OVERSEER: MULTI-AGENT WORLD MODELS BEYOND WHAT AGENTS SEE

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite the success of video world models, extending them to multiple agents is difficult because it requires generating several agent-specific video streams that are consistent in time and between agents. To address this gap, we introduce *OverSeer*, a multi-agent world model that augments egocentric agent views with auxiliary camera views. We suggest that, by fixing the auxiliary views in space, they act as a proxy for a world-centric representation: Unlike the agents' views, which can change entirely due to a simple change in agent position, changes in the auxiliary views directly map to localized changes in the world state. We show that these fixed views substantially help the model to maintain consistency. We also show that, due to their redundant and semi-static nature, the auxiliary views can be efficiently compressed with a δ-VAE that encodes only the changes relative to a static reference image, reducing the number of tokens required to process them. Experiments in Minecraft and Doom demonstrate improved world and agent consistency, with qualitative results on real-world multi-camera captures indicating that the approach is not tied to game engines. Our compression analysis further shows that δ-VAE substantially reduces the computational overhead of auxiliary-view generation. We conclude that these fixed views provide a simple and efficient way to improve multi-agent world models. Videos and supplementary materials are available on our website: https://overseer-wm.github.io/



Figure 1: **OverSeer sees beyond what agents see.** Two agents interact in a shared Minecraft environment (Left). In addition to generating each agent's egocentric view (Bottom), OverSeer models six fixed-pose auxiliary cameras (Right). These auxiliary streams establish a persistent shared reference, enforcing world consistency over time and agent consistency across views.

1 INTRODUCTION

World models aim to learn predictive representations of environments. Video world models represent the environment as a sequence of images captured from the viewpoint of an acting agent, predicting future frames conditioned on past observations and agent actions. By taking advantage of large video datasets, these models can learn to represent complex visual dynamics and interactions between characters and objects. Recent video world models have made substantial progress in visual fidelity and character controllability, but maintaining a coherent world over long horizons

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.

remains challenging. Small prediction errors can accumulate over time, causing objects to drift and environments to change inconsistently.

These challenges become more pronounced in multi-agent environments. A multi-agent world model must represent not only how the environment evolves but also how multiple characters interact within the same shared world. We identify two forms of consistency that are essential for this setting. First, *world consistency* requires persistent scene elements and the effects of past events to be preserved over time: when an agent revisits a location, the scene should reflect its history. Second, *agent consistency* requires the actions and states of different agents to agree across their observations: an interaction should be depicted consistently in all views from which it is visible. These constraints are difficult to satisfy using only player-centric streams, because each player observes only a *partial and changing projection of the underlying world*.

One way to address these challenges is to use a *world-centric representation* that explicitly maintains a shared state independent of any individual agent's viewpoint. Such representations can also model dynamics more efficiently: localized updates occur only when the environment actually changes, whereas an agent's camera motion drastically alters egocentric pixels even in a completely static scene. However, explicitly reconstructing and maintaining a world-centric representation from partial observations remains challenging.

In this paper, we propose a video-based framework that captures the key advantages of a world-centric representation without requiring explicit global state reconstruction. To this end, we introduce *OverSeer*, a multi-agent video world model augmented with *fixed-pose auxiliary camera views* that serve as a practical proxy for a shared world-centric state. At inference, the auxiliary views are automatically, jointly, and efficiently generated with the agents' views through shared attention. This design has three benefits: (i) Auxiliary views cover regions beyond individual agents' immediate fields of view, establishing a shared visual reference that grounds cross-agent interactions; (ii) because the auxiliary cameras are stationary, any temporal variation in their views reflects actual environmental changes rather than camera ego-motion; and (iii) fixed viewpoints make temporal redundancies predictable and easy to model, reinforcing structural coherence across streams.

To minimize computational overhead and exploit this inherent temporal redundancy, we introduce δ-VAE, a variational autoencoder that performs *second-stage latent compression* on top of a standard video VAE. For each auxiliary stream, δ-VAE retains an initial reference frame latent and encodes subsequent frames into compact residual latents conditioned on that reference. The static reference frame preserves persistent background geometry, enabling the residual latents to concentrate exclusively on dynamic events, such as agent movements, object manipulations, and environmental state changes. By eliminating redundant encoding of unchanged background content, six persistent auxiliary views incur only 37.5% of the token overhead of the two agent streams.

We evaluate OverSeer on Minecraft and Doom, two games with substantially different visual dynamics, spatial structures, and interaction patterns, and additionally show qualitative results on real-world multi-camera captures (Shapovalov et al., 2023). When agents lose sight of each other and later reunite, existing multi-agent world models restore the partner or the scene correctly in only 0–12.5% of episodes, whereas OverSeer does so in 38–60%, at a small fraction of the cost of modeling the auxiliary views as full video streams.

2 Related Work

Video Diffusion and Video World Models. Diffusion models have emerged as the dominant framework for video generation, evolving from early techniques that augmented pretrained 2D priors with temporal layers (Singer et al., 2023; Blattmann et al., 2023) to large-scale architectures scaled for high-fidelity synthesis (OpenAI, 2024; Google DeepMind, 2025; HaCohen et al., 2025; Wan et al., 2025). Video world models extend this paradigm by casting environment simulation as action-conditioned video diffusion, where future frame generation is guided by past observations and agent actions to yield interactive simulators (Bruce et al., 2024; Valevski et al., 2025). Recent full-stack systems further enrich this formulation by integrating explicit camera control, scene memory, and real-time streaming (DreamX et al., 2026; Jiang et al., 2026; Wang et al., 2026).

However, progress in this domain mainly focuses on expanding the input conditioning space, while output generation remains restricted to rendering a single egocentric stream for a single agent. In contrast, OverSeer shifts focus to the model's output space: given initial frames and actions from

all agents, the model jointly predicts every agent's egocentric stream *alongside* the fixed auxiliary camera views, improving both *world and agent consistency*.

Long-Horizon Consistency and Memory. Autoregressive rollouts in video world models tend to compound error and spatial drift over extended horizons (Chen et al., 2024; Huang et al., 2025). The prevailing remedy is expanding context capacity through positional extrapolation or streaming attention (Yang et al., 2026; Yesiltepe et al., 2026), memory-bank retrieval by estimated camera pose (Yu et al., 2025a; Hong et al., 2025; Yu et al., 2026), and geometry-guided retrieval (DreamX et al., 2026; Li et al., 2025). However, these memory structures are constructed entirely from an agent's own past observations. Consequently, unobserved regions are never stored, and retrieval accuracy remains bound to pose estimates that drift alongside generation. Our pose-fixed auxiliary views serve a complementary function: while traditional memory recalls previously observed regions, auxiliary streams provide active coverage of unobserved space. Furthermore, because these auxiliary views are jointly predicted rather than retrieved, they eliminate reliance on fragile pose estimation altogether.

Multi-Agent and Multi-View World Models. Multi-agent world models were first studied in reinforcement learning with compact state spaces (Egorov & Shpilman, 2022; Zhang et al., 2025), where multi-view consistency is enforced within an unrendered latent state. Recent work extends this requirement to the pixel domain, conditioning video diffusion models on multiple action streams to synthesize concurrent agent views (Enigma, 2025; Savva et al., 2026; Liu et al., 2026a; Hu et al., 2026a; Wu et al., 2026; Hu et al., 2026b). Multi-view generation also appears in generative 3D modeling and autonomous driving (Kong et al., 2024; Liu et al., 2026b; Jang et al., 2026; Gao et al., 2024a;b;c), though cameras in these settings are typically in a static scene or camera rigs rigidly attached to a single moving platform.

Existing approaches to multi-agent visual consistency broadly fall into two categories. The first couples egocentric agent streams directly via stream concatenation, shared attention, tiling, or permutation-symmetric agent encodings (Enigma, 2025; Savva et al., 2026; Hu et al., 2026a; Liu et al., 2026a), meaning every reference frame is itself a moving, partial observation. The second improves consistency via an external state, such as an editable map with estimated per-agent poses (Po et al., 2026); while this decouples the reference from individual agent trajectories, it models spatial geometry rather than visual appearance and remains vulnerable to pose drift.

In contrast, OverSeer improves consistency by introducing auxiliary streams that are jointly *predicted* alongside the agents' streams rather than supplied as external map inputs.

Efficient Video Tokenization. Modern video generators typically operate within the latent space of a video VAE (Wan et al., 2025; HaCohen et al., 2025), with recent efficiency efforts compressing representations further via stronger tokenizers (Yu et al., 2024; Chen et al., 2025), token merging (Bolya et al., 2023), lightweight decoders, and bounded KV caches (Jiang et al., 2026; Hu et al., 2026a). However, these generic techniques allocate computational capacity uniformly across all streams, causing an auxiliary camera view to incur nearly the same computational cost as an egocentric agent stream.

A closer line of work changes what a latent space *represents*: neural video codecs encode frames relative to a reference frame via residuals (Lu et al., 2019) or by conditioning (Li et al., 2021), RefTok tokenizes each frame relative to a reference frame (Fan et al., 2025), and DeltaTok compresses consecutive-frame feature differences into a single delta token (Kerssies et al., 2026). In these approaches, the reference frame is not fixed, but updated regularly over time, missing the advantages of OverSeer's fixed views, starting from the fact that they can be compressed effectively.

