SYNCWORLD: MANY VIEWS, ONE WORLD

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-player interactive worlds must support distinct views of one evolving scene. We introduce SyncWorld, which decouples shared world evolution from observer-specific video generation. Joint actions advance an authoritative state, geometric projections express that state in each observer's image plane, and one shared video model renders the observations. Additional observers reuse the same renderer weights. We instantiate the architecture with engine-authoritative dynamics and a video-diffusion backbone adapted through geometric and status conditioning, gated residual adapters, and LoRA. Our CS:GO dataset pairs multi-observer video with cameras, actions, events, and state. We will release the full five-map collection of 271,568 first-person videos and accompanying records (approximately 48.66 TB), together with 84,066 processed Dust2 clips (118 cumulative hours). We evaluate both visual fidelity and actor-condition fidelity, and introduce SYMMVC, a symmetric reprojection residual using dense correspondences and calibrated cameras, to measure cross-view geometric compatibility. Under matched training data and budget, SyncWorld improves PSNR from 21.3329 to 22.5274 dB and Presence Alignment from 0.5771 to 0.7748 over Direct Residual. SyncWorld further achieves a 480-pixel SYMMVC residual of 0.5259 versus 5.1 for Shuffled Geometric Condition. A native trajectory drives ten concurrent views; autoregressive evaluation reaches 12 windows (about 60 seconds).

Video demos | **Project page**

1 INTRODUCTION

Interactive world models generate observations in response to control inputs, bringing learned visual generation into simulated environments (Ha and Schmidhuber, 2018; Hafner et al., 2025; Bruce et al., 2024; Valevski et al., 2025). In a multi-observer setting, these observations must refer to the same evolving world: players seen from different cameras should occupy compatible locations, and each observer's video should follow the shared sequence of world states. The central problem is therefore to connect observer-specific generation to a common world while allowing additional observers to use the same model.

Existing approaches offer complementary ways to connect video to a world. Camera-conditioned generation controls viewpoint, while GEN3C (Ren et al., 2025) conditions on renderings of a persistent 3D cache. These mechanisms anchor viewpoint and scene geometry, but multi-observer generation additionally requires observations to follow a common evolving state. Solaris (Savva et al., 2026) exchanges information between players through multiplayer self-attention, coupling their video representations during generation. MultiGen (Po et al., 2026) also separates memory, observation, and dynamics and supports distributed generation from shared memory. SyncWorld follows the latter systems principle but places the shared information in an engine-authoritative 3D world state: each observer receives its own projected condition and autoregressive context without reading another observer's video tokens. A central design choice is to distinguish the actors that update the shared world from the observers that query it.

SyncWorld separates *what happens in the world* from *how that world is observed*. Joint actions advance a single authoritative state containing scene geometry and all players. The state is then projected into each observer's frame, producing geometric and status conditions for a shared video renderer. World evolution is shared across observers, while video generation remains local to each observer. This separation makes observer count an inference-time choice: adding a view adds a renderer call without changing the learned weights or the other observers' generation paths.

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.

We instantiate SyncWorld in Counter-Strike: Global Offensive (CS:GO). The server advances the shared state, and a condition builder projects geometry and players into each observer's image plane. Condition-gated residual adapters and low-rank adaptation connect these projections to a camera-conditioned video model. Each observer continues generation from its own final frame while following the shared trajectory.

Studying this interface requires observations linked to a common world state. Our replay-derived collection spans five maps and contains 271,568 first-person videos with camera, action, event, and state records, totaling approximately 48.66 TB. Shared match and episode identifiers link observations across players. From Dust2, we select 84,066 clips for the experiments. We will release the full five-map collection and this processed subset to support multi-map research and cross-view evaluation.

Model comparisons and condition interventions evaluate the renderer, while SYMMVC tests cross-view geometric compatibility. Separate experiments demonstrate ten-observer generation and approximately 60 seconds of autoregressive continuation.

The paper makes three contributions:

- **View-decoupled interactive generation.** We formulate and implement SyncWorld, a shared-state architecture that separates joint world evolution, per-observer condition construction, and shared-weight video rendering. New observers enter through independent renderer calls, with their own autoregressive contexts.
- **A multi-observer dataset with world-state annotations.** We will release 271,568 first-person videos across five maps, approximately 48.66 TB including camera, player-state, action, and event records. We also provide a processed Dust2 subset of 84,066 clips with aligned geometric and status conditions.
- **Evaluation of condition fidelity and shared-world consistency.** We evaluate actor presence, image-plane position, and action/state realization under structured conditioning, and introduce SYMMVC to measure cross-view geometric compatibility using dense correspondences and calibrated cameras. Multi-observer and autoregressive experiments further characterize the spatial and temporal behavior of the shared-world interface.

2 RELATED WORK

Interactive generation. World models support prediction and control through learned representations of environments (Ha and Schmidhuber, 2018; Hafner et al., 2020; 2025). Generative interactive systems such as Genie, GameNGen, DIAMOND, and Oasis study action-conditioned visual environments (Bruce et al., 2024; Valevski et al., 2025; Alonso et al., 2024; Decart et al., 2024). Action-conditioned autoregressive models such as Vid2World generate future visual observations from past observations and action sequences (Huang et al., 2026). Recent video-world systems further develop interactive generation, camera and history conditioning, and controllability (Zhang et al., 2025; Li et al., 2025; Mao et al., 2025; Yu et al., 2025). LingBot-World (Robbyant Team et al., 2026) provides the camera-conditioned video-generation backbone used by our renderer; SyncWorld augments it with projected world-state conditioning and the shared-world formulation. Our focus is the relationship between multiple observation streams: a shared state supplies the instance-level world that every observer queries.

Shared worlds and multi-view generation. Multi-view diffusion methods couple observations through joint generation or scene representations (Shi et al., 2024; Liu et al., 2023). Solaris exchanges information between Minecraft players through multiplayer self-attention (Savva et al., 2026). Multi-World (Wu et al., 2026) studies scalable multi-agent, multi-view generation with joint action conditioning and a shared global state representation. MultiGen distributes per-player generation around shared external memory (Po et al., 2026); its Doom representation uses a two-dimensional wall layout, planar position and yaw, and a one-dimensional depth readout. Our CS:GO implementation computes observer-plane geometry and status conditions from an engine-authoritative three-dimensional world state. We evaluate the shared-state interface with ten observers and camera-calibrated cross-view residuals.

Conditioning and evaluation. Camera-ray representations and camera-conditioned diffusion provide interfaces for view control (Sitzmann et al., 2021; He et al., 2025). GEN3C conditions

video generation on renderings of a three-dimensional cache reconstructed from input or generated images (Ren et al., 2025). Our geometric and status conditions are computed from the shared engine state. Adapters and LoRA connect these conditions to the pretrained renderer (Zhang et al., 2023; Houlsby et al., 2019; Hu et al., 2022). Per-stream evaluation uses PSNR, SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018), and FVD (Unterthiner et al., 2018) to characterize individual outputs. SymMVC evaluates the spatial relation between outputs using dense matching (Edstedt et al., 2024) and known camera geometry.

3 View-Decoupled World Models

3.1 Shared world, distinct observers

Let K denote the number of simulated actors and $\mathcal{O}$ the set of N rendered observers. The shared world state is

$$s_t = \left(G, \{\xi_t^{(k)}\}_{k=1}^K\right), \qquad \xi_t^{(k)} = \left(p_t^{(k)}, v_t^{(k)}, \psi_t^{(k)}, \phi_t^{(k)}, q_t^{(k)}\right), \tag{1}$$

where G is scene geometry; p and v are position and velocity; ψ and ϕ are yaw and pitch; and q collects the player's status variables. A dynamics operator T advances the state under the joint actor actions. A condition builder C projects that state into observer o's frame, and a renderer R_θ generates its observation:

$$\begin{aligned} s_{t+1} &= T\left(s_t, \{a_t^{(k)}\}_{k=1}^K\right), \\ c_t^{(o)} &= C(s_t, o), \qquad x_t^{(o)} \sim R_\theta\left(\cdot \mid c_t^{(o)}, x_{<t}^{(o)}\right). \end{aligned} \tag{2}$$

Changing the actor population K changes the state advanced by T, whereas adding an observer increases N and adds a query to C and R_θ. Our native demonstration uses $K = N = 10$.

3.2 Independent rendering from a shared trajectory

We call the system *view-decoupled* when all observers receive conditions from one shared state trajectory and each renderer uses only its own condition and visual history. With independent per-observer sampling randomness, the conditional distribution factorizes as

$$p_\theta\left(\{x_{1:L}^{(o)}\}_{o\in\mathcal{O}} \mid s_{1:L}\right) = \prod_{o\in\mathcal{O}} p_\theta\left(x_{1:L}^{(o)} \mid s_{1:L}, o\right). \tag{3}$$

The shared trajectory supplies the common scene realization; the factors generate its observer-dependent appearance. All factors use the same learned parameters. For a fixed trajectory and per-view generation configuration, the serial rendering cost is $O(N)$, and calls can be scheduled independently. Observer-specific conditions are computed on demand, so no fixed view count is encoded in the renderer architecture.

The renderer must express the projected world configuration in its images. Correspondences between outputs should therefore be compatible with their calibrated cameras, a relation measured by SymMVC in Section 6.2.

4 SyncWorld

Figure 1 shows the shared-state generation interface. We use the source game server for T, an observer-plane condition builder for C, and an adapted video-diffusion model for R_θ. The domain provides synchronized videos, exported player state, calibrated cameras, official map geometry, and an instrumentable server.

4.1 Action-driven world state

The server advances the authoritative state at 32 Hz. It receives button and mouse-delta inputs, resolves movement and collision, and exports the realized player states. These exports provide the common source for every observer's condition. The native rollout starts from one injected joint player-state snapshot, including initial velocities, and then uses the server's native bot actions. World evolution is computed once and shared by all view renderers.

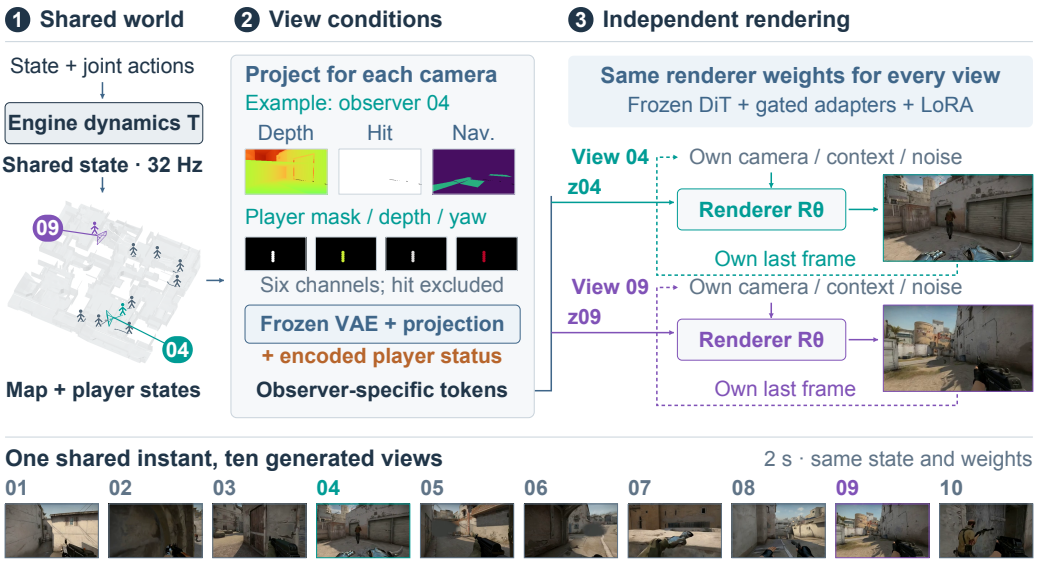

Figure 1: **SyncWorld: one shared world, independently rendered views.** Engine dynamics advance a shared trajectory; per-camera conditions drive independent calls to the same renderer. Dashed paths carry each observer's own context into its next window. Seven condition channels are shown for observer 04; six enter the VAE, excluding hit. Real assets show the historical W5 step75 run at 2 s; actor glyphs are schematic.

