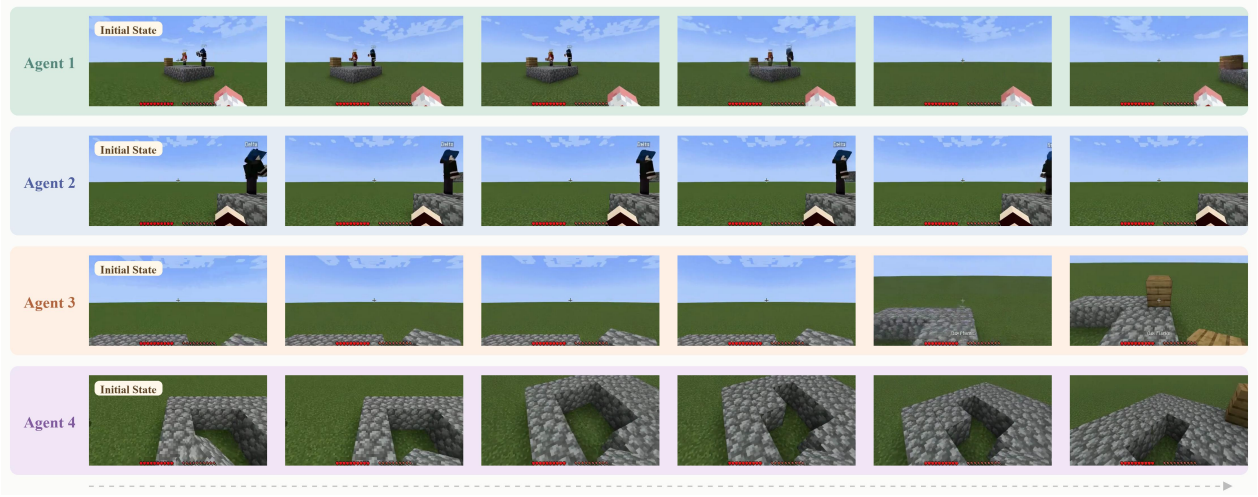

Figure 5: Qualitative example of scaling beyond two players. The first frame in each row shows the initial state for one agent, followed by synchronized rollouts across four agents.

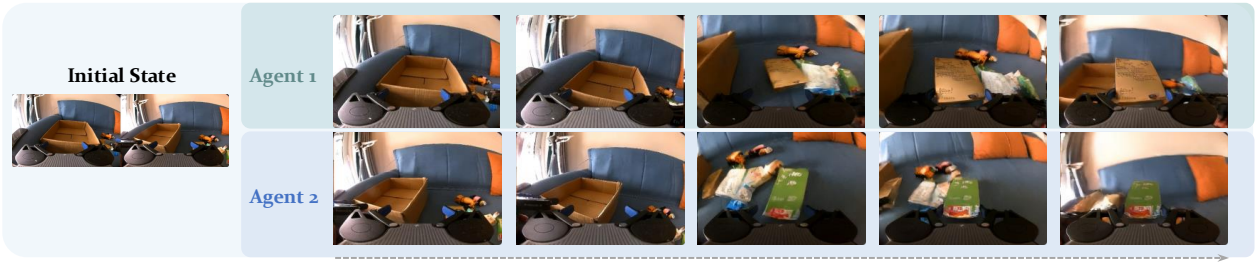

Figure 6: Qualitative examples of real-world robotic coordination.

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.

Limitations. 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.

A. 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.

B. 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

$$\bar{\mathbf{s}}_p = \mathbf{e}_p - \frac{1}{V}\mathbf{1}. \tag{14}$$

These vectors lie in the zero-mean subspace because

$$\mathbf{1}^\top \bar{\mathbf{s}}_p = 0. \tag{15}$$

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

$$\|\bar{\mathbf{s}}_p\|_2^2 = \left(\mathbf{e}_p - \frac{1}{V}\mathbf{1}\right)^\top \left(\mathbf{e}_p - \frac{1}{V}\mathbf{1}\right) \tag{16}$$

$$= 1 - \frac{2}{V} + \frac{1}{V} \tag{17}$$

$$= \frac{V-1}{V}. \tag{18}$$

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

$$\bar{\mathbf{s}}_p^\top \bar{\mathbf{s}}_q = \left(\mathbf{e}_p - \frac{1}{V}\mathbf{1}\right)^\top \left(\mathbf{e}_q - \frac{1}{V}\mathbf{1}\right) \tag{19}$$