3 OverSeer

3.1 Background and Notation

Video World Modeling. Video world models usually represent the evolution of the world from the viewpoint of an agent. Let o_t denote the visual observation at time t (i.e., an image), a_t the agent action, and $z_t = \mathcal{E}(o_t)$ a learned representation of the observation, where $\mathcal{E}$ denotes a visual encoder. Let s_t denote the world state available to the model at time t. A general one-step video world model can be formulated as:

$$p(z_{t+1} \mid s_t, a_t) \tag{1}$$

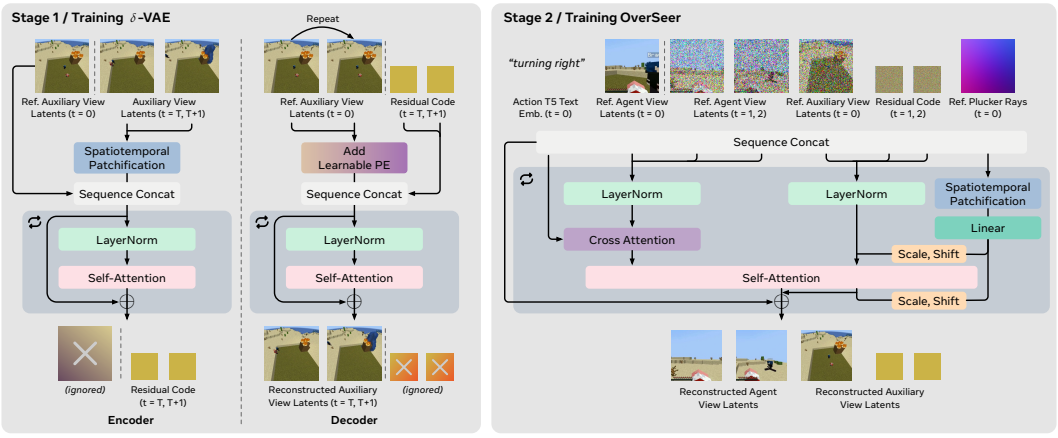


Figure 2: **OverSeer design overview.** A δ-VAE (Left) compresses each fixed auxiliary stream into compact residual codes relative to its first-frame reference, and OverSeer (Right) jointly generates agent latents and these residual codes, conditioned on T5-encoded actions via cross-attention and Plücker-ray camera poses via AdaLN.

Depending on the model, s_t may correspond to the current representation z_t, a history of representations $z_{\leq t}$, or a learned summary of the observation-action history $f(z_{\leq t}, a_{<t})$.

Multi-Agent Video World Modeling. Consider an environment with N agents. We use the superscript A to denote agent-centric camera streams. Agent i receives an observation $o_t^{A,i}$, represented as $z_t^{A,i} = \mathcal{E}(o_t^{A,i})$, and executes an action a_t^i. We denote the joint agent observations, representations, and actions by $\mathbf{o}_t^A = \{o_t^{A,i}\}_{i=1}^N$, $\mathbf{z}_t^A = \{z_t^{A,i}\}_{i=1}^N$, and $\mathbf{a}_t = \{a_t^i\}_{i=1}^N$. A multi-agent video world model extends eq. (1) to joint prediction:

$$p(\mathbf{z}_{t+1}^A \mid \mathbf{s}_t, \mathbf{a}_t), \tag{2}$$

where $\mathbf{s}_t$ represents the shared world state. Factorizing this into independent single-agent models, $\prod_i p(z_{t+1}^{A,i} \mid s_t^i, a_t^i)$, is insufficient: each s_t^i is built only from agent i's own history, so nothing enforces that agents agree where their views overlap, or that a region seen by one agent stays consistent when another later observes it. The challenge is that $\mathbf{s}_t$ must be inferred from $\mathbf{z}_{\leq t}^A$, which are partial and change as the agents move.

Auxiliary Camera Views. To provide persistent spatial references for the shared environment, OverSeer introduces K fixed-pose auxiliary camera views, denoted by $\mathbf{o}_t^X = \{o_t^{X,j}\}_{j=1}^K$, with corresponding camera poses $\mathbf{c} = \{c^j\}_{j=1}^K$, where the superscript X denotes auxiliary camera streams. Unlike agent-centric cameras, whose viewpoints change with agent motion, the auxiliary cameras remain fixed over time. These views change proportionally to changes in the underlying world's state and provide a stable spatial reference for all the agents. During training, the auxiliary views also teach the model what happens outside the agents' immediate purview.

3.2 OverSeer Model Design

We design OverSeer as a latent video diffusion model (Rombach et al., 2022) built upon Wan2.2-TI2V-5B (Wan et al., 2025), leveraging its strong pre-trained video generation prior. An overview is shown in fig. 2. Following the standard latent diffusion design, input visual observations are mapped into a compact latent space using a pre-trained video VAE (Wan et al., 2025), yielding agent latents $\mathbf{z}_t^A = \{\mathcal{E}(o); \forall o \in \mathbf{o}_t^A\}$ and auxiliary latents $\mathbf{z}_t^X = \{\mathcal{E}(o); \forall o \in \mathbf{o}_t^X\}$.

To keep the latent sequence lengths tractable during diffusion, we do not model auxiliary latents frame-by-frame. Instead, each auxiliary stream is represented by its first-frame latent as a fixed anchor, $\mathbf{z}_0^X = \{z_0^{X,j}\}_{j=1}^K$, paired with compact residual codes $\mathbf{d}_{1:Q}^X = \{d_{1:Q}^{X,j}\}_{j=1}^K$, a length-$Q$ sequence per view that encodes subsequent temporal dynamics produced by δ-VAE (see section 3.3). Extending the world-model formulation in eq. (1) to multi-agent generation with auxiliary views, OverSeer jointly models the future agent latents, initial auxiliary anchor frames, and compressed



Figure 3: **OverSeer keeps the agents consistent across views.** Generated videos in Minecraft, GZDoom and a real multi-camera capture, each at two moments (before, after). Each agent's trail is one 3D track projected into every auxiliary camera (Agent 1 blue, Agent 2 orange); it lands on the generated agent in each of them, so the agents are consistent across views.

auxiliary dynamics, conditioned on the initial agent frames and actions, and auxiliary camera poses:

$$p\left(\mathbf{z}^A_{1:T}, \mathbf{z}^X_0, \mathbf{d}^X_{1:T} \mid \mathbf{z}^A_0, \mathbf{a}_{0:T-1}, \mathbf{c}\right). \tag{3}$$

Training and Inference. Knowledge of the ground-truth auxiliary visual streams is only required during training. At inference, OverSeer receives the initial agent observations and agent actions along with auxiliary camera poses. The auxiliary camera poses are sampled from a *scene-independent prior* estimated from training poses and greedily selected to maximize field-of-view coverage without scene-specific input (see section B). The model jointly generates future agent observations and the auxiliary streams, including their initial anchor frames. Ground-truth auxiliary images are used only during training.

We train OverSeer using diffusion forcing (Chen et al., 2024), where diffusion timesteps are sampled independently across temporal positions, and target latents are noised accordingly while initial conditioning frames $\mathbf{z}^A_0$ remain clean. All latents, including agent latents, auxiliary anchor frames, and residual codes, are projected to the same DiT hidden dimension with learnable view-identification embeddings added to each view's tokens for differentiation.

Agent and auxiliary streams are governed by different modalities and therefore use complementary conditioning pathways. For each agent stream, discrete actions (including agent navigation, camera motions, and inventory interactions) are formatted into short textual descriptions, encoded via a frozen T5 text encoder (Raffel et al., 2020), and injected into the DiT layers through cross-attention. For each auxiliary stream, camera pose c^j and its known intrinsics are converted into a spatial Plücker-ray map (Sitzmann et al., 2021). These ray representations are patchified and projected into Adaptive Layer Normalization (AdaLN) (Perez et al., 2018), with scale and shift parameters to modulate the corresponding auxiliary tokens.

Tokens from all agents and auxiliary streams are then jointly processed by shared self-attention layers, enabling direct cross-modality communication. We enforce a causal temporal attention mask throughout the network: tokens at a given temporal position t can attend only to conditioning tokens and to tokens from identical or earlier temporal positions ($t' \leq t$). This prevents future latent states from leaking into the prediction of earlier states under independent diffusion forcing timesteps.

3.3 Efficient Modeling of Auxiliary Dynamics

Because the auxiliary cameras are pose-fixed, their background visual content remains predominantly static across time, with scene updates concentrated in localized regions affected by agent activity and environmental dynamics. Directly modeling full latent representations for every auxiliary frame therefore introduces severe temporal and spatial redundancy. To address this inefficiency, we introduce δ-VAE, a second-stage variational autoencoder that exploits this background stationarity to further compress the latent streams produced by the primary video VAE.

Anchor and Residual Codes. For each auxiliary view j, we retain the initial video-VAE latent $z_0^{X,j} \in \mathbb{R}^{H\times W\times C}$ as a static reference anchor and partition the subsequent temporal sequence into Q non-overlapping chunks. Let $\mathcal{T}_q$ denote the time indices of the target latents in chunk q, where $|\mathcal{T}_q| = \tau$ represents the temporal compression factor. For each chunk q, δ-VAE encodes the anchor $z_0^{X,j}$ alongside the corresponding target latents $\{z_t^{X,j}\}_{t\in\mathcal{T}_q}$ into a compact residual code $d_q^{X,j} \in \mathbb{R}^{\frac{H}{s}\times\frac{W}{s}\times C_d}$, where s denotes the spatial compression factor. These residual codes achieve an $s^2\times$ spatial and $\tau\times$ temporal token reduction relative to the primary video-VAE latents. The resulting compressed sequence for auxiliary view j is represented as $\widetilde{z}^{X,j} = [z_0^{X,j}; d_1^{X,j}; \cdots; d_Q^{X,j}]$.