4.2 Observer-plane geometric conditioning

For each observer, C rasterizes a seven-channel tensor using the map geometry, player states, and calibrated camera. Three environment channels encode depth, a geometry-hit indicator, and navigation-place semantics. Four player channels encode an occlusion-tested capsule mask, player depth, and observer-relative facing yaw as sine and cosine. Projection uses the player's three-dimensional position and the observer's yaw and pitch, so elevation affects image location and camera depth. Depth testing against environment geometry determines the visible portion of each projected player.

The geometric encoder uses frozen VAE features to represent projected depth, surfaces, player occupancy, and orientation. A separate status encoder projects the observer's alive flag, normalized health, and an opponent-death mask into the same token space. For geometric tensor d and status tensor b, their spatially and temporally aligned outputs form the combined condition:

$$z_{\text{dense}} = E_{\text{geo}}(d), \qquad z_{\text{status}} = E_{\text{status}}(b), \\ z = z_{\text{dense}} + z_{\text{status}}. \tag{4}$$

Camera extrinsics enter through the backbone's Plücker-ray interface. Appendix D specifies channel packing, projection dimensions, and token alignment.

4.3 Shared conditional video renderer

We adapt camera-conditioned LingBot-World (Robbyant Team et al., 2026), a two-expert latent video-diffusion transformer built on Wan (Peebles and Xie, 2023; Rombach et al., 2022; Team Wan et al., 2025). The backbone and VAE remain frozen. After each transformer block, a condition-gated adapter combines its output h_l with the aligned condition z:

$$u_l = \text{SiLU}(A_l h_l) + \text{SiLU}(B_l z), \\ g_l = \tanh(G_l z), \qquad h_l^+ = h_l + O_l(u_l \odot g_l). \tag{5}$$

Here A_l, B_l, and O_l are learned affine maps, G_l is a bias-free linear map, and $\odot$ denotes elementwise multiplication. The condition both supplies features and modulates the fused update through g_l. The

output map O_l starts at zero, so each adapter initially preserves the block output. For comparison, Direct Residual combines hidden and condition features by addition before a single nonlinearity:

$$h_l^+ = h_l + O_l(\mathrm{SiLU}(A_l h_l + B_l z)) . \tag{6}$$

Its output projection is also zero-initialized. The contrast with Equation 5 is the condition-dependent modulation of the fused features; Section 6.1 evaluates the two updates with the same training data and budget.

Rank-64 LoRA adapts attention and feed-forward projections. The condition projections, adapters, and LoRA contain 522M trainable parameters, 1.9% of the two-expert backbone (approximately 14B per expert). All observers reuse these weights.

4.4 Training the conditional renderer

We train the condition projections, gated adapters, and LoRA with flow-matching supervision, while freezing the backbone and VAE. Each noise expert predicts velocity in its assigned noise band, conditioned on geometry, status, camera rays, and the initial observation. Appendix D gives the objective and optimization details.

During data preparation, approximately half the clips are selected deterministically by identifier for initial-frame augmentation with Gaussian blur, JPEG compression, and additive Gaussian noise. The augmented image is re-encoded as context while the target video remains unchanged. Appendix C gives the parameters. At inference, each observer uses its generated context and conditions from the continuing shared trajectory.

4.5 Per-observer autoregressive continuation

Generation uses 81-frame windows at 16 fps. Each observer's final frame becomes the context for its next window, which adds 80 new frames with conditions from the continuing shared trajectory.

On the video sampling grid, window w uses $\mathcal{I}_w = \{80(w-1), \ldots, 80(w-1)+80\}$ for all observers. Temporal synchronization comes from these common state indices, while contexts and sampling randomness remain local to each view. Algorithm 1 in Appendix D specifies the recurrence.

5 A Replay-Derived Multi-Observer Dataset

Shared-state observations. We use data obtained by parsing and rendering CS:GO match replays. The raw collection covers Dust2, Ancient, Mirage, Inferno, and Overpass: 271,568 first-person RGB video files and approximately 48.66 TB including accompanying records. Match, episode, viewpoint, and time metadata link views to shared player states, cameras, actions, and events, supporting synchronized view selection and observer-specific condition construction.

Processed dataset and split. From Dust2, we construct 84,066 distinct clips across 188 match records and 3,885 episodes: 49,066 selected for visibility and temporal coverage, and 35,000 for player events. Each clip has 81 frames at 16 fps and 832×480 resolution, with aligned cameras, spatial conditions, and status records. The approximately 118 cumulative hours are computed as clip count times $81/16$ seconds; windows may partially overlap.

A fixed match hash separates training from 14 held-out matches. Training uses 78,947 distinct clips from 174 matches, approximately 111 cumulative hours. Sampling event clips twice yields 111,342 training records; dataset size counts distinct clips.

Planned public release. The planned release covers the entire five-map raw collection and the processed Dust2 subset with accompanying annotations, supporting alternative sample selection, multi-map studies, and conditional video generation. The experiments reported here use the Dust2 subset. Appendix C gives selection, alignment, formats, and augmentation details.

6 Experiments

We evaluate visual and actor-condition fidelity, then cross-view geometry, concurrent observers, and autoregressive persistence, progressing from per-view quality to shared-world behavior.

Table 1: **Model comparison and condition ablation.** Visual fidelity and actor-level condition fidelity are reported for comparison methods and ablations/interventions. Arrows indicate preferred direction; bold marks the best value across all rows, and underlining marks the strongest non-SyncWorld comparison baseline.

	Visual Fidelity				Actor Condition Fidelity		
Model / Variant	PSNR ↑	SSIM ↑	LPIPS ↓	FVD ↓	Presence Alignment ↑	Position Error ↓	Action Fidelity ↑
Comparison Methods							
LingBot Base	14.4322	0.3605	0.5642	1517.0	0.3747	0.1128	0.1728
Direct Residual	<u>21.3329</u>	<u>0.5955</u>	<u>0.3437</u>	<u>923.2</u>	<u>0.5771</u>	0.0923	<u>0.3426</u>
Vid2World	20.1279	0.5885	0.4470	1655.0	0.2868	<u>0.0712</u>	0.2112
MultiWorld	19.1343	0.5346	0.4337	1410.3	0.4987	0.0973	0.3102
Solaris	18.2699	0.5230	0.4111	1532.9	0.4422	0.0860	0.2260
SyncWorld	**22.5274**	**0.6183**	**0.2956**	838.1	**0.7748**	**0.0611**	**0.4611**
Ablations & Interventions							
w/o Status Conditioning	21.8784	0.6044	0.3137	885.7	0.6487	0.0721	0.4121
w/o Dense Conditioning	21.1081	0.5894	0.3482	890.4	0.6008	0.1155	0.2955
w/o Status & Dense Conditioning	21.0812	0.5888	0.3488	966.3	0.5883	0.1145	0.2445
Time-Shift (+5)	21.3536	0.5804	0.3494	**832.1**	0.5604	0.1159	0.2659
Cross-Group Shuffle	21.1659	0.5730	0.3586	876.2	0.5635	0.1199	0.1799

Experimental setup. Evaluation uses the match-disjoint held-out setup from Section 5. SyncWorld uses an 81-frame, 16 fps, 832×480 delivery configuration, while external baselines retain native interfaces and generation settings. Table 1 reports visual and actor-condition fidelity; protocols and metric definitions are given in Appendices B and E.

6.1 Comparison with video world models and condition ablation

Table 1 contains six comparison methods, three condition ablations, and two alignment interventions, scored for visual and actor-condition fidelity.

Comparison protocol. Direct Residual and SyncWorld form the controlled comparison: they share the LingBot-family backbone interface, target-domain split and training budget, and the same dense geometric and status conditions, differing primarily in direct versus condition-gated structured residual injection. Among external baselines, Vid2World uses its official CS:GO checkpoint for action-conditioned autoregressive prediction, MultiWorld is adapted on the same target-domain split under its native environment–observation interface, and Solaris is evaluated zero-shot with controls mapped to its native keyboard/mouse interface. We retain each system's native interface and preprocessing; full details are given in Appendix E.2.

Overall model comparison. SyncWorld is strongest overall among the Comparison Methods. Direct Residual is the strongest non-SyncWorld baseline on most visual and actor-condition metrics, while Vid2World has the lowest baseline Position Error (0.0712). Together, these results show that visual quality and actor-condition fidelity capture different aspects of model performance.

Matched-training comparison. With interface, split, budget, and conditions matched, Direct Residual versus SyncWorld is the clearest test of condition integration: SyncWorld improves PSNR (21.3329→22.5274), LPIPS (0.3437→0.2956), Presence Alignment (0.5771→0.7748), Position Error (0.0923→0.0611), and Action Fidelity (0.3426→0.4611). This supports the benefit of condition-gated structured residual adaptation under the matched setting.

Condition ablation. The status branch encodes observer status, including alive state, normalized health, and opponent-death masks, whereas the dense geometric branch encodes spatial layout, player occupancy/depth, and observer-relative orientation. Removing status conditioning causes a moderate degradation, whereas removing dense conditioning leads to a larger increase in Position Error and a larger drop in Action Fidelity; removing both further reduces actor-condition fidelity. This pattern is

consistent with the dense geometric condition supporting Action Fidelity indirectly through actor localization, valid crop availability, and spatial cues, rather than through explicit action labels.

Condition alignment interventions. Both Time-Shift (+5) and Cross-Group Shuffle substantially degrade actor-condition fidelity relative to the full SyncWorld: Presence Alignment drops from 0.7748 to 0.5604/0.5635, Position Error increases from 0.0611 to 0.1159/0.1199, and Action Fidelity decreases from 0.4611 to 0.2659/0.1799. Since these interventions retain the full model weights and alter only condition alignment, the degradation supports that SyncWorld relies on correctly aligned structured conditions to control generated actor behavior.

We next measure cross-view geometry with SymMVC.

6.2 Cross-view geometric consistency

Metric. For synchronized images I_i and I_j, RoMa (Edstedt et al., 2024) supplies dense correspondences $\mathcal{M}_{ij} = \{(u_i, u_j)\}$. Known camera intrinsics and camera-to-world transforms define projection functions π_i and π_j. We triangulate X^* from each pair of camera rays and compute

$$r(u_i, u_j) = \tfrac{1}{2}\left(\|\pi_i(X^*) - u_i\|_2 + \|\pi_j(X^*) - u_j\|_2\right). \tag{7}$$

The scorer selects synchronized frames from mutually visible view pairs and filters correspondences by matching confidence, positive triangulated depth in both cameras, and ray-intersection angle. Let $\mathcal{R}_{ij}$ be the pooled retained residuals over these frames and $\mathcal{P}$ the evaluated view pairs. The reported dataset score is

$$S_{ij} = \underset{r \in \mathcal{R}_{ij}}{\text{median}}\, r, \qquad \text{SymMVC}(\mathcal{P}) = \underset{(i,j)\in\mathcal{P}}{\text{median}}\, S_{ij}. \tag{8}$$

The residual is measured in pixels. Pooling within pairs and then taking the median across pairs gives each evaluated pair equal weight in the final score. Lower values indicate closer agreement with the calibrated camera geometry. Applying the same scorer to synchronized real images gives the matching and calibration reference.

Controls and paired evaluation. The Shuffled Geometric Condition control replaces the geometric condition with one from a different time while holding the renderer and evaluation set fixed. The unadapted backbone provides a comparison before learned conditioning. For each reported significance test, we apply a paired Wilcoxon signed-rank test over view pairs and identify the compared control in Table 2.

Table 2: **Cross-view geometric consistency.** Median SymMVC residual across view pairs in pixels; lower is better. Legacy and 480-pixel rows use different configurations. Legacy p-values use Shuffled Geometric Condition; the 480-pixel p-value uses Backbone.

Evaluation	Pairs	Real	Backbone	Shuffled Geometric Condition	SyncWorld	p
Legacy Strict	8	0.078	15.2	4.99	**0.347**	.016
Legacy Extended	15	1.13	19.0	11.2	**1.38**	.00012
Strict, 480 px	8	0.0784	15.13	5.1	**0.5259**	.0078

Results. On legacy Strict, shared-state conditioning lowers the residual from 4.99 under Shuffled Geometric Condition to 0.347 ($p = 0.016$). Legacy Extended shows the same direction, with 1.38 for our model and 11.2 for the same control ($p = 0.00012$). At 480-pixel resolution, our model scores 0.5259, compared with 5.1 for the same control and 15.13 for the backbone ($p = 0.0078$ against the backbone). This comparison links the improvement to the supplied world configuration, while the backbone comparison measures the effect of the adapted renderer as a whole.