$$= 0 - \frac{1}{V} - \frac{1}{V} + \frac{1}{V} \tag{20}$$

$$= -\frac{1}{V}. \tag{21}$$

After normalization, we obtain

$$\mathbf{s}_p = \sqrt{\frac{V}{V-1}}\bar{\mathbf{s}}_p. \tag{22}$$

Therefore,

$$\|\mathbf{s}_p\|_2^2 = 1, \tag{23}$$

$$\mathbf{s}_p^\top \mathbf{s}_q = -\frac{1}{V-1}, \qquad p \neq q. \tag{24}$$

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

$$\|\mathbf{s}_p - \mathbf{s}_q\|_2^2 = \|\mathbf{s}_p\|_2^2 + \|\mathbf{s}_q\|_2^2 - 2\mathbf{s}_p^\top \mathbf{s}_q \tag{25}$$

$$= 2 + \frac{2}{V-1} \tag{26}$$

$$= \frac{2V}{V-1}, \qquad p \neq q. \tag{27}$$

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

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

$$\boldsymbol{\theta}_p = \alpha \mathbf{s}_p, \tag{28}$$

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

$$\|\boldsymbol{\theta}_p - \boldsymbol{\theta}_q\|_2^2 = \alpha^2 \|\mathbf{s}_p - \mathbf{s}_q\|_2^2 \tag{29}$$

$$= \alpha^2 \frac{2V}{V-1}. \tag{30}$$

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

$$\boldsymbol{\Phi}_p = \exp(i\boldsymbol{\theta}_p). \tag{31}$$

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

$$\|\boldsymbol{\Phi}_p - \boldsymbol{\Phi}_q\|_2^2 = \|\exp(i\boldsymbol{\theta}_p) - \exp(i\boldsymbol{\theta}_q)\|_2^2 \tag{32}$$

$$= \sum_{r=1}^{d_p/2} \left| e^{i\theta_p^r} - e^{i\theta_q^r} \right|^2 \tag{33}$$

$$= \sum_{r=1}^{d_p/2} 2\left(1 - \cos(\theta_p^r - \theta_q^r)\right). \tag{34}$$

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,

$$\|\boldsymbol{\Phi}_p - \boldsymbol{\Phi}_q\|_2^2 \approx \sum_{r=1}^{d_p/2} (\theta_p^r - \theta_q^r)^2 \tag{35}$$

$$= \|\boldsymbol{\theta}_p - \boldsymbol{\theta}_q\|_2^2 \tag{36}$$

$$= \alpha^2 \frac{2V}{V-1}, \qquad p \neq q. \tag{37}$$

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.

C. More Architecture

C.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 inventory interaction,

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

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.

D. Additional Implementation Details

Architecture. Both bidirectional teacher model and causal student model are based on Cosmos-Predict2.5-2B [2], 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. 3.2), 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

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

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. 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 [23] to optimize the student model with the DMD [61] 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.

E. 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.

F. More Experimental Results

F.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

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

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.

References

[1] N. Agarwal, A. Ali, M. Bala, Y. Balaji, E. Barker, T. Cai, P. Chattopadhyay, Y. Chen, Y. Cui, Y. Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025. 3

[2] A. Ali, J. Bai, M. Bala, Y. Balaji, A. Blakeman, T. Cai, J. Cao, T. Cao, E. Cha, Y.-W. Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025. 2, 14

[3] E. Alonso, A. Jelley, V. Micheli, A. Kanervisto, A. Storkey, T. Pearce, and F. Fleuret. Diffusion for world modeling: Visual details matter in atari. *NeurIPS*, 37:58757–58791, 2024. 2, 3

[4] A. Bar, G. Zhou, D. Tran, T. Darrell, and Y. LeCun. Navigation world models. In *CVPR*, pages 15791–15801, 2025. 3

[5] J. Bruce, M. D. Dennis, A. Edwards, J. Parker-Holder, Y. Shi, E. Hughes, M. Lai, A. Mavalankar, R. Steigerwald, C. Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024. 2

[6] B. Chen, D. Martí Monsó, Y. Du, M. Simchowitz, R. Tedrake, and V. Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024. 3, 5, 7

[7] H. Chen, Y. Zhang, X. Cun, M. Xia, X. Wang, C. Weng, and Y. Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *CVPR*, pages 7310–7320, 2024. 3

[8] G. Deepmind. Veo3 video model, 2025. https://deepmind.google/models/veo/. 3

[9] Enigma-team. Introducing Multiverse: The first AI multiplayer world model. Enigma Blog, 2025. 8, 9

[10] L. Fan, G. Wang, Y. Jiang, A. Mandlekar, Y. Yang, H. Zhu, A. Tang, D.-A. Huang, Y. Zhu, and A. Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35:18343–18362, 2022. 3

[11] Z. Feng, R. Xue, L. Yuan, Y. Yu, N. Ding, M. Liu, B. Gao, J. Sun, X. Zheng, and G. Wang. Multi-agent embodied ai: Advances and future directions. *Science China Information Sciences*, 69(5):151202, 2026. 2

[12] K. Frans, D. Hafner, S. Levine, and P. Abbeel. One step diffusion via shortcut models. *arXiv preprint arXiv:2410.12557*, 2024. 3