δ-VAE Architecture Design. The encoder $\mathcal{E}_{\delta\text{-VAE}}$ spatially patchifies the video-VAE features and jointly processes the anchor latent $z_0^{X,j}$ and the target latents $\{z_t^{X,j}\}_{t\in\mathcal{T}_q}$ using a spatio-temporal transformer. It parameterizes a Gaussian latent distribution $\mathcal{G}_q^{X,j}$ (Kingma & Welling, 2013) from which the residual code $d_q^{X,j}$ is sampled. The decoder $\mathcal{D}_{\delta\text{-VAE}}$ reconstructs each target chunk from the residual code $d_q^{X,j}$ and the anchor $z_0^{X,j}$. The residual code remains at its compressed spatial resolution, while the anchor is repeated over the τ target temporal positions by adding a learnable temporal positional embedding. These replicated anchor features then attend to $d_q^{X,j}$ through cross-attention layers, producing the reconstructed target latents $\{\hat{z}_t^{X,j}\}_{t\in\mathcal{T}_q}$.

Training Objective. We train the δ-VAE with three losses: an ℓ_1 reconstruction loss $\mathcal{L}_{\text{rec}}$ between the reconstructed and target video-VAE latents, the standard VAE KL divergence $\mathcal{L}_{\text{KL}}$, and a feature alignment loss $\mathcal{L}_{\text{align}}$ inspired by REPA (Yu et al., 2025b). Reconstruction alone lets the residual code spend its capacity on static background that the anchor already provides; $\mathcal{L}_{\text{align}}$ instead aligns the code with video-VAE features re-weighted toward dynamic regions. To build this alignment target, we first compute a spatial change weight map for each target frame from its channel-wise latent distance to the anchor,

$$\widetilde{M}_t^{X,j} = \left\| z_t^{X,j} - z_0^{X,j} \right\|_{2,C}, \qquad M_t^{X,j} = \frac{\widetilde{M}_t^{X,j}}{\left\| \widetilde{M}_t^{X,j} \right\|_2}, \tag{4}$$

where $\|\cdot\|_{2,C}$ denotes the ℓ_2 norm over channels and $M_t^{X,j} \in \mathbb{R}^{H\times W}$ assigns higher weights to spatial locations exhibiting greater temporal variation relative to the anchor frame. We modulate the target video-VAE features using this weight map via $r_t^{X,j} = M_t^{X,j} \odot z_t^{X,j}$. Importantly, the latent distance is used solely to isolate dynamic regions; the feature alignment target is constructed directly from the target VAE features themselves, as naive subtraction within a non-linear VAE latent space does not generally yield semantically meaningful representations.

For each chunk, the weighted feature $\{r_t^{X,j}\}_{t\in\mathcal{T}_q} \in \mathbb{R}^{\tau\times H\times W\times C}$ is pooled over $\tau \times s \times s$ temporal-spatial blocks, yielding a target feature representation in $\mathbb{R}^{\frac{H}{s}\times\frac{W}{s}\times C}$. A learned linear projection $\mathcal{P}$ maps $d_q^{X,j}$ from C_d to C, establishing the feature alignment loss. Combining this with the ℓ_1 reconstruction loss $\mathcal{L}_{\text{rec}}$ and the KL divergence regularization $\mathcal{L}_{\text{KL}}$, the loss terms are defined as:

$$\mathcal{L}_{\text{rec}} = \|\hat{z}_t^{X,j} - z_t^{X,j}\|_1,\ \mathcal{L}_{\text{KL}} = \text{KL}\big(\mathcal{G}_q^{X,j} \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})\big),\ \mathcal{L}_{\text{align}} = \left\| \mathcal{P}\left(d_q^{X,j}\right) - \text{Pool}\left(\{r_t^{X,j}\}_{t\in\mathcal{T}_q}\right) \right\|_2^2. \tag{5}$$

The overall training objective for δ-VAE is: $\mathcal{L}_{\delta\text{-VAE}} = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}}$ with loss weights specified in section D. We first train the δ-VAE individually on auxiliary-camera streams and freeze its weights prior to training the primary world model OverSeer. During OverSeer training, ground-truth auxiliary videos are encoded into anchor latents $\mathbf{z}_0^X$ and residual codes $\mathbf{d}_{1:Q}^X$, which serve as diffusion targets alongside future agent latents $\mathbf{z}_{1:T}^A$ in eq. (3).

4 Experiments

We structure our experimental evaluation to test the central hypothesis of this work: extending a multi-agent video world model with pose-fixed, efficiently represented auxiliary views significantly improves both temporal and agent consistency compared to baselines (section 4.1). We then conduct ablation studies investigating how performance scales with the number of auxiliary views and how the δ-VAE compression ratio influences synthesis quality and computational overhead (section 4.2). For qualitative results, please refer to our website and appendix.

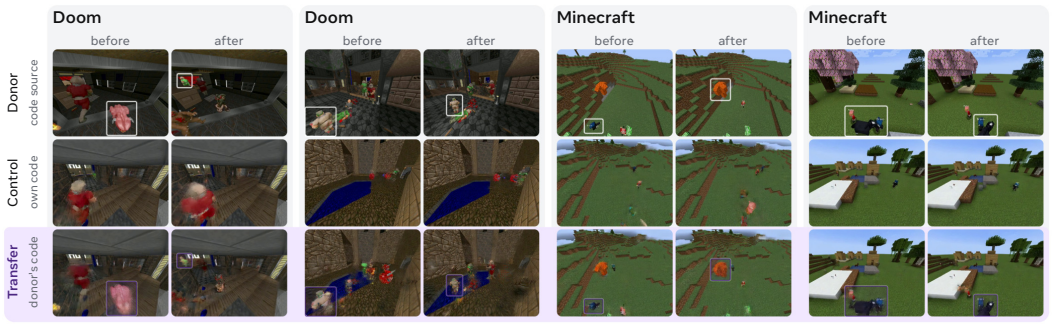


Figure 4: **δ-VAE residual codes carry the movers, not the scene.** Four cross-scene transfers: a host anchor is decoded with a donor clip's residual code (transfer, purple band) instead of its own (control), at two moments (before, after). The donor's mover (white box) appears in the host scene (purple box) while the host scene itself is preserved.

Environments and Benchmarks. We evaluate world models by conditioning future frame rollouts on initial visual observations and multi-agent action trajectories. Primary quantitative evaluations are conducted in **Minecraft**, whose persistent and editable 3D world naturally tests spatial memory, multi-agent interaction, and cross-view consistency (tables 1 and 2). To evaluate long-horizon spatial reasoning, we additionally run *revisit* tasks in **Doom** (table 2) and showcase qualitative generalization on real-world video captures from **Replay** (Shapovalov et al., 2023). In Minecraft, evaluation spans two benchmark suites:

1. **Solaris Benchmark** (Savva et al., 2026): Evaluates five multi-agent capabilities using VLM-judged scripted questions, requiring all questions within an episode to be answered correctly.
2. **MAC-Bench (Multi-agent Consistency Benchmark)**: Designed specifically to evaluate world and agent consistency in a harder setting. Specifically. we design the following three tasks:
   - *Reunite at Start*: Both agents return to their exact initial poses. Consequently, an agent's partner must reappear at a previously observed spatial location.
   - *Reunite Far*: Agents navigate long, converging routes without visual contact in transit, turning to face each other upon arrival. A partner must appear at close range within previously unobserved space.
   - *Dynamic Event*: Agents observe the initial stage of a localized dynamic process (e.g., a wooden structure catching fire) before separating. Upon return, the object state must reflect correct off-screen temporal evolution.

In Doom, we design two *Reunite* tasks across six distinct maps with extended navigation paths.

Evaluation Metrics. We report VLM Episode Accuracy (VLM $\uparrow$) and Frame Fréchet Inception Distance (FID $\downarrow$). VLM accuracy is evaluated using Claude Opus 5 conditioned on ground-truth camera poses and environment event logs. Metrics are averaged across 32 episodes per task in Minecraft and 100 episodes for each Doom task.

Baselines. We compare OverSeer against two state-of-the-art two-player world models for Minecraft: Solaris (Savva et al., 2026) and γ-World (Liu et al., 2026a). For a fair comparison, all models are evaluated with their non-distilled, causally fine-tuned checkpoints and receive identical conditioning frames and action sequences, operating strictly without auxiliary views at inference.

4.1 Analysis of Multi-Agent World Modeling Benchmarks

Main results. As shown in table 1, OverSeer leads on grounding, memory, and consistency, the Solaris capabilities that require spatial reasoning beyond the current view. Conversely, it trails the best baseline on building and movement tasks, which depend primarily on short-horizon egocentric streams. On MAC-Bench, OverSeer yields substantial gains in VLM accuracy over prior baselines, increasing scores from $0.0\%/6.2\%/12.5\%$ to $60.2\%/37.9\%/40.5\%$ across our three designed tasks (also qualitatively shown in fig. 7), while achieving the lowest (best) FID score across all tasks. Agent consistency is further demonstrated qualitatively in fig. 3: projected 3D agent trajectories precisely align with generated agents across every auxiliary view in Minecraft, Doom, and real-world captures. Extended visualization rollouts are provided in section G.