Real-image reference. Real synchronized images provide an empirical reference for each evaluation set. They range from 0.0784 on Strict at 480 pixels to 1.13 on legacy Extended. The Strict set contains six pairs with viewing-angle differences of 0–10° and two with differences of 30–60°. Table 3 reports their real-image medians, which differ by approximately 3.6×. Matching and calibration errors therefore vary with pair geometry, motivating comparison against the real-image reference on the same set.

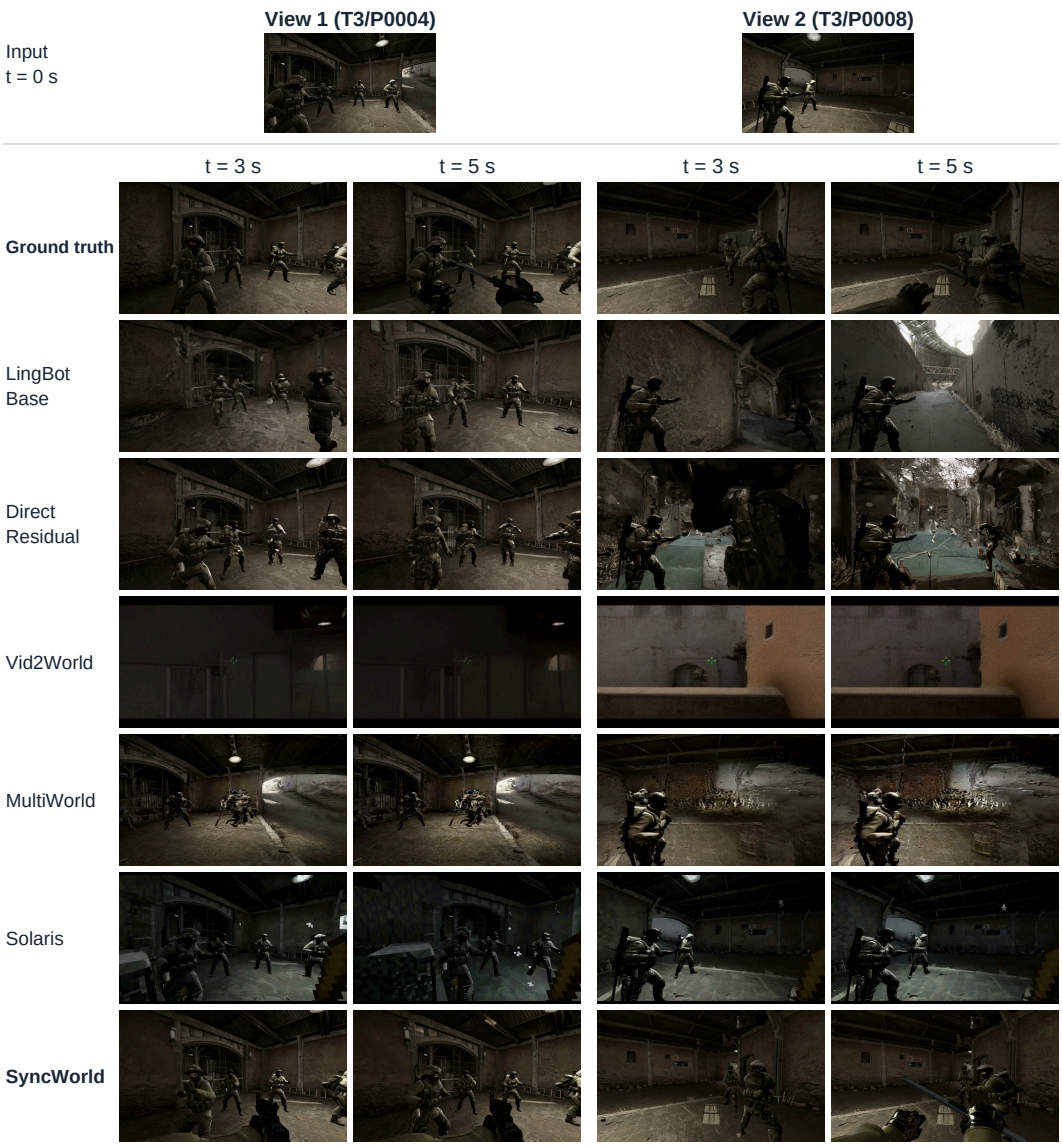

Figure 2: **Qualitative comparison: player positions and visibility.** Two views at 3 and 5 seconds, compared across six methods and ground truth. Top images show the initial references. Columns align view and time; compare scene layout, player location, and visibility against GT. Panels use a common display canvas; Solaris padding is qualitative only.

6.3 Qualitative comparison across methods

Figure 2 compares two synchronized views against LingBot Base, Direct Residual, Vid2World, MultiWorld, and Solaris. In this example, our outputs retain the courtyard layout and the foreground/background player arrangement across both views. View 2 particularly exposes differences in camera framing, scene structure, and player placement. These visual comparisons complement the aggregate metrics in Table 1. External models use their native input interfaces; Appendix H gives those inputs and additional cases.

6.4 One model, ten concurrent observers

We instantiate the observer interface with $N = 10$ on a native shared trajectory. One joint ten-player state snapshot initializes the server, which then advances under native bot actions. The



Figure 3: **Ten observations of one native world trajectory.** A synchronized time step from the ten-observer rollout, arranged as a 2×5 grid. Each view uses its own camera and generation history, with conditions projected from the same authoritative state.

resulting trajectory contains 321 state ticks at 32 Hz and supplies conditions for 161 generated frames per observer at 16 fps. Each observer uses the same renderer and maintains its own two-window generation chain. The run completes all 20 window decodes with zero failures.

Figure 3 shows one synchronized time step across the ten outputs. Each view uses its own camera and context with the common state tick and learned weights. Appendix G summarizes initialization and execution checks.

6.5 Autoregressive temporal persistence

We evaluate continuation with an autoregressive horizon based on LPIPS (Zhang et al., 2018). Let ℓ_n be the distance between generated window n and a piecewise teacher-forced reference, and let σ denote teacher-forced window-to-window variation. The horizon is

$$\mathrm{ARH} = \min\{n : \ell_n - \ell_1 > 2\sigma\} - 1. \tag{9}$$

At the calibrated threshold $2\sigma = 0.0396$, our model reaches $\mathrm{ARH} = 12$ windows. With 80 new frames per chained window after the first, this corresponds to approximately 60 seconds at 16 fps. The recurrence reuses the observer's own generated context and advances its conditions along the state trajectory, as described in Section 4.5. This experiment measures the temporal persistence of the same per-observer generation mechanism used by the multi-observer architecture.

7 Discussion and Conclusion

We presented SyncWorld, a view-decoupled architecture in which joint actions advance one shared world and independent calls to a shared video model render its observations. We also constructed a state-annotated multi-observer dataset and plan to release the five-map raw collection and processed Dust2 clips. Model comparisons show improved visual fidelity, and SymMVC measures stronger cross-view geometric agreement than the Shuffled Geometric Condition and backbone controls. Ten-observer generation and 12-window autoregressive continuation demonstrate the architecture's spatial and temporal use of shared state.

The current evaluation covers Dust2, with 8- and 15-pair cross-view sets; the five-map raw collection provides a basis for future transfer studies across maps, trajectories, and viewing angles. Shared geometry leaves appearance details to individual renderers, and reprojection consistency measures spatial alignment rather than persistent identity or action semantics. The engine supplies world dynamics in this implementation.

A central future direction is to generate the 3D map itself and expand it as exploration proceeds. Incorporating generated geometry into the persistent world state would let observers revisit existing regions and explore new ones through the same rendering interface. This would extend the framework from rendering a fixed environment toward open-ended 3D world generation, with stable spatial structure across revisits and scalable multi-observer rendering as core objectives.

AI USE STATEMENT

During manuscript preparation, generative AI tools assisted with organizing and editing the text, checking consistency between the implementation and its mathematical description, interpreting reported experimental results, and preparing figures and document layout.

REPRODUCIBILITY STATEMENT

Section 5 specifies dataset scale, the match-disjoint split, and our planned release of the five-map raw collection and processed Dust2 data. Appendix C documents selection, alignment, and file formats; Appendices A, B, D, and E give evaluation, renderer, and comparison-protocol details. The trained adapter weights for both noise experts and inference code are available on our anonymous project page. Demonstration videos are available on the anonymous demonstration page.

ETHICS STATEMENT

The source material is obtained by parsing and rendering CS:GO match replays from a commercial game. The work uses gameplay observations and state, without player identity, demographic attributes, or biometric annotations. Generated footage should be identified as generated when released. Game assets, recordings, and base-model weights remain subject to their respective licenses.

REFERENCES

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

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In International Conference on Machine Learning, volume 235 of Proceedings of Machine Learning Research, pp. 4603–4623, 2024.

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

Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Robust dense feature matching. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19790–19800, 2024.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In Advances in Neural Information Processing Systems, pp. 2455–2467, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In International Conference on Learning Representations, 2020.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. Nature, 640(8059):647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling camera control for video diffusion models. In International Conference on Learning Representations, 2025.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pp. 2790–2799, 2019.

Siqiao Huang, Jialong Wu, Qixing Zhou, Shangchen Miao, and Mingsheng Long. Vid2World: Crafting video diffusion models to interactive world models. In International Conference on Learning Representations, 2026.

H. W. Kuhn. The Hungarian method for the assignment problem. Naval Research Logistics Quarterly, 2(1–2):83–97, 1955. doi: 10.1002/nav.3800020109.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In International Conference on Learning Representations, 2022.

Jiaqi Li, Junshu Tang, Zhiyong Xu, Longhuang Wu, Yuan Zhou, Shuai Shao, Tianbao Yu, Zhiguo Cao, and Qinglin Lu. Hunyuan-GameCraft: High-dynamic interactive game video generation with hybrid history condition. arXiv preprint arXiv:2506.17201, 2025.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In IEEE/CVF International Conference on Computer Vision, pp. 9298–9309, 2023.

Xiaofeng Mao, Shaoheng Lin, Zhen Li, Chuanhao Li, Wenshuo Peng, Tong He, et al. Yume: An interactive world generation model. arXiv preprint arXiv:2507.17744, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In IEEE/CVF International Conference on Computer Vision, pp. 4195–4205, 2023.

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.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. GEN3C: 3D-informed world-consistent video generation with precise camera control. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6121–6132, 2025.

Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, et al. Advancing open-source world models. arXiv preprint arXiv:2601.20540, 2026. https://arxiv.org/abs/2601.20540.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10684–10695, 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. arXiv preprint arXiv:2602.22208, 2026. https://arxiv.org/abs/2602.22208.

Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. MVDream: Multi-view diffusion for 3d generation. In International Conference on Learning Representations, 2024.

Vincent Sitzmann, Semon Rezchikov, William T. Freeman, Joshua B. Tenenbaum, and Frédo Durand. Light field networks: Neural scene representations with single-evaluation rendering. In Advances in Neural Information Processing Systems, volume 34, pp. 19313–19325, 2021.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, et al. Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314, 2025.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. arXiv preprint arXiv:1812.01717, 2018.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In International Conference on Learning Representations, 2025.

Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. VideoMAE V2: Scaling video masked autoencoders with dual masking. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14549–14560, 2023.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4): 600–612, 2004. doi: 10.1109/TIP.2003.819861.

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

Jiwen Yu, Yiran Qin, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. GameFactory: Creating new games with generative interactive videos. In IEEE/CVF International Conference on Computer Vision, 2025.

Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. DETRs beat YOLOs on real-time object detection. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 16965–16974, 2024.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In IEEE/CVF International Conference on Computer Vision, pp. 3836–3847, 2023.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

Yifan Zhang, Chunli Peng, Boyang Wang, Puyi Wang, Qingcheng Zhu, Fei Kang, et al. Matrix-Game: Interactive world foundation model. arXiv preprint arXiv:2506.18701, 2025.

A GEOMETRIC EVALUATION DETAILS

A.1 REAL-IMAGE REFERENCES AND PAIR COMPOSITION

Matching, calibration, occlusion boundaries, and view-dependent appearance produce nonzero residuals on synchronized real images. Their score is an empirical reference for each evaluation set. Table 3 gives Strict's angle composition; the scorer also records pair counts and co-visible coverage.