[13] P. Fung, Y. Bachrach, A. Celikyilmaz, K. Chaudhuri, D. Chen, W. Chung, E. Dupoux, H. Gong, H. Jégou, A. Lazaric, et al. Embodied ai agents: Modeling the world. *arXiv preprint arXiv:2506.22355*, 2025. 2

[14] S. Gao, W. Liang, K. Zheng, A. Malik, S. Ye, S. Yu, W.-C. Tseng, Y. Dong, K. Mo, C.-H. Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026. 3

[15] Y. Gao, H. Guo, T. Hoang, W. Huang, L. Jiang, F. Kong, H. Li, J. Li, L. Li, X. Li, et al. Seedance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113*, 2025. 3

[16] Gen Robot. 10kh-realomin-opendata, 2025. 10

[17] Z. Geng, M. Deng, X. Bai, J. Z. Kolter, and K. He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025. 3

[18] Y. Guo, C. Yang, A. Rao, Z. Liang, Y. Wang, Y. Qiao, M. Agrawala, D. Lin, and B. Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *ICLR*, 2024. 3

[19] X. He, C. Peng, Z. Liu, B. Wang, Y. Zhang, Q. Cui, F. Kang, B. Jiang, M. An, Y. Ren, et al. Matrix-game 2.0: An open-source real-time and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025. 2, 3

[20] R. Henschel, L. Khachatryan, H. Poghosyan, D. Hayrapetyan, V. Tadevosyan, Z. Wang, S. Navasardyan, and H. Shi. Streamingt2v: Consistent, dynamic, and extendable long video generation from text. In *CVPR*, pages 2568–2577, 2025. 3

[21] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[22] A. Hu, L. Russell, H. Yeo, Z. Murez, G. Fedoseev, A. Kendall, J. Shotton, and G. Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023. 3

[23] X. Huang, Z. Li, G. He, M. Zhou, and E. Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025. 3, 5, 7, 15

[24] T. HunyuanWorld. Hy-world 1.5: A systematic framework for interactive world modeling with real-time latency and geometric consistency. *arXiv preprint*, 2025. 2, 3

[25] M. Kang, R. Zhang, C. Barnes, S. Paris, S. Kwak, J. Park, E. Shechtman, J.-Y. Zhu, and T. Park. Distilling diffusion models into conditional gans. In *ECCV*, pages 428–447, 2024. 3

[26] J. Kim, J. Kang, J. Choi, and B. Han. Fifo-diffusion: Generating infinite videos from text without training. *Advances in Neural Information Processing Systems*, 37:89834–89868, 2024. 3

[27] W. Kong, Q. Tian, Z. Zhang, R. Min, Z. Dai, J. Zhou, J. Xiong, X. Li, B. Wu, J. Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. 3

[28] Kuaishou. Kling video model, 2024. https://klingai.com/global/. 3

[29] C. Li, O. Michel, X. Pan, S. Liu, M. Roberts, and S. Xie. Pisa experiments: Exploring physics post-training for video diffusion models by watching stuff drop. *arXiv preprint arXiv:2503.09595*, 2025. 3

[30] C. Li, Y. Yang, J. Shao, H. Zhou, K. Schwarz, and Y. Liao. Rerope: Repurposing rope for relative camera control. *arXiv preprint arXiv:2602.08068*, 2026. 6

[31] L. Li, Q. Zhang, Y. Luo, S. Yang, R. Wang, F. Han, M. Yu, Z. Gao, N. Xue, X. Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026. 3

[32] S. Li, Y. Gao, D. Sadigh, and S. Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025. 3

[33] J. Liang, R. Liu, E. Ozguroglu, S. Sudhakar, A. Dave, P. Tokmakov, S. Song, and C. Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. *arXiv preprint arXiv:2406.16862*, 2024. 3

[34] S. Lin, A. Wang, and X. Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024. 3

[35] S. Lin, X. Xia, Y. Ren, C. Yang, X. Xiao, and L. Jiang. Diffusion adversarial post-training for one-step video generation. In *ICML*, 2025. 3

[36] S. Lin, C. Yang, H. He, J. Jiang, Y. Ren, X. Xia, Y. Zhao, X. Xiao, and L. Jiang. Autoregressive adversarial post-training for real-time interactive video generation. *arXiv preprint arXiv:2506.09350*, 2025. 3

[37] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. In *ICLR*, 2023. 3, 4

[38] Y. Lu, Y. Ren, X. Xia, S. Lin, X. Wang, X. Xiao, A. J. Ma, X. Xie, and J.-H. Lai. Adversarial distribution matching for diffusion distillation towards efficient image and video synthesis. In *ICCV*, pages 16818–16829, 2025. 3