	Solaris Benchmark										MAC-Bench					
	Movement		Grounding		Memory		Building		Consistency		Reunite at Start		Reunite Far		Dynamic Event	
Method	**VLM↑**	**FID↓**	**VLM↑**	**FID↓**	**VLM↑**	**FID↓**	**VLM↑**	**FID↓**	**VLM↑**	**FID↓**	**VLM↑**	**FID↓**	**VLM↑**	**FID↓**	**VLM↑**	**FID↓**
Solaris	**68.8**	22.38	66.7	14.92	20.8	47.11	22.9	76.01	56.3	68.42	0.0	17.06	6.2	20.23	12.5	26.52
γ-World	64.6	**17.13**	62.5	29.39	16.7	29.57	**43.8**	**24.62**	47.9	56.39	0.0	23.83	0.0	25.71	8.3	31.72
OverSeer	59.1	18.27	**91.7**	**11.33**	**87.5**	**12.95**	41.7	32.50	**60.4**	**26.47**	**60.2**	**13.99**	**37.9**	**12.79**	**40.5**	**17.61**

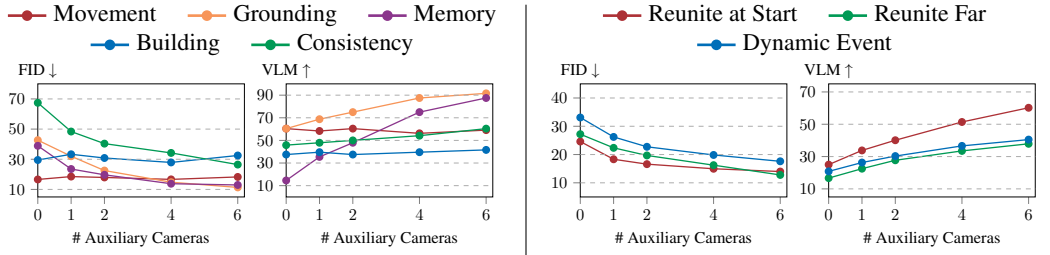
Table 1: **Results on Minecraft.** Comparison on the Solaris benchmark and our MAC-Bench. VLM denotes VLM-judged accuracy (%) and FID denotes frame-level FID; MAC-Bench results are averaged over 5 runs. **Bold** and underline denote the best and second-best results in each column, respectively.

	Minecraft						Doom			
	Reunite at Start		Reunite Far		Dynamic Event		Reunite at Start		Reunite Far	
Variant	**VLM↑**	**FID↓**	**VLM↑**	**FID↓**	**VLM↑**	**FID↓**	**VLM↑**	**FID↓**	**VLM↑**	**FID↓**
OverSeer	**60.2**	14.0	37.9	**12.8**	40.5	17.6	62.0	23.2	67.0	20.9
w/o Aux. Views	25.0	24.6	16.7	27.2	20.8	33.1	12.0	50.6	0.0	87.4
w/o δ-VAE	59.4	**8.2**	**40.6**	13.4	**41.7**	**10.3**	**78.0**	**15.8**	**84.0**	**15.4**

Table 2: **Ablation of auxiliary views and δ-VAE.** Comparison on MAC-Bench in Minecraft and Doom. w/o δ-VAE models auxiliary views with uncompressed video-VAE latents, using 8× more auxiliary tokens. **Bold** denotes the best result in each column.

FID does not reflect generation consistency. On *Reunite at Start*, Solaris obtains an FID of 17.06, within 3.1 points of OverSeer, yet answers none of the episodes correctly. While its generated rollouts remain *visually in-distribution*, they fail to reconstruct the specific scene previously left behind by the agents, frequently filling in existing craters or omitting partner agents entirely (fig. 7). This disconnect demonstrates that standard frame-level distributional metrics like FID are insensitive to structural memory failures, motivating the content-level queries proposed in our new benchmark.

Without a shared view, agents drift apart. Both baselines enforce cross-agent agreement by coupling the agent streams directly, through concatenation or shared attention. Under this design, every reference frame is itself a dynamic, partial observation, restricting agreement exclusively to moments when the agents' fields of view overlap. Empirical results directly reflect this structural limitation: once agents lose visual contact, γ-World does not achieve any of the reunite tasks. The same argument accounts for the gap between *Reunite at Start* (60.2) and *Reunite Far* (37.9), as sampled cameras cover the start region more densely than the periphery.



(a) Solaris Benchmark (b) MAC-Bench

Figure 5: **Effect of auxiliary camera count.** Adding auxiliary views systematically improves FID and VLM accuracies on consistency and memory metrics across both benchmarks, with diminishing returns once spatial coverage saturates.

4.2 Ablations

Effect of δ-VAE. Removing δ-VAE improves visual quality and, on Doom, consistency as well (table 2), but at the cost of an 8× larger auxiliary-stream token footprint (864 vs. 108 tokens per frame at 576 × 384). This performance trade-off is domain-dependent: a residual code remains faithful only insofar as the scene stays close to its static anchor. Doom's rapid lighting changes and enemy movements generate much larger residual updates than Minecraft's slow, blocky dynamics.

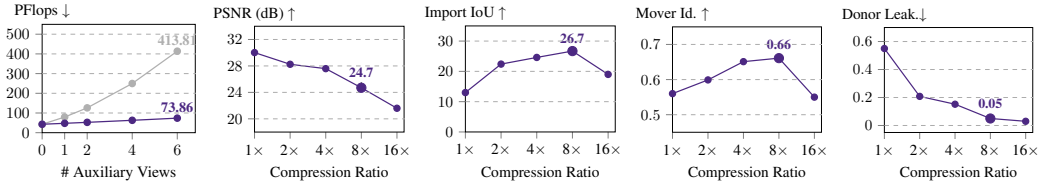


Figure 6: **Effect of δ-VAE compression.** δ-VAE keeps compute nearly flat as auxiliary views are added. Stronger compression trades reconstruction fidelity for disentanglement: donor leakage drops steadily, while mover transfer peaks at the default $8\times$ before degrading.

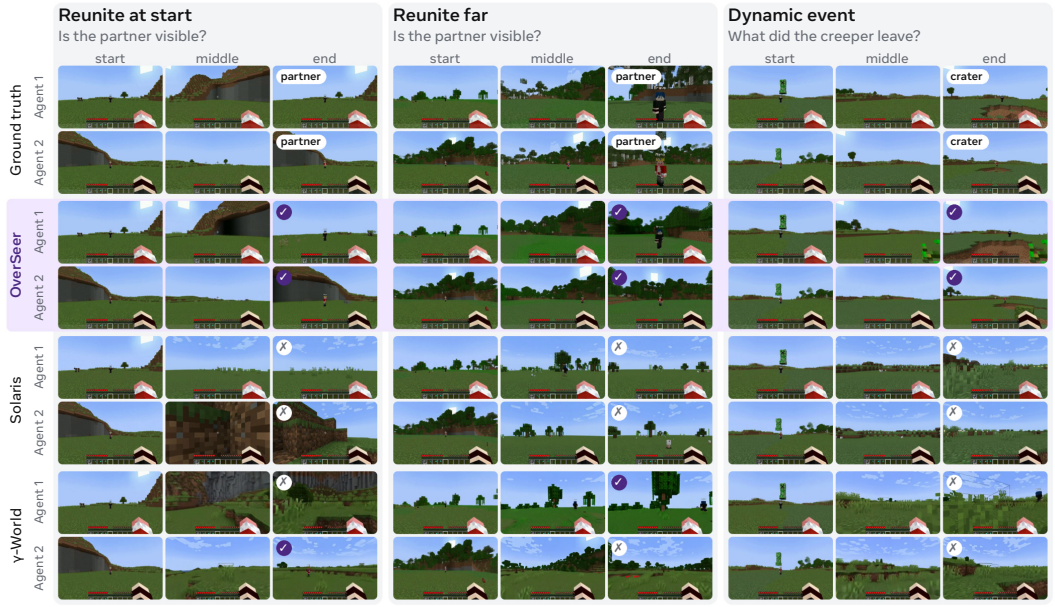


Figure 7: **OverSeer keeps the world consistent over time.** On the three revisiting tasks (reunite at start, reunite far, dynamic event), OverSeer shows the partner and the crater where the ground truth does, while Solaris and γ-World mostly do not.

Effect of auxiliary camera count. Increasing the number of auxiliary cameras steadily improves every metric relying on persistent world state: grounding, memory, and consistency on Solaris, and all three revisit tasks in MAC-Bench, while movement and building performance remain flat (fig. 5). This selective improvement confirms that performance gains stem specifically from expanded spatial visual coverage rather than generic model capacity scaling, which would otherwise uniformly elevate all capabilities. Gains naturally plateau once visual coverage saturates.

Effect of latent compression. In cross-scene transfer experiments (decoding residual codes with a different "control" anchor latent), higher compression lowers PSNR but systematically suppresses donor background leakage. Crucially, spatial placement precision and identity preservation of the transferred mover (Import IoU and Mover Id.) *improve* up to $8\times$ compression—peaking at 26.7 and 0.66, respectively, before degrading at $16\times$ (fig. 6). Compression is thus more than an efficiency knob: an informational bottleneck too narrow to encode static background forces the model to allocate capacity to dynamic elements, causing dynamic-static disentanglement to naturally emerge from the constraint itself. We therefore adopt $8\times$ compression by default (fig. 6).