Table 3: Real-image SYMMVC reference by view-angle band in the Strict evaluation.

View-angle band	Pairs	Real-image reference (px)
0–10°	6	0.0609
30–60°	2	0.2178

A.2 RESOLUTION CONVENTION

Legacy videos were generated at 464 rows; the delivery configuration uses 480 rows. When scoring a 464-row frame in the 480-row camera convention, the scorer accounts for the centered 8-pixel vertical offset. In a separate resolution-sensitivity evaluation, the model scored 0.3279 for the former configuration and 0.5259 for the latter, with real-image references of 0.0781 and 0.0784. Generation resolution and samples changed together in that evaluation. These measurements document the scorer convention and sensitivity to the evaluation configuration.

A.3 CORRESPONDENCES AND AGGREGATION

The scorer applies the correspondence filters and pair aggregation in Section 6.2. It records match confidence, geometric support, and frame-processing status alongside each residual. Paired significance values use the controls named in Table 2.

Frame selection and camera projection. Pair eligibility uses the number of mutually visible frames and the projected player support in both viewing directions. Candidate synchronized frames are ranked by the sum of their two directional visibility counts. The scorer uses RoMa's outdoor matcher and converts sampled correspondences to pixel coordinates at each decoded frame's dimensions. Frames with fewer than eight matches after confidence filtering are skipped.

For camera-to-world rotation R and camera center t, the world-to-image projection matrix is

$$P = K_\delta[R^\top \mid -R^\top t], \qquad K_\delta = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y - \delta \\ 0 & 0 & 1 \end{bmatrix}, \tag{10}$$

where $\delta = (480 - H)/2$ is computed from decoded frame height H. OpenCV triangulates homogeneous points using the two projection matrices; dehomogenized points are retained when their depths are positive in both cameras and their triangulation angle passes the filter. Pair view angles are computed separately from the cameras' forward axes and summarized by their median over evaluated frames.

Support coverage. Correspondence retention is the fraction of confidence-passing RoMa correspondences that remain after the geometric validity filters and therefore contribute to SYMMVC. It is a metric-support diagnostic, not a generation-accuracy metric. Pair-level and frame-level coverage remain near-complete across all three evaluation configurations, while correspondence support differs across methods.

Table 4: SYMMVC correspondence support after geometric filtering. Values are the fractions of confidence-passing RoMa correspondences retained after the geometric validity filters; they measure metric support rather than generation accuracy.

Method	Legacy Strict	Legacy Extended	Strict, 480 px
Real	91.42%	85.01%	91.14%
SyncWorld	86.61%	82.92%	85.87%
Shuffled Geometric Condition	89.40%	72.67%	86.05%
Backbone	66.75%	59.46%	65.29%

Under Legacy Strict, all methods retain 8/8 valid view pairs; frame coverage is 64/64 for Real, 63/64 for Backbone, 64/64 for Shuffled Geometric Condition, and 64/64 for SyncWorld. Under Legacy Extended, all methods retain 15/15 valid view pairs; frame coverage is 120/120 for Real, 118/120 for Backbone, 120/120 for Shuffled Geometric Condition, and 120/120 for SyncWorld. Under Strict, 480 px, all methods retain 8/8 valid view pairs; frame coverage is 64/64 for Real, 63/64 for Backbone, 64/64 for Shuffled Geometric Condition, and 64/64 for SyncWorld. Thus, frame-level support is at least 98.3% across methods and evaluation configurations.

Missing observations and paired statistics. The implementation records missing videos, missing frames, insufficient matches, and retained residual counts separately. Frame coverage counts frames that complete decoding and confidence-filtered matching; the retained-point count records support after geometric filtering. An empty residual pool has no numerical score. A paired comparison includes only pairs with scores for both methods. The signed-rank test is two-sided and excludes zero differences. Bootstrap intervals resample paired differences with replacement 5,000 times using seed 123, taking the 2.5th and 97.5th percentiles of the resampled medians.

B ACTOR-CONDITION FIDELITY METRICS

Beyond image fidelity and cross-view geometry, we evaluate whether the rendered actors follow the supplied state. Presence Alignment, Position Error, and Action Fidelity separately measure actor presence, image-plane position, and action/state realization. All three metrics share a frozen actor-detection and GT-anchored correspondence pipeline.

B.1 ACTOR EXTRACTION AND CORRESPONDENCE

For evaluation clip v and future frame t, let $\mathcal{G}_{v,t}$ denote the canonically visible ground-truth actors. Each actor a has a canonical image-space bounding box $B^{\mathrm{GT}}_{v,a,t}$ obtained from the actor/world state, calibrated camera intrinsics and extrinsics, and the canonical projection procedure. On each generated RGB frame, the frozen detector produces anonymous detections

$$\mathcal{D}_{v,t} = \{B^{\mathrm{det}}_{v,1,t}, \ldots, B^{\mathrm{det}}_{v,n,t}\}. \tag{11}$$

The detector does not assign actor identity; identity is established only by matching detections to the GT boxes.

Matching is performed independently at each future frame. For GT box i and detection j, let $c(\cdot)$ denote the box center and define

$$d^{\mathrm{ctr}}_{ij} = \frac{\|c(B^{\mathrm{GT}}_i) - c(B^{\mathrm{det}}_j)\|_2}{\sqrt{W^2 + H^2}}, \qquad C_{ij} = d^{\mathrm{ctr}}_{ij} + \left(1 - \mathrm{IoU}(B^{\mathrm{GT}}_i, B^{\mathrm{det}}_j)\right). \tag{12}$$

The Hungarian algorithm (Kuhn, 1955) obtains a frame-wise one-to-one assignment. An assignment (i, j) is accepted as a valid actor correspondence only if

$$d^{\mathrm{ctr}}_{ij} \le \tau_{\mathrm{ctr}}, \qquad \tau_{\mathrm{ctr}} = 0.15, \tag{13}$$

where this threshold is fixed before world-model evaluation. Assignments exceeding this normalized center-distance threshold are rejected and treated as unmatched. An accepted pair assigns the GT actor identity to the generated detection; an unmatched GT actor is a missing actor observation, and an unmatched generated detection is an unsupported additional actor observation. This frame-wise matching is shared by Presence Alignment, Position Error, and the generated actor crops used

by Action Fidelity. Implementation and training details of the frozen actor detector are given in Appendix B.5. Future evaluation uses frames $t = 1, \ldots, 80$; the conditioning frame $t = 0$ is excluded. The detector itself performs no identity recognition, long-term tracking, action recognition, or cross-view re-identification.

B.2 Presence Alignment

For clip v, accumulate over future frames the accepted correspondences TP_v, unmatched GT actors FN_v, and unmatched generated detections FP_v defined above. The clip-level Presence Alignment score is

$$S_{\text{pres}}^{(v)} = \frac{TP_v}{TP_v + FP_v + FN_v}, \qquad S_{\text{pres}} = \frac{1}{N} \sum_{v=1}^{N} S_{\text{pres}}^{(v)}. \tag{14}$$

Higher values are better. The score jointly penalizes missing expected actors, unsupported additional actors, and temporal appearance or disappearance errors insofar as they change the frame-wise actor set.

B.3 Position Error

Position Error is conditional on successful actor correspondence. For a matched actor a in clip v at frame t, define the normalized center error

$$e_{v,a,t}^{\text{pos}} = \frac{\left\| \mathbf{c}_{v,a,t}^{\text{pred}} - \mathbf{c}_{v,a,t}^{\text{GT}} \right\|_2}{\sqrt{W^2 + H^2}}, \tag{15}$$

where the centers are those of the matched generated and canonical GT boxes. Let $\mathcal{M}_v$ be the accepted matched actor-frame observations. The clip and macro scores are

$$S_{\text{pos}}^{(v)} = \frac{1}{|\mathcal{M}_v|} \sum_{(a,t) \in \mathcal{M}_v} e_{v,a,t}^{\text{pos}}, \qquad S_{\text{pos}} = \frac{1}{N_{\text{valid}}} \sum_{v:\, |\mathcal{M}_v| > 0} S_{\text{pos}}^{(v)}. \tag{16}$$

Lower values are better. Missing actors are handled by Presence Alignment; Position Error measures spatial fidelity conditional on successful correspondence and therefore assigns no arbitrary position penalty to a missing actor.

B.4 Action Fidelity

Action Fidelity evaluates GT-positive action and state queries from the label space

$$\mathcal{E} = \{\text{crouch}, \text{reload}, \text{fire}, \text{throw}, \text{corpse}\}. \tag{17}$$

The five action/state categories correspond to the output heads of the frozen action evaluator described in Appendix B.5. The categories are not mutually exclusive: fire denotes firing, throw denotes grenade release, reload denotes the reload process, crouch denotes the crouched state, and corpse denotes a visibly rendered dead-body state. For event e, let $y_{q,e}$ be its GT label and define the positive query set $Q_e^+ = \{q : y_{q,e} = 1\}$. Each query fixes an actor, event or state type, the relevant GT event annotation, and a derived positive query support $\mathcal{T}_e^+$ containing exactly eight evaluator timestamps. The query universe and positive query support are fixed from GT annotations before evaluation and are shared across all compared routes.

The eight evaluator timestamps are defined from GT event annotations according to the following query-support rules. For fire, with $L_{\text{fire}} = t_{\text{end}} - t_{\text{start}} + 1$, the anchored eight-frame window itself defines $\mathcal{T}_{\text{fire}}^+$ when $L_{\text{fire}} \leq 8$ and is not required to remain within the raw annotated fire interval; when $L_{\text{fire}} > 8$, eight frames are uniformly sampled from $[t_{\text{start}}, t_{\text{end}}]$. Throw uses eight consecutive frames beginning at the grenade-release anchor. Reload and crouch uniformly sample eight frames from their complete GT intervals, and corpse uniformly samples eight frames from its GT-visible

corpse interval:

$$\begin{aligned}
\mathcal{T}^{+}_{\text{fire}} &= \{t_{\text{anchor}}, \dots, t_{\text{anchor}} + 7\} \quad (L_{\text{fire}} \le 8), \\
\mathcal{T}^{+}_{\text{fire}} &= \text{UniformSample}([t_{\text{start}}, t_{\text{end}}], 8) \quad (L_{\text{fire}} > 8), \\
\mathcal{T}^{+}_{\text{throw}} &= \{t_{\text{anchor}}, \dots, t_{\text{anchor}} + 7\}, \\
\mathcal{T}^{+}_{e} &= \text{UniformSample}([t_{\text{start}}, t_{\text{end}}], 8), \quad e \in \{\text{reload}, \text{crouch}\}, \\
\mathcal{T}^{+}_{\text{corpse}} &= \text{UniformSample}([t_{\text{corpse-start}}, t_{\text{corpse-end}}], 8).
\end{aligned} \tag{18}$$

For the interval-based cases (long Fire, Reload, Crouch, and Corpse), deterministic temporal resampling with repetition is used when fewer than eight distinct evaluator frames are available; both sampling and repetition remain within the corresponding annotated interval, which is not extended merely to obtain eight unique frames. Generated video is sampled at the timestamps in the fixed $\mathcal{T}^{+}_{e}$, and the frame-wise correspondence in Appendix B.1 retrieves the crop of the GT-specified actor. A route cannot shift a query to a different time at which it happens to realize the action. For model m and query q, let $K_{m,q} \in \{0, \dots, 8\}$ be the number of fixed query timestamps at which an accepted actor correspondence provides a valid generated crop. After $\mathcal{T}^{+}_{e}$ has been fixed, when $6 \le K_{m,q} < 8$, only the available crops at their original query timestamps are deterministically temporally resampled or repeated within that already-defined query support to form the required eight-frame input; no crop from outside the query, black frame, or shifted action time is introduced. Thus, at least six observations retain at least 75% of the positive query support, while isolated detection or correspondence misses are tolerated.