[39] R. Mereu, A. Scannell, Y. Hou, Y. Zhao, A. Jitta, A. Dominguez, L. Acerbi, A. Storkey, and P. Chang. Generative world modelling for humanoids: 1x world model challenge technical report. *arXiv preprint arXiv:2510.07092*, 2025. 3

[40] OpenAI. Sora video model, 2024. https://sora.chatgpt.com/. 3

[41] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205, 2023. 3, 4

[42] X. Ren, Y. Lu, T. Cao, R. Gao, S. Huang, A. Sabour, T. Shen, T. Pfaff, J. Z. Wu, R. Chen, et al. Cosmos-drive-dreams: Scalable synthetic driving data generation with world foundation models. *arXiv preprint arXiv:2506.09042*, 2025. 2

[43] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3

[44] T. Salimans and J. Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 3

[45] A. Sauer, F. Boesel, T. Dockhorn, A. Blattmann, P. Esser, and R. Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In *SIGGRAPH Asia*, pages 1–11, 2024. 3

[46] A. Sauer, D. Lorenz, A. Blattmann, and R. Rombach. Adversarial diffusion distillation. In *ECCV*, pages 87–103, 2024. 3

[47] G. Savva, O. Michel, D. Lu, S. Waiwitlikhit, T. Meehan, D. Mishra, S. Poddar, J. Lu, and S. Xie. Solaris: Building a multiplayer video world model in minecraft. *arXiv preprint arXiv:2602.22208*, 2026. 2, 6, 8, 9

[48] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. 3

[49] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 5

[50] W. Sun, H. Zhang, H. Wang, J. Wu, Z. Wang, Z. Wang, Y. Wang, J. Zhang, T. Wang, and C. Guo. Worldplay: Towards long-term geometric consistency for real-time interactive world model. *arXiv preprint*, 2025. 2, 3

[51] D. Valevski, Y. Leviathan, M. Arar, and S. Fruchter. Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837*, 2024. 2, 3

[52] T. Wan, A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 3, 4

[53] Z. Wang, C. Lu, Y. Wang, F. Bao, C. Li, H. Su, and J. Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *NeurIPS*, 36:8406–8441, 2023. 3

[54] T. Wiedemer, Y. Li, P. Vicol, S. S. Gu, N. Matarese, K. Swersky, B. Kim, P. Jaini, and R. Geirhos. Video models are zero-shot learners and reasoners. *arXiv preprint arXiv:2509.20328*, 2025. 3

[55] Z. Xiao, Y. Lan, Y. Zhou, W. Ouyang, S. Yang, Y. Zeng, and X. Pan. Worldmem: Long-term consistent world simulation with memory. *arXiv preprint arXiv:2504.12369*, 2025. 2, 3

[56] M. Yang, Y. Du, K. Ghasemipour, J. Tompson, D. Schuurmans, and P. Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023. 3

[57] Z. Yang, Y. Chen, J. Wang, S. Manivasagam, W.-C. Ma, A. J. Yang, and R. Urtasun. Unisim: A neural closed-loop sensor simulator. In *CVPR*, pages 1389–1399, 2023. 3

[58] Z. Yang, J. Teng, W. Zheng, M. Ding, S. Huang, J. Xu, Y. Yang, W. Hong, X. Zhang, G. Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *ICLR*, 2024. 3

[59] S. Ye, Y. Ge, K. Zheng, S. Gao, S. Yu, G. Kurian, S. Indupuru, Y. L. Tan, C. Zhu, J. Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026. 3

[60] T. Yin, M. Gharbi, T. Park, R. Zhang, E. Shechtman, F. Durand, and B. Freeman. Improved distribution matching distillation for fast image synthesis. *NeurIPS*, 37:47455–47487, 2024. 3

[61] T. Yin, M. Gharbi, R. Zhang, E. Shechtman, F. Durand, W. T. Freeman, and T. Park. One-step diffusion with distribution matching distillation. In *CVPR*, pages 6613–6623, 2024. 3, 8, 15

[62] T. Yin, Q. Zhang, R. Zhang, W. T. Freeman, F. Durand, E. Shechtman, and X. Huang. From slow bidirectional to fast autoregressive video diffusion models. In *CVPR*, pages 22963–22974, 2025. 3, 7

[63] J. Yu, J. Bai, Y. Qin, Q. Liu, X. Wang, P. Wan, D. Zhang, and X. Liu. Context as memory: Scene-consistent interactive long video generation with memory retrieval. *arXiv preprint arXiv:2506.03141*, 2025. 2, 3

[64] J. Yuan, X. Zhang, F. Friedrich, N. Beltran-Velez, M. Hall, R. Askari-Hemmat, X. Han, N. Ballas, M. Drozdzal, and A. Romero-Soriano. Inference-time physics alignment of video generative models with latent world models. *arXiv preprint arXiv:2601.10553*, 2026. 3