5 Conclusion

We introduced OverSeer, a multi-agent video world model that jointly generates egocentric agent views and fixed-pose auxiliary views as a lightweight proxy for a world-centric state. These views substantially improve consistency when agents separate and later reunite, and δ-VAE makes them cheap to model by encoding only changes relative to a static reference. We hope this simple yet effective design will pave the way for world models that simulate shared environments beyond what any single agent sees.

Ethics Statement

This work trains and evaluates world models on two video games and a public multi-view dataset. Minecraft and Doom data are generated by scripted agents in simulation and contain no personal information; Doom experiments use the open-source Freedoom assets. Real-world results use the publicly released Replay dataset (Shapovalov et al., 2023), whose participants were recorded with consent by its original authors, and we collect no new data involving human subjects. As with other video generation models, multi-agent world models could be misused to fabricate realistic multi-view footage of people, including footage resembling fixed surveillance cameras. Our real-world results are qualitative, limited to a public dataset, and do not target identity-specific generation; we recommend that any released models carry usage restrictions against impersonation and fabricated footage. Finally, MAC-Bench uses a proprietary VLM as judge. The judge queries are restricted to verifiable facts such as partner visibility and event state.

Reproducibility Statement

The model architecture, δ-VAE, and training objectives are described in sections 3.2 and 3.3; auxiliary camera pose sampling is detailed in section B; action prompts are listed in section F.2, and hyperparameters, compression rates, context length, and compute are listed in section D. The backbone, Wan2.2-TI2V-5B (Wan et al., 2025), is publicly available, and both baselines are evaluated from their public non-distilled checkpoints with identical inputs. Task specifications, episode counts, and the evaluation protocol and prompt of MAC-Bench are given in section A and section F.1, with additional ablations in section C. We will try our best to release code, trained models, and the benchmark upon publication.

LLM Usage Statement

LLMs were used to assist with proofreading the manuscript. LLMs were also used to assist coding, particularly to help with visualizations, and for debugging the code. LLMs were not involved in the ideation or development of the research. The authors have carefully reviewed and verified all LLM-assisted outputs and take full responsibility for the scientific integrity of this work.

References

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C.Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Junyu Chen, Dongyun Zou, Wenkun He, Junsong Chen, Enze Xie, Song Han, and Han Cai. DC-AE 1.5: Accelerating diffusion model convergence with structured latent space. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2025.

DreamX, Yancheng Bai, Rui Chen, Xiangxiang Chu, Rujing Dang, Hao Dou, Bingjie Gao, Qiwen Gu, Siyu Hong, Jiachen Lei, Geng Li, Jifan Li, Ruimin Lin, Qingfeng Shi, Bingze Song, Lei Sun, Jing Tang, Ruitian Tian, Jun Wang, Jiahong Wu, Pengfei Zhang, Shen Zhang, and Jiashu Zhu.

Dreamx-world 1.0: A general-purpose interactive world model, 2026. URL https://arxiv.org/abs/2606.16993.

Vladimir Egorov and Aleksei Shpilman. Scalable multi-agent model-based reinforcement learning. *arXiv preprint arXiv:2205.15023*, 2022.

Team Enigma. Introducing multiverse: The first ai multiplayer world model, 2025. URL https://enigma.inc/blog.

Xiang Fan, Xiaohang Sun, Kushan Thakkar, Zhu Liu, Vimal Bhat, Ranjay Krishna, and Xiang Hao. RefTok: Reference-based tokenization for video generation. *arXiv preprint arXiv:2507.02862*, 2025.

Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin Brualla, Pratul Srinivasan, Jonathan Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a.

Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. Magicdrive: Street view generation with diverse 3d geometry control. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024b.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024c.

Google DeepMind. Veo 3 technical report. Technical report, Google DeepMind, 2025. URL https://storage.googleapis.com/deepmind-media/veo/Veo-3-Tech-Report.pdf.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2025.

Yicong Hong, Yiqun Mei, Chongjian Ge, Yiran Xu, Yang Zhou, Sai Bi, Yannick Hold-Geoffroy, Mike Roberts, Matthew Fisher, Eli Shechtman, Kalyan Sunkavalli, Feng Liu, Zhengqi Li, and Hao Tan. Relic: Interactive video world model with long-horizon memory, 2025. URL https://arxiv.org/abs/2512.04040.

Anthony Hu, Václav Volhejn, Adrien Ramanana Rahary, Chris Mulder, Aditya Makkar, Alyx Liao, Amélie Royer, Manu Orsini, Adam Jelley, Eloi Alonso, Florian Laurent, Fredrik Norén, James Swingos, Jan Hünermann, Kent Rollins, Lucas Hosseini, Matthieu Le Cauchois, Maxim Peter, Pim de Witte, Tim Brown, Vincent Micheli, Moritz Böhle, Gabriel de Marmiesse, Viktoriia Sharmanska, Lucia Specia, Michael Black, and Patrick Pérez. Multiplayer interactive world models with representation autoencoders, 2026a. URL https://arxiv.org/abs/2607.05352.

Teng Hu, Mingchun Lu, Yating Wang, Jiangning Zhang, Jinkun Hao, Ye Pan, Ran Yi, Lizhuang Ma, and Dacheng Tao. Metaworld: Scaling multi-agent video world model from single-view video data. *arXiv preprint arXiv:2606.02753*, 2026b.

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

Wonbong Jang, Shikun Liu, Soubhik Sanyal, Juan Camilo Perez, Kam Woh Ng, Sanskar Agrawal, Juan-Manuel Perez-Rua, Yiannis Douratsos, and Tao Xiang. Rays as pixels: Learning a joint distribution of video and camera trajectories. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026.

Fan Jiang, Zhaoxu Sun, Mengchao Wang, Ziyu Zhu, Chiyu Wang, Yunpeng Zhang, Wenlin Liu, Yun Wang, Xue Zheng, Rui Sun, Junfeng Ni, Hongyu Pan, Zhongxu Sun, Fei Yu, Zengye Ge, Mengmeng Du, Nianfei Fan, Mingchao Sun, Yu Liu, Yongchang, Yanqing Zhu, Jiahang Wang, Ning Ying, Yuze Xuan, Di Yang, Zhicheng Liu, Zhe Gao, Tingbing Xu, Jiacheng Sui, Wenjin

Yang, Junnan Lai, Shufeng Liu, Yuan Liu, Zheng Zhou, Yingliang Peng, Dawei Cao, Kaifeng Sheng, Yuxiang Cai, Fei Lu, Mu Xu, and Ning Guo. Abot-world-0: Infinite interactive world rollout on a single desktop gpu, 2026. URL https://arxiv.org/abs/2607.19191.

Tommie Kerssies, Gabriele Berton, Ju He, Qihang Yu, Wufei Ma, Daan de Geus, Gijs Dubbelman, and Liang-Chieh Chen. A frame is worth one token: Efficient generative world modeling with delta tokens. *arXiv preprint arXiv:2604.04913*, 2026.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Xin Kong, Shikun Liu, Xiaoyang Lyu, Marwan Taher, Xiaojuan Qi, and Andrew J Davison. Eschernet: A generative model for scalable view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Jiahao Li, Bin Li, and Yan Lu. Deep contextual video compression. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Runjia Li, Philip Torr, Andrea Vedaldi, and Tomas Jakab. VMem: Consistent interactive video scene generation with surfel-indexed view memory. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2025.

Fangfu Liu, Kai He, Tianchang Shen, Tianshi Cao, Sanja Fidler, Yueqi Duan, Jun Gao, Igor Gilitschenski, Zian Wang, and Xuanchi Ren. Gamma-world: Generative multi-agent world modeling beyond two players. *arXiv preprint arXiv:2605.28816*, 2026a.

Shikun Liu, Kam Woh Ng, Wonbong Jang, Jiadong Guo, Junlin Han, Haozhe Liu, Yiannis Douratsos, Juan Camilo Perez, Zijian Zhou, Khanh Chi Phung, Tao Xiang, and Juan-Manuel Perez-Rua. Scaling sequence-to-sequence generative neural rendering. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026b.

Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. DVC: An end-to-end deep video compression framework. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

OpenAI. Video generation models as world simulators. Technical report, OpenAI, 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proc. AAAI*, 2018.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*, 2020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

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, 2026. URL https://arxiv.org/abs/2602.22208.

Roman Shapovalov, Yanir Kleiman, Ignacio Rocco, David Novotny, Andrea Vedaldi, Changan Chen, Filippos Kokkinos, Ben Graham, and Natalia Neverova. Replay: Multi-modal multi-view acted videos for casual holography. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

Vincent Sitzmann, Semon Rezchikov, William T. Freeman, Joshua B. Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. In *Proc. NeurIPS*, 2021.

The Freedoom project. Freedoom: Phase 1 and phase 2. https://freedoom.github.io/, 2024. Free game data for the Doom engine, 36 and 32 maps.

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

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

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

Haoyu Wu, Jiwen Yu, Yingtian Zou, and Xihui Liu. Multiworld: Scalable multi-agent multi-view video world models. *arXiv preprint arXiv:2604.18564*, 2026.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, et al. Longlive: Real-time interactive long video generation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Hidir Yesiltepe, Tuna Meral, Adil Kaan Akan, Kaan Oktay, and Pinar Yanardag. Infinity-rope: Action-controllable infinite video generation emerges from autoregressive self-rollout. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

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. In *SIGGRAPH Asia*, 2025a.