For model m, query q, and event e, let $p_{m,q,e} \in [0, 1]$ be the frozen action model's sigmoid probability for the target head. The penalty-aware query score is

$$\widetilde{p}_{m,q,e} = \begin{cases} \dfrac{K_{m,q}}{8}\, p_{m,q,e}, & K_{m,q} \ge 6, \\ 0, & K_{m,q} < 6. \end{cases} \tag{19}$$

Thus, 8/8 observations retain the full action probability, while 7/8 and 6/8 observations receive factors of $7/8$ and $6/8$, respectively. Queries with fewer than six observations remain in the fixed GT-positive denominator and receive zero. For each supported event, and then across events, we report

$$S_e(m) = \frac{1}{|Q^{+}_{e}|} \sum_{q \in Q^{+}_{e}} \widetilde{p}_{m,q,e}, \qquad S_{\text{act}}(m) = \frac{1}{|\mathcal{E}_{+}|} \sum_{e \in \mathcal{E}_{+}} S_e(m), \quad \mathcal{E}_{+} = \{e \in \mathcal{E} : |Q^{+}_{e}| > 0\}. \tag{20}$$

All supported event types receive equal weight, rather than being micro-averaged according to event frequency. Higher values are better.

B.5 Supporting Models and Calibration

The actor detector is a pretrained RT-DETR (Zhao et al., 2024) with an R50VD backbone, fine-tuned end-to-end on real RGB frames from the training split using canonical GT actor boxes obtained from actor state and calibrated camera projection. Detector train, validation, and held-out partitions are match-disjoint. The detector has one primary actor-detection class; actor identity is not a detector class. Fine-tuning keeps the R50VD backbone trainable with a smaller learning rate than the hybrid encoder, transformer decoder, and detection heads, freezes backbone normalization layers, and uses the standard classification, box-regression, and GIoU objectives. Checkpoint and confidence-threshold selection use validation data only. Before formal world-model comparison, the detector checkpoint, confidence threshold, preprocessing, input resolution, box postprocessing, NMS/filtering, matching protocol, and spatial gate are frozen and then shared by all three metrics.

On held-out real reference RGB, the frozen detector is matched to canonical GT boxes with the same correspondence procedure. Its detector-only normalized center localization error is

$$E_{\text{det}} = \frac{\left\| c(B^{\text{det}}) - c(B^{\text{GT}}) \right\|_2}{\sqrt{W^2 + H^2}}. \tag{21}$$

This detector localization floor is a diagnostic reference for Position Error attributable to the learned detector; it is not subtracted from model scores.

The action model uses VideoMAE V2 Base (Wang et al., 2023) on an actor-centric temporal crop with exactly eight sampled RGB frames. A shared VideoMAE feature extractor is followed by $\mathrm{LayerNorm}(D)$ and a $\mathrm{Linear}(D, 5)$ head producing independent logits for crouch, reload, fire, throw, and corpse; each head uses a sigmoid rather than a softmax. Training uses real RGB actor crops produced with the frozen detector and GT-anchored matching, supervised by the corresponding GT action/state query. In Stage A, the VideoMAE backbone is frozen and the normalization and five-head prediction layer are trained. In Stage B, the final two encoder blocks are unfrozen and trained with those layers while earlier blocks remain frozen. After model selection, the action-model checkpoint is frozen and applied unchanged to every compared generation route.

Held-out evaluator validation. The frozen detector and action evaluator are evaluated on real RGB from a match-disjoint held-out evaluation split whose matches are disjoint from evaluator training and model-selection data. All detector and action-evaluator checkpoints, confidence thresholds, crop transforms, temporal sampling rules, and matching rules are frozen before this evaluation, and no generated world-model outputs are used to tune either evaluator. Detector validation measures detection and localization reliability, while action-evaluator validation reports per-head discrimination and classification quality. Formal world-model evaluation later applies the same frozen evaluators to generated RGB, so real-to-generated domain shift remains a limitation.

Table 5: Held-out detector validation on real RGB.

Detector metric	Value
Precision	0.8495
Recall	0.8295
AP	0.8127
AP50	0.8639
Mean IoU	0.9362
Norm. Center Error ↓	0.0015

Mean IoU and normalized center error are computed over matched detections.

Table 6: Held-out action-evaluator validation on real RGB.

Action	Pos.	Neg.	AUROC	AUPRC	Precision	Recall	F1
Crouch	521	1042	0.9376	0.8917	0.8516	0.8484	0.8500
Reload	59	118	0.8837	0.8254	0.8276	0.8136	0.8205
Fire	193	386	0.9003	0.8597	0.8457	0.8238	0.8346
Throw	122	244	0.9279	0.8827	0.8655	0.8443	0.8548
Corpse	144	288	0.8924	0.8329	0.8429	0.8194	0.8310
Macro-F1	–	–	–	–	–	–	0.8382

Pos./Neg. denote the numbers of valid GT-positive and GT-negative queries after the frozen query-construction and validity rules.

C Dataset Construction and Formats

C.1 Source collection and counting conventions

The five-map raw collection contains 271,568 first-person RGB video files, approximately 48.66 TB including accompanying records. Its Dust2 subset contains 188 match records, 3,982 episodes, and 39,626 RGB videos, approximately 7.82 TB. Storage uses decimal units. Processed clips are selected from 3,885 of these Dust2 episodes.

Table 7 separates distinct clips from training-manifest records. Duration is cumulative clip playback, $n \times 81/16$ seconds for n clips. It sums player-view windows, including partial temporal overlap; it

does not measure non-overlapping match time. The visibility-selected pool is included once in the combined dataset.

Table 7: Processed Dust2 dataset composition. Training records count event oversampling; dataset size counts distinct clips.

Selection branch	Full dataset: clips	Training: clips	Training: records
Visibility and coverage	49,066	46,552	46,552
Event-focused	35,000	32,395	64,790
Total	84,066	78,947	111,342

C.2 Visibility and event selection

The visibility branch generates candidate windows along player recordings and checks other-player visibility at 21 temporal positions. Candidates are scored by the number of visible positions and projected player area. Round-robin traversal of groups defined by match, episode, and temporal bin balances coverage. The resulting pool retains 14,222 initial clips and adds 34,844 clips.

The event branch combines actions, visibility, and world-event records using programmatic rules. It selects death, kill, firing, weapon-switch, grenade-operation, sniper or scope-related, and high-motion covisibility cases, placing selected events near window centers. The two branches are merged using match, episode, player viewpoint, and source start frame to identify distinct clips.

C.3 Temporal alignment and stored records

Each selected window is sampled every two source frames to obtain 81 RGB frames at 16 fps, resized to 832×480. Camera poses use OpenCV camera-to-world transforms. The frozen video encoder produces 21 latent-time positions: RGB positions $0, 4, \ldots, 80$ correspond to source-frame offsets $0, 8, \ldots, 160$. Spatial and status conditions use these same positions. Shared match, episode, viewpoint, and source-time metadata retain the relation between each clip and the replay timeline.

Raw MP4 videos are accompanied by JSON player-state, action, and camera records and JSONL world-event records. Processed samples associate RGB clips, initial images, text conditions, camera parameters, video latents, and geometric and status conditions through JSONL records. Camera arrays have shapes $81 \times 4 \times 4$ for poses and 81×4 for intrinsics. PyTorch caches hold video latents of shape $16 \times 21 \times 60 \times 104$ and initial-image conditions of shape $20 \times 21 \times 60 \times 104$, ordered as channels, time, height, and width.

Each spatial timestamp is a compressed NPZ containing a seven-channel float32 array at 176×320 resolution; a clip references 21 such timestamps. Status NPZ files contain length-21 arrays for ego alive status and normalized health, plus a $21 \times 60 \times 104$ opponent-death-location mask. The channel construction and six-channel VAE packing are specified in Appendix D.

C.4 Initial-frame augmentation

Approximately half the clips are selected deterministically by identifier for initial-frame augmentation. Gaussian blur, JPEG compression, and additive Gaussian pixel noise are applied sequentially, after which the image condition is re-encoded. Blur standard deviation is sampled uniformly from 1 to 3, JPEG quality uniformly from integers 30 to 60, and noise standard deviation uniformly from 2 to 6 in the 0–255 pixel domain. Values are clipped to the valid pixel range. Augmentation assignment is fixed during preparation, and target-video supervision remains unchanged.

D SyncWorld Implementation Details

D.1 Condition construction

The rasterizer output is $d = (d_0, \ldots, d_6)$: environment depth, geometry-hit indicator, navigation-place semantics, player mask, player depth, relative-yaw sine, and relative-yaw cosine. Official

collision geometry and navigation-mesh data define the environment channels. The place channel encodes visible navigation-place identifiers and a separate value for geometry outside visible navigation areas. Capsules are placed at the players' three-dimensional positions and projected through the observer's camera. Their visible pixels are selected by comparing player depth with environment depth. For player k and observer o, the orientation channels encode $\sin(\psi^{(k)} - \psi^{(o)})$ and $\cos(\psi^{(k)} - \psi^{(o)})$ on the projected player region.

D.2 Condition encoding and injection

Let $P_1(d)$ and $P_2(d)$ denote the two pseudo-RGB inputs packed from d, and let V be the frozen Wan2.1 VAE encoder (Team Wan et al., 2025). Each input produces a 16-channel latent feature map. Concatenating the maps and applying a learned projection gives

$$E_{\text{geo}}(d) = W_{\text{geo}} \,\text{Concat}[V(P_1(d)), V(P_2(d))] + b_{\text{geo}}, \tag{22}$$

Here W_{geo} projects the concatenated 32 feature channels to 128. The status encoder is a zero-initialized affine projection from three channels to the same dimension. Its alive and health channels are constant spatial planes, while the death mask carries image-plane locations. The two branches therefore align spatial evidence with auxiliary status before injection.

The encoder packs the rasterized channels into two pseudo-RGB inputs:

$$P_1(d) = 2[d_0, d_2, d_3] - 1, \qquad P_2(d) = 2[d_4, d_5, d_6] - 1. \tag{23}$$

Brackets denote channel concatenation. This packing uses six of the rasterizer's seven channels; the exported geometry-hit indicator d_1 is not a separate VAE input. The affine value map is applied to both packed inputs, including the signed yaw channels.

VAE features are resized to the transformer's token grid when necessary. Nearest-neighbor sampling aligns status maps to that grid before their $3 \to 128$ projection. Flattening the frame and spatial dimensions gives aligned geometric and status sequences for Equation 4.

In Equation 5, z has 128 channels and the internal features u_l and gate g_l have 512. The output projection restores the block's hidden dimension. Adapters follow every DiT block, and LoRA acts on self-attention, cross-attention, and feed-forward projections. The additive control uses Equation 6. Table 8 summarizes the full renderer.

Table 8: Renderer and generation configuration.

Component	Configuration
Frozen backbone	LingBot-World camera-conditioned; two noise experts, approximately 14B each
Trainable parameters	522M (1.9%)
Condition encoder	Frozen Wan2.1 VAE; two 16-channel features; learned $32 \to 128$ projection
Residual adaptation	Condition-gated update after each DiT block; internal width 512; zero-initialized output
LoRA	Rank 64; attention and feed-forward projections
Geometric condition	Seven rasterized channels; six packed as in Equation 23
Auxiliary status	Alive, health, opponent-death mask; zero-initialized $3 \to 128$ projection
Camera	Plücker-ray interface
Generation window	81 frames at 16 fps; 80 new frames per continuation
Delivery resolution	832×480 pixels

D.3 Optimization and autoregressive context

Adaptation uses flow-matching supervision. For a clean video latent y_0, Gaussian noise ϵ, and noise level τ, the noisy input and target velocity are

$$y_\tau = (1 - \tau)y_0 + \tau\epsilon, \qquad v^\star = \epsilon - y_0. \tag{24}$$

Low- and high-noise experts receive samples from their respective noise bands. The model predicts the target velocity conditioned on the geometric and status tokens, camera rays, and an observation context frame. Optimization updates the condition projections, residual adapters, and LoRA parameters while keeping the backbone and VAE frozen.

The trainer uses AdamW with one parameter group for trainable residual adapters, geometric projections, and the status projector, and a separate group for LoRA. Each microbatch loss is divided by the accumulation count before backpropagation, and one optimizer update follows the accumulated microbatches. Frozen backbone and VAE parameters are excluded from these groups.

With one shared boundary frame between adjacent windows in Algorithm 1, a chain of m windows contains $81 + 80(m-1)$ frames. Thus 12 windows contain 961 frames, approximately 60 seconds at 16 fps. Equation 9 defines the temporal evaluation's threshold and horizon.