Lijun Yu, José Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David Ross, and Lu Jiang. Language model beats diffusion - tokenizer is key to visual generation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025b.

Wei Yu, Runjia Qian, Yumeng Li, Liquan Wang, Songheng Yin, Sri Siddarth Chakaravarthy P, Dennis Anthony, Yang Ye, Yidi Li, Weiwei Wan, and Animesh Garg. Mosaicmem: Hybrid spatial memory for controllable video world models, 2026. URL https://arxiv.org/abs/2603.17117.

ZDoom developers. GZDoom. https://zdoom.org/, 2025. Open-source source port of the Doom engine.

Yang Zhang, Chenjia Bai, Bin Zhao, Junchi Yan, Xiu Li, and Xuelong Li. Decentralized transformers with centralized aggregation are sample-efficient multi-agent world models. *Transactions on Machine Learning Research (TMLR)*, 2025.

A Dataset Curation Pipelines

Data. The three domains are three separate datasets, and we train one model on each. Minecraft and Doom are collected by us; the real captures are an existing dataset that we re-use in the same two-agent, many-camera arrangement. In all three, a clip is 241 frames of every view at once, sampled at 20 frames per second (12 s), 576×320 per view in Minecraft and in the captures and 480×320 in Doom.

Minecraft. The gameplay is collected with the Solaris engine (Savva et al., 2026): two Mineflayer bots play a two-player episode on a Minecraft 1.21 server and each view is recorded from a real Minecraft client, so the frames are the game's own graphics and every frame carries the bot's ground-truth action. The bots explore by a potential field, walls pushing them away and unvisited ground pulling them in, which keeps the pair moving over new terrain instead of circling one spot, and each episode is additionally labeled with one of eleven interaction modes (mining, building, fighting, chasing, exploring, reuniting and so on). We extend the engine with six fixed auxiliary cameras. They are placed in a second pass: the episode is played once to establish the terrain, then the six cameras are sampled at random in the free space around the agents' starting area and the episode is captured with them. Placement therefore follows the ground the episode covers rather than aiming at one point; measured over 200 episodes, a camera sits a median 4.9 blocks above the agents and 3.0 blocks from the nearest point of their path, tilted 52 degrees down with a 70 degree field of view, and while a single camera has an agent in its frustum in about half of the frames, some camera has one in 90% of them. Every view is recorded at 1280×720 and 20 frames per second. We collect 20,000 episodes of outdoor terrain and 11,000 of village worlds, 264 frames each.

Doom. Doom episodes are played with Doom (ZDoom developers, 2025) on the 68 maps of Freedoom Phase 1 and Phase 2 (The Freedoom project, 2024), two marines and eight fixed auxiliary cameras per episode, and the marines are driven by the same potential-field exploration controller as the Minecraft bots. Every view is recorded at 1024×768 and 30 frames per second with a 120-degree field of view. The eight cameras sit about 100 map units (3 m) above eye height, four tilted 10 degrees down and four 28 degrees, and between them they see the marines almost everywhere they walk, which is what the maps' corridors and halls allow: a single camera has a marine in its frustum in about 85% of the frames. We train on 20,000 traversal episodes and hold out a separate set of 4,000 arena episodes, in which the two marines start 128 map units apart facing each other, for evaluation.

Real captures. The real sequences come from Replay (Shapovalov et al., 2023), 68 scenes of acted social interaction of about five minutes each, filmed by twelve temporally synchronized, geometrically and color-calibrated cameras. We use eight of them in the arrangement the two games are built on: the two head-mounted cameras, which move with their wearers, are the agent views, and six static cameras are the auxiliary views. Unlike the games this gives real people, real cameras, and measured poses: every camera comes with a per-frame camera-to-world pose and pinhole intrinsics; the static cameras hold still to within 5 mm and a wearer covers 3 to 4 m in a clip. We hold out 45 clips of 241 frames, drawn from 35 of the scenes, and train on the rest of the dataset.

MAC-Bench. Table 3 lists the task specifications. In all tasks, the agents begin mutually visible and lose sight of each other mid-episode. In *Dynamic Event*, the scripted event is placed at the midpoint between the agents and is one of sprouting crops, a standing creeper, or a plank cube that ignites at frame 20. Doom uses the two reunite tasks on six Freedoom maps with 1047-unit walks (32 units per block), with the partner 84° off-axis at the midpoint. We use 32 episodes per task in Minecraft and 100 per task in Doom. For VLM accuracy, Claude Opus 5 is given the generated frames and answers the task query for each agent;

An episode counts as correct only if the answers for both agents match the ground truth. FID is computed between generated and ground-truth agent frames. During evaluation, accuracy is averaged over 5 independent runs of 32 episodes; episodes with failed generation or unparsable judge output are excluded.

Task	Start	Mid-episode	End	Query
Reunite at Start	13 blocks apart, facing	each wanders ≤ 9 blocks	exact start poses	Is the partner visible?
Reunite Far	17 blocks apart	$\sim$59 blocks each, converging; partner 86° off-axis	2.9 blocks apart, facing	Is the partner visible?
Dynamic Event	13 blocks apart, event at midpoint	separate to 40 blocks by frame 76	4.3 blocks apart	What happened to the event?

Table 3: **MAC-Bench task specifications in Minecraft.** Mid-episode angles are medians over episodes.

B Auxiliary Camera Pose Sampling

As described in section 3.2, OverSeer requires auxiliary camera poses $\mathbf{c}$ in eq. (3) but no auxiliary images at inference time. During training, auxiliary camera poses and their corresponding videos are provided by the dataset. At inference time, we sample the poses as follows.

Scene-Independent Pose Prior. We define a canonical coordinate system whose origin is the initial position of the first agent. Candidate auxiliary camera poses are sampled relative to this coordinate system from a scene-independent camera-pose prior. The horizontal offset from the origin, camera height, yaw, and pitch are modeled as Gaussian variables whose parameters are estimated from the training camera poses. Camera height is constrained to be non-negative, and roll is fixed to zero.

Coverage-Based Selection. We sample a candidate set of poses $\mathcal{C}$ from this prior and greedily select K cameras to maximize geometric field-of-view coverage. Let Ω denote a bounded discretized volume in the canonical coordinate system, and let $\mathcal{F}(\mathbf{c}) \subseteq \Omega$ denote the cells geometrically covered by the camera frustum associated with pose $\mathbf{c}$. The computation of $\mathcal{F}(\mathbf{c})$ depends only on camera geometry and does not use scene occupancy or occlusion information. Let $\mathcal{R}_k$ denote the region covered by the first k selected cameras, with $\mathcal{R}_0 = \emptyset$. We greedily select

$$\mathbf{c}_{k+1} = \arg\max_{\mathbf{c}\in\mathcal{C}} |\mathcal{F}(\mathbf{c}) \setminus \mathcal{R}_k|, \qquad \mathcal{R}_{k+1} = \mathcal{R}_k \cup \mathcal{F}(\mathbf{c}_{k+1}), \tag{6}$$

until K cameras are selected. Since both candidate sampling and coverage computation depend only on the canonical coordinate system and camera geometry, this procedure requires no additional scene-specific visual or geometric input.

C More Ablations

	δ-VAE	Minecraft Multi-agent Consistency Benchmark					
	PSNR↑	Reunite at Start		Reunite Far		Dynamic Event	
Variant	dB	VLM↑	FID↓	VLM↑	FID↓	VLM↑	FID↓
w/o $\mathcal{L}_{\text{align}}$	25.1	56.9	13.47	33.4	12.26	38.1	17.08
Full δ-VAE	24.7	60.2	13.99	37.9	12.79	40.5	17.61

Table 4: **Ablation of the feature alignment loss.** We evaluate its effect on δ-VAE reconstruction quality and downstream world consistency in Minecraft.

Feature alignment loss. Table 4 ablates $\mathcal{L}_{\text{align}}$. Removing it slightly improves δ-VAE reconstruction (25.1 vs. 24.7 dB PSNR) and FID, but lowers VLM accuracy on all three MAC-Bench tasks, by 2.4–4.5 points. Without alignment, the residual code spends its capacity on reconstructing the whole frame; with it, the code is pushed towards dynamic regions, which is what the world model needs to keep revisited scenes consistent. Reconstruction fidelity is thus not a sufficient proxy for how useful a compressed representation is downstream, in line with the compression trend in section 4.2.

D Implementation Details

By applying 2×2 spatial ($s = 2$) and $2\times$ temporal ($\tau = 2$) compression rates, the δ-VAE reduces the auxiliary view representation from 864 tokens per frame (at 576×384 resolution) to just 108 tokens, achieving an $8\times$ overall bandwidth reduction. OverSeer jointly generates both agents' egocentric streams and K auxiliary views (K=6 by default for Minecraft and K=8 for Doom) over a 241-frame horizon, conditioned on the initial frame of agent views and both agents' action sequences.

Built upon Wan2.2-TI2V-5B (Wan et al., 2025), the model is fully fine-tuned per domain for 20k steps using Adam with a learning rate of 10^{-4} and a batch size of 64 across 64 NVIDIA GB300 GPUs. Training incorporates diffusion forcing and causal masking over a context window of 12 latent frames (48 RGB frames), with autoregressive rollout used during inference.

For training the δ-VAE, we use Adam with a learning rate of 10^{-4} and we set $\lambda_{\text{rec}} = 1$, $\lambda_{\text{KL}} = 10^{-5}$, and $\lambda_{\text{align}} = 10^{-3}$.

E Limitations and Future Work

OverSeer's benefits depend on auxiliary coverage. Regions far from the sampled cameras, as in *Reunite Far*, remain harder to keep consistent. Short-horizon egocentric capabilities such as movement do not benefit from auxiliary views. Aggressive residual compression also trades some fidelity and, in fast-changing scenes such as Doom, consistency for efficiency. Promising directions include adaptive or learned camera placement that follows agents into unexplored regions, auxiliary views that are re-anchored for unbounded worlds, and scaling to larger agent populations. More broadly, we hope this work encourages treating the *output space* of world models, meaning what they choose to render and not only what they condition on, as a first-class design axis for building coherent, shared simulated worlds.

F Prompts

F.1 The judge

Every question in both benchmarks is asked of the same judge, Claude Opus 5, in one turn: the frame or the pair of frames first, then the question, under the system prompt *"You are a helpful assistant that evaluates Minecraft screenshots."* Thinking is adaptive at low effort. The answer is reduced to its first non-empty line, lowercased and stripped of quotes and trailing punctuation, and validated by exact match against the expected word, so every prompt ends by naming the words it accepts. An episode counts as correct only if every question about it is answered correctly.

Revisiting benchmark. The two reunite tasks ask only about the partner, in one sentence:

```
Here is a Minecraft/Doom screenshot.  Is there another
player visible on-screen?  Answer with a single word:
"yes", "no".
```

It is put to each agent's own view at four moments read off the recorded trajectory: the first generated frame, the frame of maximum separation, the midpoint of the way back, and the rendezvous, which makes eight questions per episode. The judge sees one frame and this sentence, with no description of the task and no earlier frame, so nothing but the frame can decide the answer; the expected answer comes from the poses, the partner counting as visible when it lies well inside the view cone and within 20 blocks and as not visible when it lies outside the cone or beyond 35 blocks, with the judge's own answer on the ground-truth frame standing in for the band between the two. The dynamic events add two more question shapes. The outcome question names the states the event can be in, one set per event kind; for the creeper,

```
Here is a Minecraft screenshot.  What is visible where the
creeper stood:  a creeper still standing, a crater or hole
blasted into the ground, or flat undamaged ground?  Answer
with a single word from "creeper", "crater", "flat".
```

with "sprouts"/"wheat"/"none" for the growing crops, "intact"/"burning"/"partly burnt"/"gone" for the burning plank cube. The progression question shows the first and the return frame together:

```
Here are two Minecraft screenshots from roughly the same
viewpoint, the first taken earlier and the second later.  In
this scene {process}.  Compared with the first image, is
that process further along in the second image (progressed),
at the same stage (same), or at an earlier stage than in the
first image (reverted)?  Answer with a single word:
"progressed", "same", "reverted".
```

Entry	Sentence
forward	The player is walking forward
left	The player is strafing to the left
back	The player is walking backward
right	The player is strafing to the right
jump	The player is jumping into the air
sneak	The player is sneaking and crouching down
sprint	The player is sprinting and running fast
drop	The player is dropping the held item
inv	The player is opening the inventory
esc	The player is pressing escape to pause
tab	The player is switching the active hotbar tab
hot1–hot9	The player is selecting item slot one … nine in the hotbar
swap_hands	The player is swapping the items in both hands
attack	The player is attacking and hitting with their weapon
use	The player is using or interacting with an object
mouse_turn_left	The player is turning the camera left
mouse_turn_right	The player is turning the camera right
mouse_look_up	The player is looking up
mouse_look_down	The player is looking down
idle	The player is standing still and doing nothing

Table 5: The action vocabulary, one sentence per entry. The nine hotbar entries differ only in the number they name. drop, inv, esc, tab and swap_hands are never pressed in our episodes and are kept only so that the vocabulary matches the game's input.

where {process} is a phrase for the event, for instance *"a creeper explodes, disappears, and leaves a crater blasted into the ground"*. Each event kind also has a presence question asked at the first generated frame (*"Is there a creeper (green four-legged hostile mob) visible on-screen?"* and its equivalents).

One sentence we add. Claude reads the viewer's own first-person arm in the corner of a sharp frame as another player. Where that matters, we append, to every prompt and identically in both passes,

```
The large blocky shape in the bottom-right corner is the
viewer's own first-person arm and held item, not another
player.
```

The prompt actually used is recorded in every result file.

F.2 Action prompts

Actions reach the model as language. Each of the 28 entries of the action vocabulary is written as one English sentence, encoded once by the text encoder, and stored as a table of token sequences; at training and at inference the sentence of the frame's action is looked up and its tokens are used as the frame's conditioning context. Table 5 lists the sentences.

Per frame exactly one entry is chosen. The keyboard flags score one when pressed; the mouse contributes through its yaw and pitch, whose sign picks turning left or right and looking up or down and whose magnitude is clipped to one; if nothing scores above zero, the frame is idle. The highest score wins, so a frame that both walks and turns is conditioned on the stronger of the two. The conditioning frame's tokens are zeroed, since it is given rather than generated, and during training the whole action context is dropped with a fixed probability so that the model can be sampled with classifier-free guidance on the actions.

G More Visualizations

Figures 8 and 9 show additional MAC-Bench episodes in Minecraft and Doom, comparing OverSeer with an egocentric-only variant trained without auxiliary views. Figures 10 to 12 show long rollouts in Minecraft, Doom and real-world captures, with both agent views and all auxiliary views; for real

captures we also show a top-down layout of the cameras and agent trajectories. Videos are available on our website.



Figure 8: **Additional MAC-Bench results in Minecraft.** Top: the three benchmark tasks. Bottom: three further dynamic events. OverSeer recovers the partner or the evolved event state where the ground truth does, while the egocentric-only variant does not.

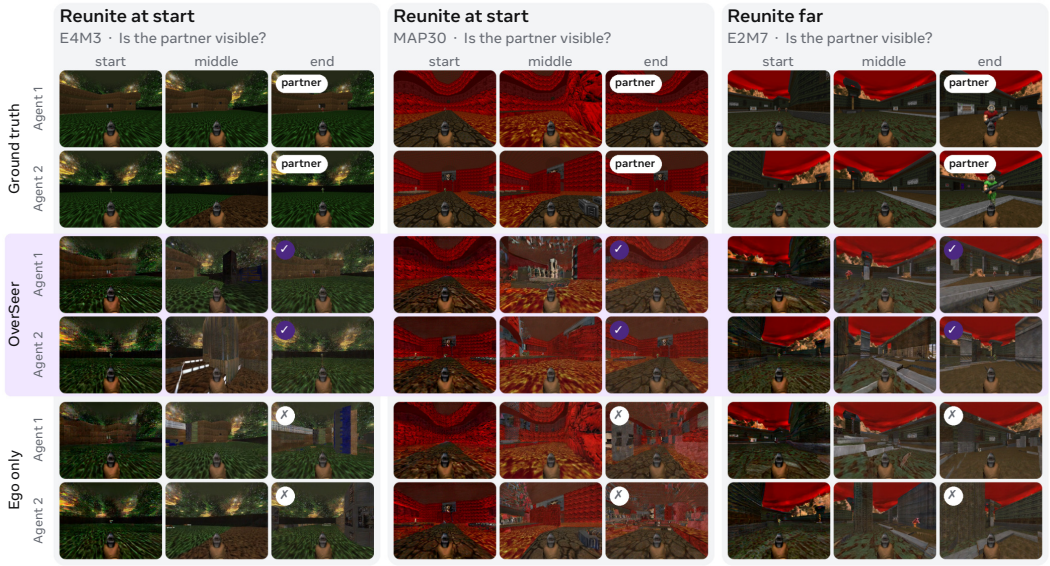


Figure 9: **MAC-Bench results in Doom.** Reunite tasks on three Freedoom maps. OverSeer restores the partner after the agents separate; the egocentric-only variant does not.