D.4 Independent observer rollout

Algorithm 1 Shared-state generation with independent observer contexts

Require: Shared state trajectory $s_{0:80m}$ advanced by T under joint actions; observers $\mathcal{O}$; initial images $\{\chi^{(o)}\}$; shared renderer R_θ

Ensure: Synchronized observation streams $\{X^{(o)}\}_{o\in\mathcal{O}}$

Initialize $X^{(o)} \leftarrow [\,]$ for each observer.
for $w = 1, \dots, m$ **do**
  Set $\mathcal{I}_w \leftarrow \{80(w-1), \dots, 80(w-1)+80\}$.
  for each observer $o \in \mathcal{O}$ independently **do**
    Project $c_t^{(o)} \leftarrow C(s_t, o)$ for each $t \in \mathcal{I}_w$.
    Sample $Y_w^{(o)} \sim R_\theta(\,\cdot \mid c_{\mathcal{I}_w}^{(o)}, \chi^{(o)})$.
    if $w = 1$ **then**
      Append all 81 frames of $Y_w^{(o)}$ to $X^{(o)}$.
    else
      Append the 80 new frames, omitting the shared boundary frame.
    end if
    Update $\chi^{(o)} \leftarrow \mathrm{last}(Y_w^{(o)})$.
  end for
end for

E Comparison and Evaluation Protocol

We separate controlled architectural comparisons from external-system comparisons. Evaluation is performed on a common held-out evaluation manifest. External models retain their native input and rollout interfaces, while model-specific inputs and outputs are adapted to a common evaluation protocol. Training and adaptation status is disclosed explicitly as target-domain adaptation status rather than being presented as identical training histories across systems.

E.1 Evaluation Set Construction

The full processed Dust2 dataset contains 84,066 distinct clips. The training subset contains 78,947 clips from 174 matches. The remaining held-out pool contains 5,119 clips from 14 matches that are match-disjoint from training. The formal evaluation set is selected exclusively from this held-out metadata before model inference.

We apply a deterministic, match-balanced procedure to construct a fixed formal evaluation manifest of 1,024 clips. Twelve held-out matches contribute 73 clips each and two contribute 74 clips each, so $12 \times 73 + 2 \times 74 = 1{,}024$. The selector jointly covers the two data-source branches: the resulting cohort contains 503 visibility/coverage samples and 521 event-focused samples. Candidate ordering and tie-breaking use a fixed SHA256-based ordering, so the same held-out pool and selection rules reproduce the same subset.

The selector rejects repeated canonical samples, defined by the same game, episode, observer, and raw start time, and suppresses excessive within-view temporal concentration. For candidates from the same audited viewpoint, start-frame gaps below 81 frames are rejected, gaps of at least 161 frames are preferred, and gaps from 81 to 160 frames may be accepted when needed. This rule does not prohibit synchronized observations from different viewpoints from covering the same physical interval. The final cohort has 1,024 unique clip identifiers and no duplicate canonical keys. The resulting manifest is frozen before inference and reused across all compared methods, with no model-specific evaluation-subset selection.

E.2 Model Inputs, Training, and Adaptation

The comparison distinguishes a controlled architectural comparison from native-interface comparisons to external systems. The controlled comparison holds the LingBot-family backbone interface, underlying target-domain training split, and training budget fixed between Direct Residual and SyncWorld. External systems retain their native model interfaces and preprocessing, while their target-domain adaptation status is reported explicitly.

LingBot Base. We use the original camera-conditioned LingBot-World checkpoint as the pretrained backbone reference. This route does not receive the SyncWorld target-domain adaptation, geometry/status adapters, or the additional target-domain training used by the controlled comparison. It therefore measures the performance of the underlying pretrained video-world-model backbone before the proposed adaptation.

Direct Residual. Direct Residual uses the same LingBot-family backbone, the same underlying 174-match Dust2 training split, and the same target-domain training budget as SyncWorld. It receives the same geometry and status conditioning signals, but injects them through a direct residual pathway rather than the condition-gated residual adaptation used by SyncWorld. This route therefore serves as the controlled architectural baseline for isolating the effect of the proposed adaptation mechanism.

SyncWorld. SyncWorld is initialized from the LingBot-World backbone and adapted on the 174-match Dust2 training split, with no held-out evaluation data used for training. The model incorporates geometry and status conditioning through condition-gated residual adapters together with LoRA-based adaptation. The target-domain training configuration and compute budget are matched to Direct Residual for the controlled architectural comparison.

Vid2World. We use the official CS:GO checkpoint released for Vid2World, which is already trained in the target game domain by the original authors; no additional Dust2 fine-tuning is applied in our experiments. We preserve its native preceding-frame/action modeling interface rather than converting it to the LingBot-family conditioning format. Consequently, Vid2World serves as an external target-domain baseline whose prior training history differs from the controlled LingBot-family comparison.

Solaris. We use the official pretrained Solaris checkpoint without additional CS:GO/Dust2 target-domain adaptation. The Dust2 controls are mapped to Solaris's native keyboard-and-mouse action representation, while the model retains its own observation and action interface. Solaris therefore evaluates out-of-domain transfer under a native external-system interface rather than a matched target-domain training setting.

MultiWorld. MultiWorld is initialized from its official pretrained checkpoint and further adapted using the same underlying 174-match Dust2 training split, without using held-out evaluation data. To better approximate its native high-rate temporal sampling without introducing synthetic interpolation, we train from the original raw 32-FPS Dust2 videos using 81 consecutive frames with unit temporal stride, comprising 80 future intervals and yielding a 2.5-s future horizon. RGB frame $s + t$ is paired directly with the action record at the same raw-frame index, without temporal interpolation or multi-frame action aggregation. We retain MultiWorld's native single-view training target and paired environment-observation interface, and optimize the DiT, action encoder, and environment-encoder connector while keeping the remaining components frozen. We match its adaptation budget in GPU-hours to the controlled target-domain training budget.

E.3 Inference Interface Alignment

All methods are evaluated from the same underlying held-out evaluation samples, but heterogeneous models require different native inference inputs. We therefore use model-specific input adapters that translate each canonical evaluation sample into the valid representation expected by the corresponding model. These adapters preserve each model's original inference semantics rather than forcing all systems to share an identical internal interface. In particular, models may retain their own context length, spatial preprocessing, action representation, auxiliary conditioning, rollout chunking, native temporal rate, and sampler or integration semantics. No model receives future ground-truth RGB observations during prediction. We therefore align the underlying evaluation samples and prediction task, rather than forcing heterogeneous models to share the same internal inference mechanism.

LingBot-family models. The LingBot-family variants share the same camera-conditioned backbone interface. The canonical initial RGB observation is provided at the 832×480 model-facing resolution; no additional cropping or padding is applied. Per-frame camera poses and intrinsics are converted to the backbone's camera-conditioning representation, while the structured geometric and status conditions are temporally aligned with the corresponding video-latent sequence. Base does not use the additional geometry/status conditioning; Direct Residual injects those conditions through direct additive residual injection, whereas SyncWorld uses the condition-gated residual-adapter pathway. Zero-Status, Zero-Dense, and Zero-Both inherit the same SyncWorld input pipeline and disable only the designated conditioning branches. Time-Shift (+5) and Cross-Group Shuffle use the same inference machinery while perturbing only the specified conditioning source.

Vid2World. Vid2World inputs are constructed from the permitted pre-prediction observations according to its native preceding-frame and action interface. The adapter loads nine RGB history frames, applies the model's official spatial transform, and appends a copy of the last history frame as the current placeholder. It also converts the canonical action sequence to Vid2World's native 51-dimensional action representation, with action slices aligned to the recurrent prediction calls; the underlying control trajectory is unchanged.

The model-facing spatial transform is applied as $\mathrm{Resize}(275, 512)$ followed by $\mathrm{CenterCrop}(320, 512)$, using the height–width convention of the implementation. Because the requested crop is taller than the resized image, this official crop operation pads the vertical extent (22 pixels above and 23 below for the resulting 512 (W) $\times$ 320 (H) model-facing frame). This is Vid2World's model-facing preprocessing, not an evaluator-side transform. Native inference preserves the original recurrent one-frame autoregressive rollout at 4 fps: each prediction remains in the native 512 (W) $\times$ 320 (H) representation and is fed back directly as the next context frame. No evaluation-specific reverse transform or temporal resampling is inserted into the loop, and future ground-truth RGB frames are never used as recurrent conditioning.

MultiWorld. MultiWorld's paired view/action input is constructed from two synchronized observations of the same canonical evaluation sample. For the video input branch, each 832×480 observer frame is bicubic-resized to the native 480×480 representation without cropping or padding. The separate environment encoder retains its own native preprocessing. The two views are assembled as a joint 960×480 model input. The canonical action sequence is mapped to the native representation with a 10-way discrete action vector and a 2-dimensional continuous control vector for each agent at each native time step.

MultiWorld preserves its native chunked autoregressive rollout. Fixed-size native chunks use one shared boundary frame between adjacent chunks; after a chunk, the final generated frame from each view is used as the next image context and the environment observation is rebuilt from those generated views. The native action sequence is sliced according to the same overlapping chunk convention. Inference remains at the native 60 fps rate, and the internal rollout dynamics are not altered by the evaluation adapter. No future ground-truth RGB frames are used as rollout conditioning.

Solaris. Solaris's native observation/action interface is constructed from synchronized conditioning observations and mapped controls. The adapter repeats each initial RGB observation across the native temporal input and converts the canonical control sequence to Solaris's native keyboard-and-mouse action representation; the underlying model separates 23 keyboard channels and 2 mouse channels. No camera pose, dense condition, or status sidecar is added by the adapter.

Solaris applies its official model-facing preprocessing: it center-crops each input to the target $640 : 352$ aspect ratio and resizes the crop to 640×352 with JAX linear interpolation, followed by the model's input normalization. For an 832×480 source, this retains approximately the central 832×457 region before resizing. The native causal multiplayer rollout is preserved without altering its internal temporal mechanism. Future ground-truth RGB observations are not used during rollout.

Across methods, we standardize the underlying held-out sample identity, the prediction task, the permitted conditioning information, and the future interval being predicted. We preserve model-specific context length, model-facing resolution, action parameterization, native rollout rate, autoregressive chunking, and sampler or integration mechanics. Temporal timestamp alignment and interface-normalized spatial processing for metric computation are described separately in Sec. E.4.

E.4 Output Normalization and Metric Computation

After native inference, each prediction is converted to a valid scoring representation before metric computation. Paired fidelity metrics use a shared temporal support, while deterministic spatial normalization follows the valid output or interface representation of each model. This avoids charging a model for resize, crop, or padding operations that are imposed by a heterogeneous model interface rather than by prediction error. Thus, the protocol standardizes the underlying evaluation content, physical timestamps, metric implementation, and aggregation, while applying the corresponding deterministic transform consistently to a prediction and its target.

Common temporal alignment. Paired metrics evaluate the same five-second future interval at the 20 physical timestamps $t_k = k/4$ seconds for $k = 1, \ldots, 20$, namely $0.25, 0.50, \ldots, 5.00$ seconds. The conditioning frame at $t = 0$ is excluded. Each model retains its native rollout rate; only the metric queries are aligned to these common timestamps. The LingBot-family outputs are sampled from their 16-fps timeline at indices $4, 8, \ldots, 80$, Vid2World contributes its native 4-fps predictions at indices $1, 2, \ldots, 20$, and MultiWorld is sampled at the corresponding physical times on its native timeline. Solaris is queried on the corresponding timestamps of its native rollout timeline. This timestamp-based selection prevents higher-rate outputs or repeated export frames from receiving extra metric weight.

Route-specific spatial representations. *LingBot-family.* For LingBot Base, Direct Residual, SyncWorld, and the SyncWorld condition ablations and interventions, the prediction and corresponding target are already represented at 832×480; paired scoring therefore uses the identity transform.

MultiWorld. For MultiWorld, a native per-view prediction at 480×480 is resized to 832×480 with the audited bicubic output transform. The corresponding 832×480 target is first passed through the MultiWorld video-input transform to 480×480 and then through the same bicubic output transform. The comparison is made between the two resulting 832×480 frames. The separate environment-encoder preprocessing is not part of this visual prediction scoring path. Applying the same deterministic forward/reverse path to the target prevents the $832 \times 480 \leftrightarrow 480 \times 480$ interface conversion from being counted as prediction error.

Vid2World. For Vid2World, the native prediction is a 512 (W) $\times$ 320 (H) frame. The official vertical padding is removed by retaining rows 22:297, yielding a 512×275 valid-content frame, which is resized by the official-reverse bilinear transform to 832×480 with `align_corners=False`. The target follows the symmetric path: the canonical 832×480 frame is processed by the official $\text{Resize}(275, 512)$ and $\text{CenterCrop}(320, 512)$ operations, the same rows 22:297 are retained, and the resulting 512×275 content is passed through the same bilinear reverse transform. Metrics therefore compare transformed prediction and transformed target, rather than a native prediction against an untouched target; the padded canvas and its black border are not scored.

Solaris. For Solaris, the native prediction remains at 640×352. The canonical 832×480 target is center-cropped to the official target aspect ratio using the exact integer crop rule and resized with the official JAX linear transform to 640×352. For this source size, the crop retains 457 rows, $y = 11{:}468$. The native prediction is compared with this transformed target in the valid native representation. No fit-and-pad or black-border reconstruction is used for paired scoring, because Solaris's official preprocessing does not define an inverse source-space reconstruction analogous to Vid2World.

Metric computation and aggregation. PSNR is computed on RGB values from the pixel-wise floating-point MSE with data range 255 as $10\log_{10}(255^2/\mathrm{MSE})$, without Y-channel conversion, masking, or an additional border crop. SSIM uses the scikit-image implementation with RGB channel axis 2, data range 255, and window size 7; the verified runtime is scikit-image 0.25.2. LPIPS uses the AlexNet network with version 0.1 under lpips 0.1.4. RGB frames are mapped from $[0, 255]$ to $[-1, 1]$ and converted from HWC to NCHW, with no additional scorer-side resize after route-specific normalization and `normalize=False`.

For each prediction clip or observer sequence, metric values are first averaged arithmetically over the selected timestamps. Synchronized observers are then averaged within their group, and evaluation groups receive equal weight in the final aggregate. The protocol does not globally pool frames or weight a group merely because it contains more observer clips.

FVD. FVD is computed separately from paired fidelity metrics as a set-level video-distribution measure. The conditioning frame is excluded, and each method is evaluated on the same 80-frame future support in the common canonical 16-fps video representation. The same frozen preprocessing implementation and TorchScript I3D feature extractor are used for generated videos and for the real reference distribution. Fréchet distance is computed between the generated and real feature distributions; it does not require paired sample identity and is not evaluated on the route-specific paired-frame representations above.

Actor-condition fidelity. Presence Alignment, Position Error, and Action Fidelity evaluate whether the generated actors follow the supplied state. They are computed from the same normalized generated outputs using the frozen actor-detection, GT-anchored-correspondence, and action-recognition pipeline defined in Appendix B.

The protocol therefore standardizes held-out content, the physical prediction interval, paired timestamps, metric implementation, aggregation, and the FVD reference distribution, while preserving each model's native inference mechanism and valid deterministic spatial interface. Cross-view geometric consistency is evaluated with SymMVC; correspondence filtering, triangulation, pair aggregation, real-image references, and paired statistics are detailed in Appendix A.

E.5 Compute and Reproducibility

Experiments are conducted on 8 NVIDIA H200 GPUs. The reported evaluation uses the method-specific checkpoints and adaptation settings described in Appendix E.2; this section focuses on the compute and provenance controls used during evaluation.

Randomness and inference configurations are fixed per method before evaluation, while each model retains its native inference semantics as described in Appendix E.3. For the LingBot-family routes, the formal runner uses seed 20260818 and initializes a fresh seeded generator for each generation call. External baselines likewise use fixed route-specific seed and inference configurations. For each evaluation sample, we evaluate a single generation under a pre-specified seed and inference configuration; no candidate is selected based on evaluation metrics.

The formal evaluation cohort is fixed before inference and contains 1,024 samples from 14 held-out matches, complementing the evaluation-set construction described in Appendix E.1. Run metadata and validation records are retained to trace generated artifacts to their evaluation samples and effective inference configurations, including the method or route, checkpoint or configuration, seed where recorded, and output validation status. Dataset construction, interface alignment, and metric computation are specified separately in Appendix E.1, Appendix E.3, and Appendix E.4.

F Qualitative Demonstration Configuration

Figure 4 uses a validation clip from the archived three-view player-reveal demonstration. The low- and high-noise experts use training steps 2,700 and 2,850, respectively, with sampling seed 20260701. The three rows correspond to observers T2/P0003, T2/P0006, and T2/P0011. Each stream contains 81 frames at 16 fps in the legacy archived configuration at 832×464 resolution. The current formal/delivery configuration uses 832×480. We extract frames 0, 48, and 80 from the synchronized composite and preserve each observer's full image extent.

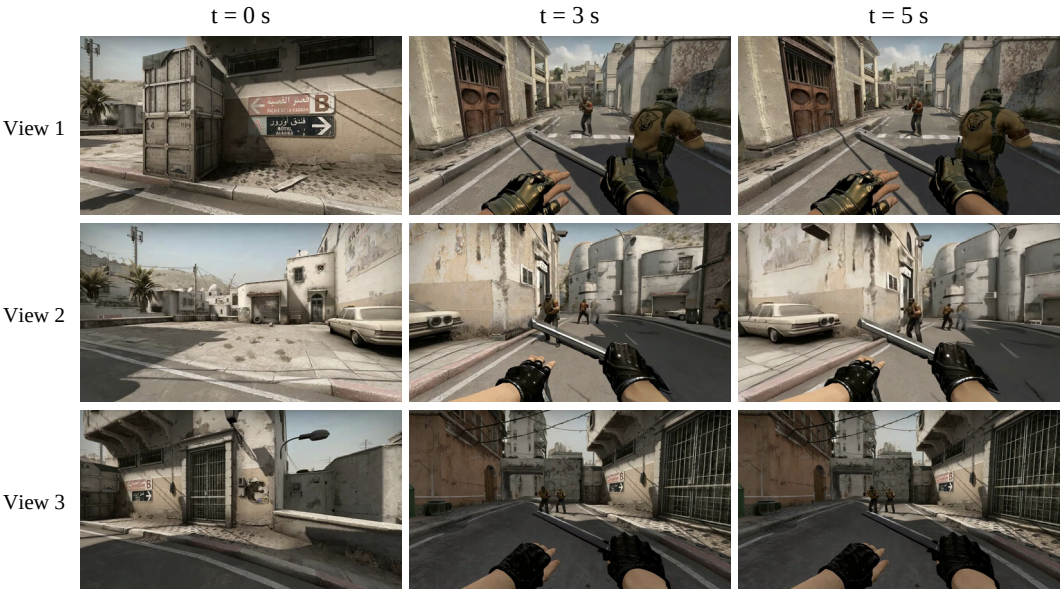

Figure 4: **Three views through time.** Frames at 0, 3, and 5 seconds from a synchronized demonstration. Each row follows one observer, showing the changing camera view and visible players. Columns align the three observations in time.

G Native Rollout and Reproducibility

Initialization and execution. The native rollout starts from one joint ten-player snapshot and initial velocities. The game server then advances under its native bot policy and exports the state trajectory for rendering. The record contains 321 state ticks at 32 Hz and 161 generated frames per observer at 16 fps. All ten observers complete two window decodes, giving 20 completed decodes and zero failures.

The recorded maximum initial-position error is 1.46×10^{-6} world units, and the maximum state step is 25.49 units. The audit records tick stride, displacement, per-view generation completeness, and future replay-action and replay-state read flags; both future-read flags are false. These fields connect the initialized state, native evolution, and synchronized observer outputs.

H QUALITATIVE MODEL COMPARISONS

Figure 2 in the main text shows player visibility (Case B); Figures 5 and 6 add shared map geometry (Case A) and large camera turns (Case C). Each compares LingBot Base, Direct Residual, Vid2World (Huang et al., 2026), MultiWorld (Wu et al., 2026), Solaris, and SyncWorld against ground truth (GT). Two views are shown at 3 and 5 seconds, with shared GT input references at 0 seconds. External models use their native input interfaces: Vid2World receives preceding frames and actions, MultiWorld paired images and mapped actions, and Solaris an initial image and mapped actions. Panels use a common 832×480 display canvas; Solaris's native valid content is padded only for qualitative display, while quantitative scoring uses the native valid region without padding as described in Appendix E.4.

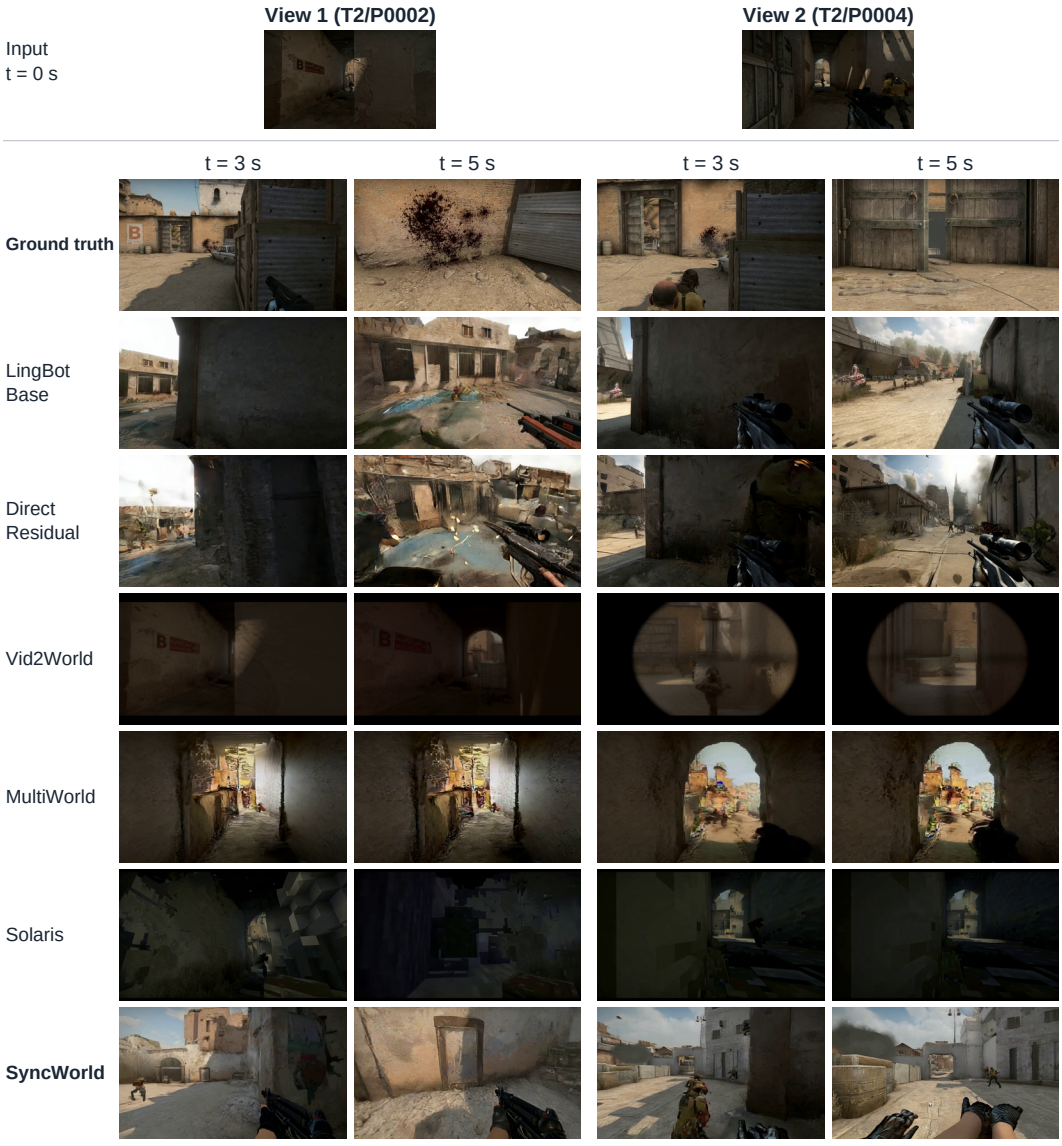

Figure 5: **Case A: shared map geometry.** Two views of the same scene, following the observers from the tunnel toward the courtyard. Compare the placement and shape of walls, doorways, and open areas against GT. Columns align display times across methods; the top images are shared initial references.