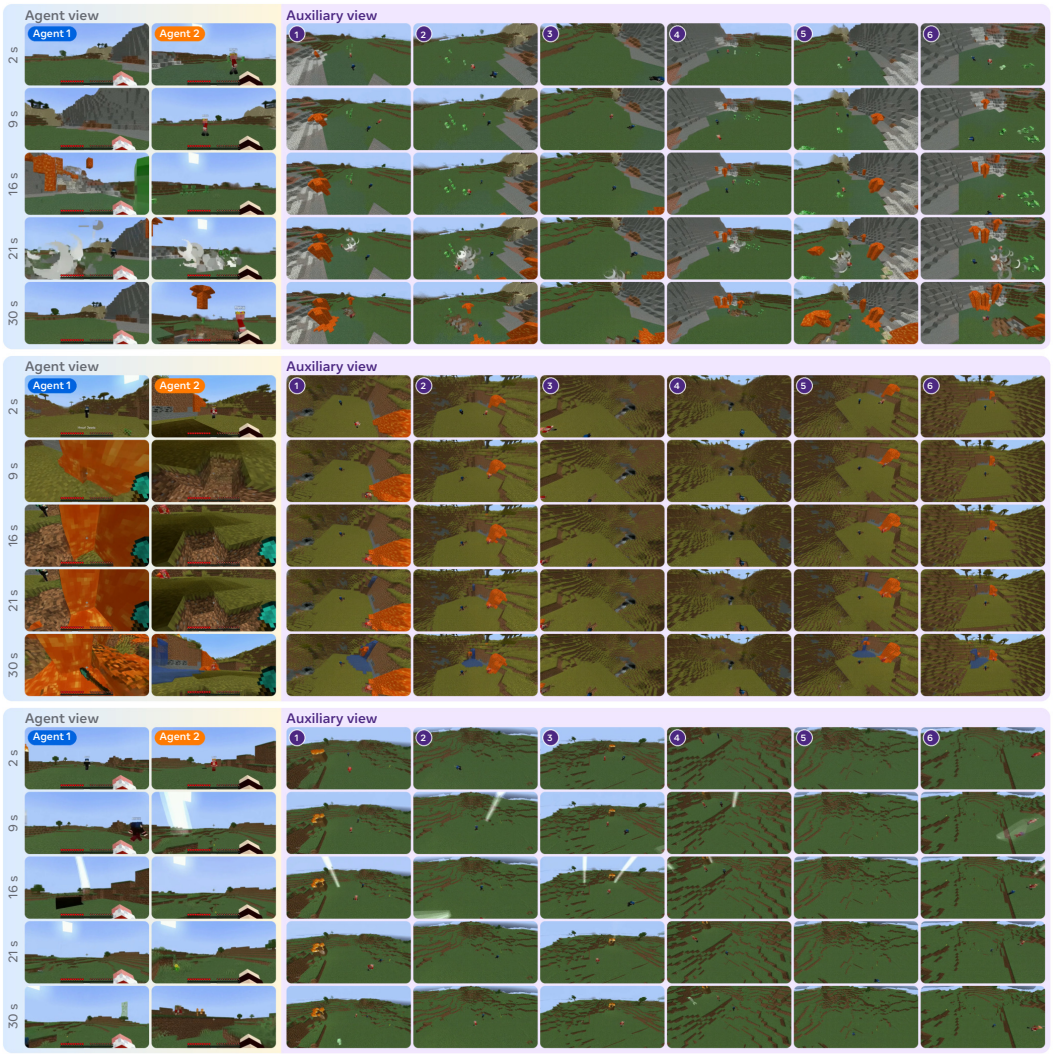


Figure 10: **Long rollouts in Minecraft.** Each block shows both agent views and all six auxiliary views over 30 s.

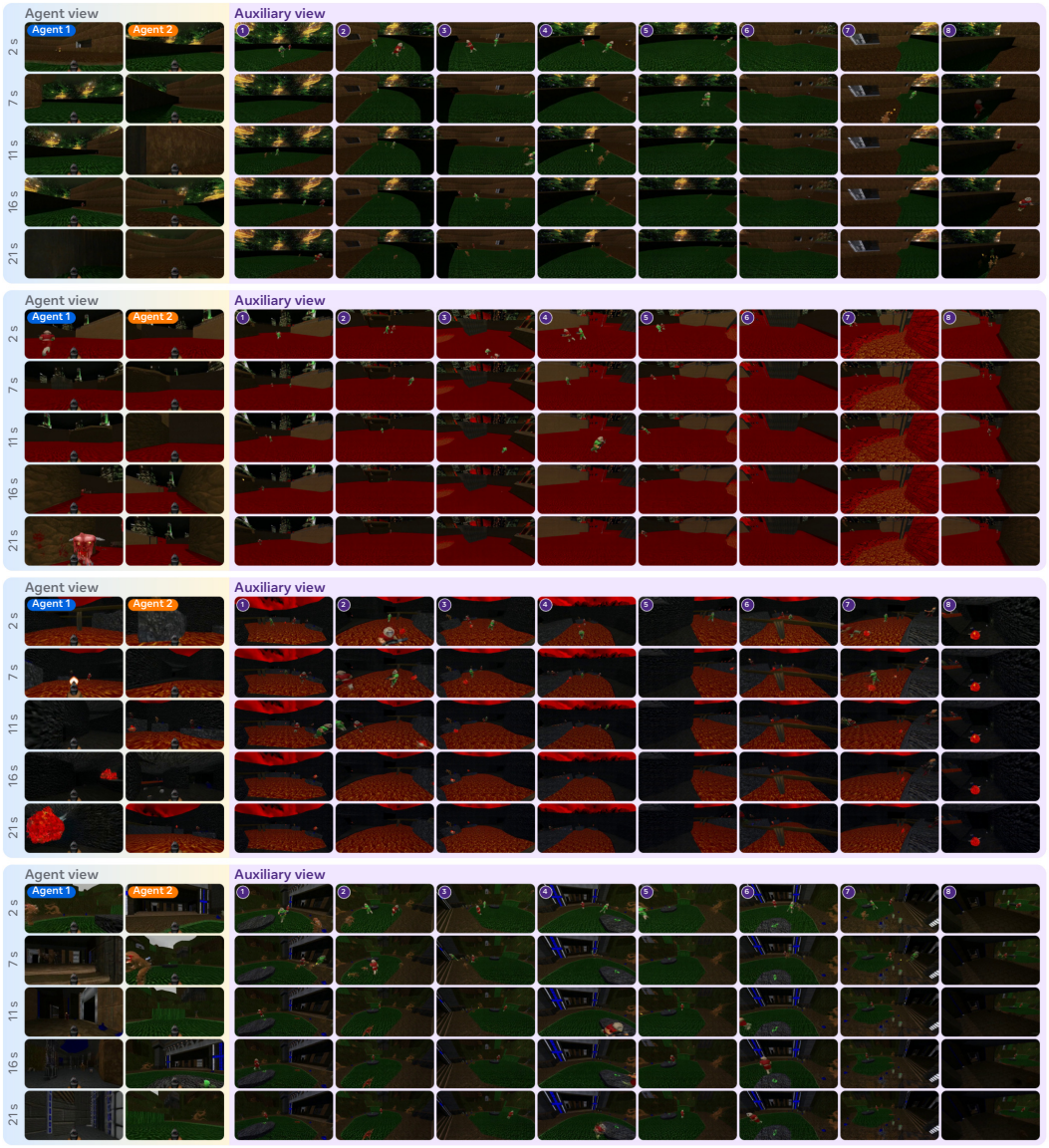


Figure 11: **Long rollouts in Doom.** Each block shows both agent views and all eight auxiliary views over 21 s.

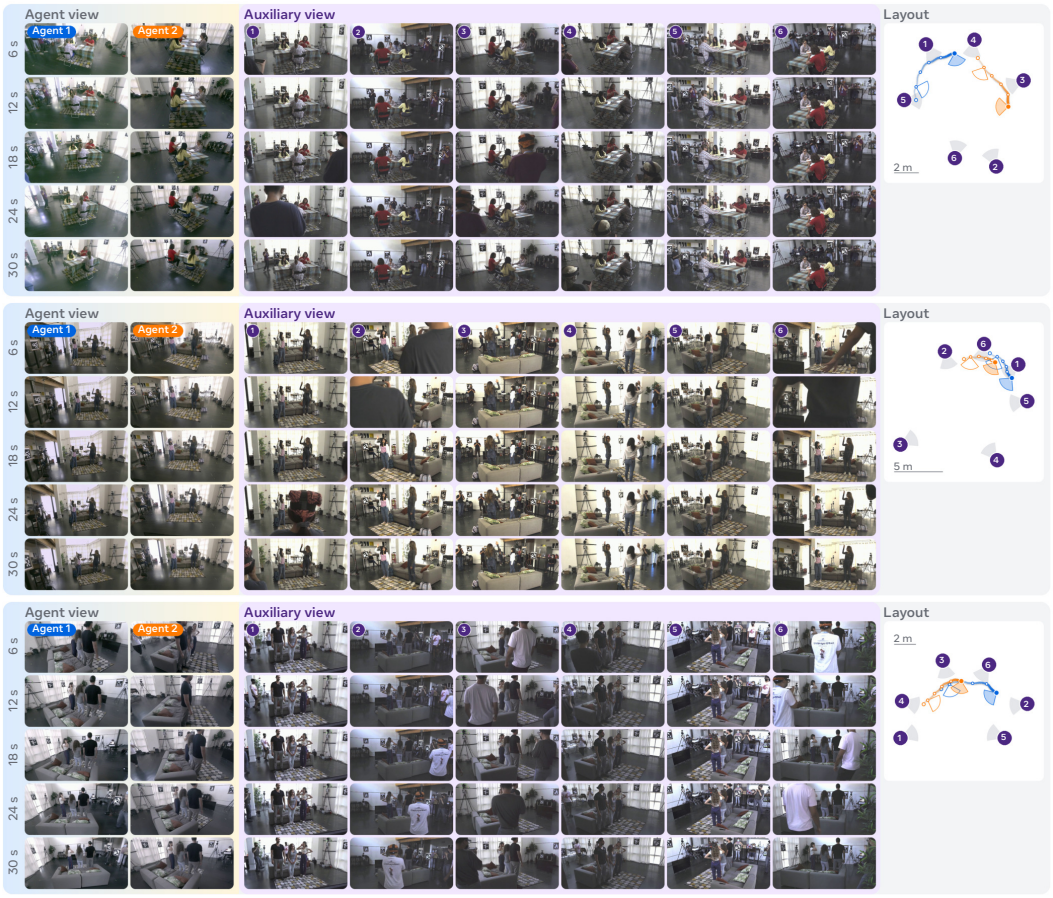


Figure 12: **Long rollouts on real-world multi-camera captures.** Left: both agent views and all six auxiliary views over 30 s. Right: top-down layout of the auxiliary cameras and agent trajectories.