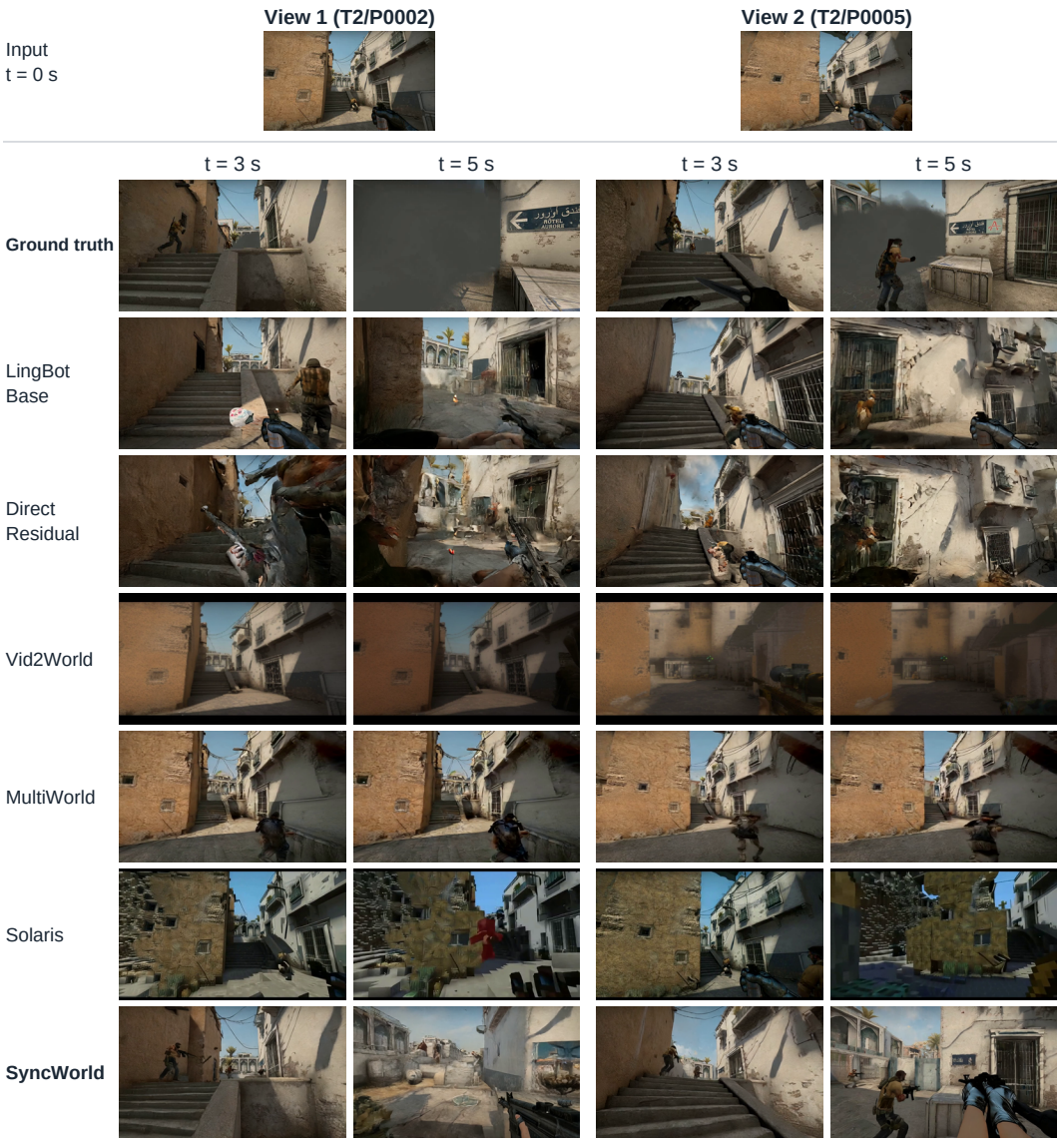

Figure 6: **Case C: large camera turns.** Initial views and later frames from a scene with substantial camera rotation. Compare the changing field of view and the placement of scene structures against GT. Rows denote methods, and the two column pairs follow the two observers.

I Qualitative Condition Ablations

Figures 7–9 provide qualitative examples of the condition variants in Section 6.1. Each comparison uses synchronized views at the same displayed time, with the same Full model weights and sampling seed across generated rows. Full frames are retained; cyan outlines, where present, identify a fixed GT-defined screen region enlarged in the rightmost column. These selected examples complement the quantitative evaluation by showing changes in scene content and player arrangement.

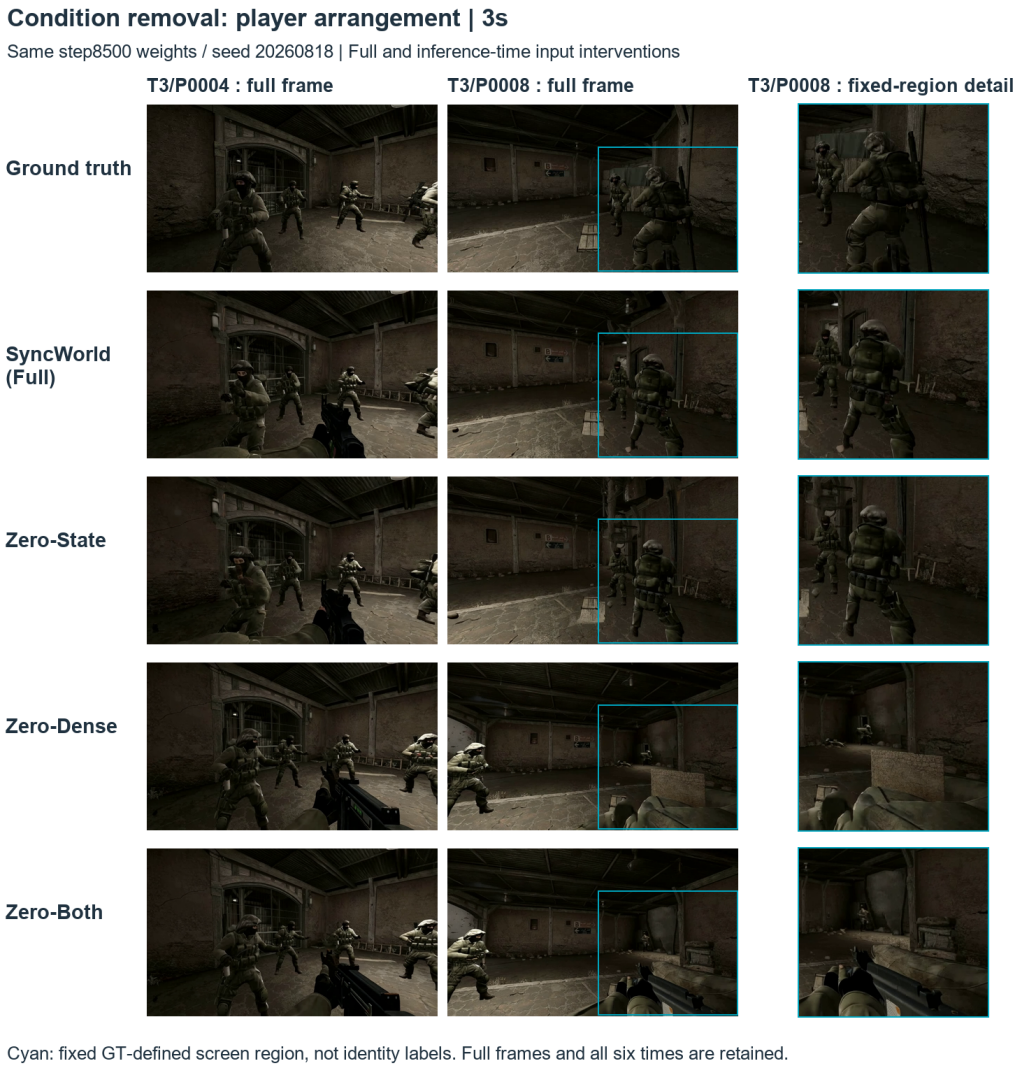

Figure 7: **Condition ablation: player arrangement.** Two synchronized views at 3 seconds (S06). In View 2, Full and Zero-State retain the near/far player arrangement more closely than Zero-Dense and Zero-Both. The difference is less pronounced in View 1. The rightmost column enlarges the same screen region across all rows.



Figure 8: **Condition mismatch across three synchronized views.** Frames at 3 seconds (S07), comparing Full, Time-Shift (+5), and Cross-Group Shuffle against GT. Full retains the stair-oriented composition of the first two reference views more closely, while the mismatched conditions change the visible scene structures and player arrangement. The third view provides additional context.

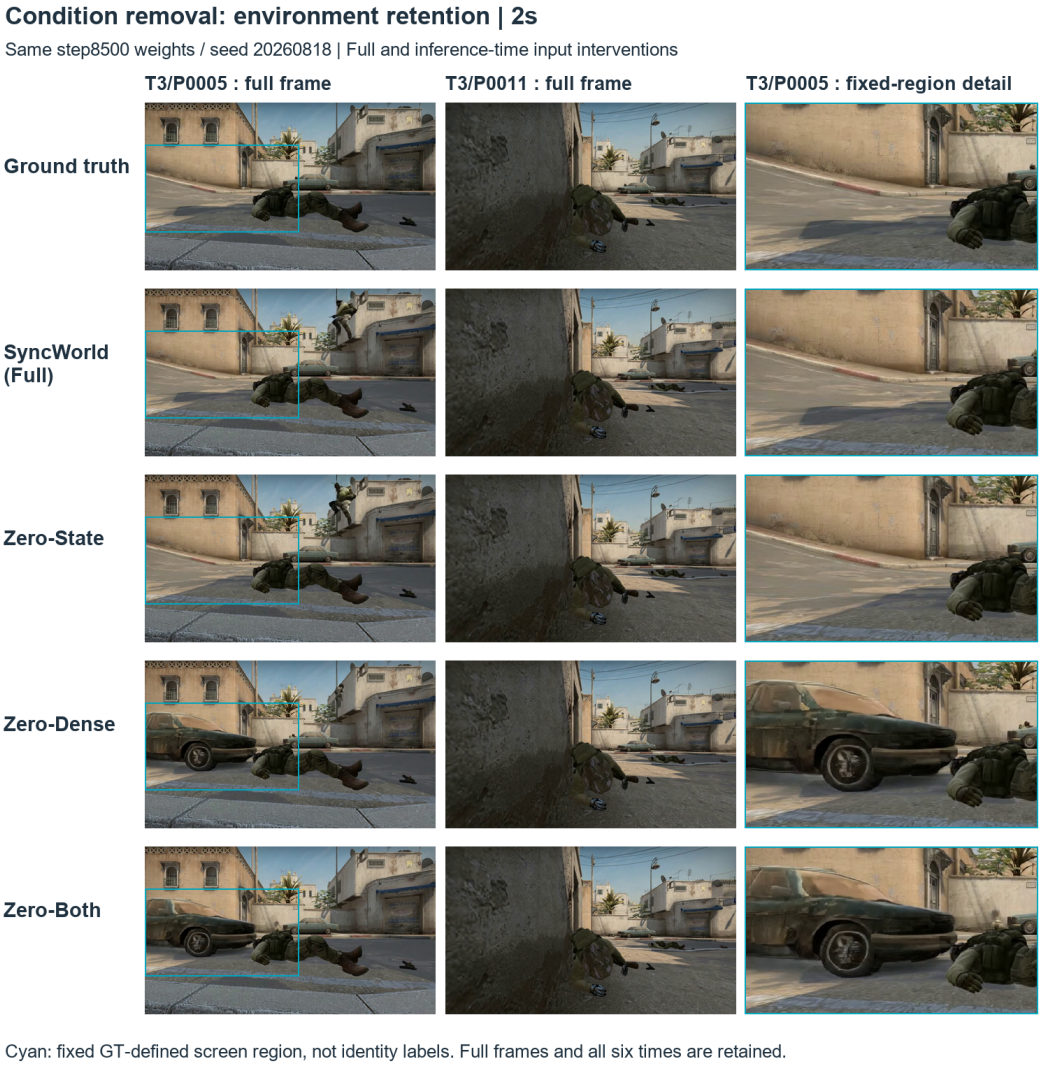

Figure 9: **Condition ablation: environment retention.** Two synchronized views at 2 seconds (S04). Zero-Dense and Zero-Both introduce a large foreground car in View 1 that is absent from GT, Full, and Zero-State. The difference is weaker in View 2. Full and Zero-State retain this local environment content. The rightmost column enlarges the fixed screen region marked in View 1.