# INERTIAL VIDEO WORLD MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

The physical world has inertia: it evolves from what already exists rather than being recreated at every instant. However, many existing video world models lack an explicit mechanism for inertial state evolution, leaving the continuity of motion and structure implicit in video prediction. During autoregressive rollouts, repeated reconstruction can accumulate errors and destabilize the predicted dynamics. To address this limitation, we introduce INERTIALWORLD, an Inertial Video World Model that makes inertia a structural prior for autoregressive prediction: existing scene evidence is carried across time rather than reconstructed at every step. Central to this formulation is Inertial Memory, which turns memory from passive predictive context into an evolving video state with two complementary views: a dense Eulerian grid preserves scene-aligned evidence, while persistent Lagrangian carriers bind local content to learned canonical coordinates. At each step, the model transports the carrier content according to action-conditioned motion, predicts the complete next video state from the transported carriers and dense grid, and uses a learned gate to combine transported and newly predicted content into the memory for the next step. Extensive experiments across multiple domains, including robot manipulation, 3D game simulation, and open-world navigation, demonstrate that INERTIALWORLD enables more accurate and stable autoregressive prediction across interactive video domains. Codes link: https://anonymous.4open.science/r/Anonymous_InertialWorld-7165/.

## 1 INTRODUCTION

Learned world models provide predictive environments for planning and control by modeling action-conditioned dynamics (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2025). Within this broader family, video world models instantiate learned dynamics through action-conditioned video prediction, enabling interactive simulation in games, robotic manipulation, and embodied navigation (Micheli et al., 2023; Bruce et al., 2024; Yang et al., 2024; Bar et al., 2025). Autoregressive and diffusion-based formulations have substantially improved generation fidelity and controllability (Wu et al., 2024; Alonso et al., 2024; Huang et al., 2026; Zhu et al., 2026), while post-training further refines predicted rollouts (Wu et al., 2025a; 2026). Despite these advances, physical evolution is inertial: the world evolves from what already exists rather than being recreated at every instant. Most video world models, however, still formulate temporal evolution as repeated conditional generation, predicting persistent scene content again at every step rather than explicitly carrying it forward. Since each generated state conditions the next transition, small reconstruction errors can repeatedly re-enter the rollout and destabilize predicted motion and structure (Wang et al., 2017; Babaeizadeh et al., 2021).

A prominent line of research mitigates this limitation by augmenting video prediction with memory. Recurrent video predictors retain temporal features across prediction steps (Wang et al., 2017; 2018), while transformer world models expose longer histories to the transition function (Chen et al., 2022; Wu et al., 2024). This additional context is particularly valuable when occlusion or accumulated generation noise makes the current frame ambiguous. Earlier features can retain identity and layout cues that are no longer directly visible but remain relevant to the next transition. Spatial and persistent memories preserve scene-aligned evidence across recurrent prediction steps (Wu et al., 2025b; Zhou et al., 2025b). Object-centric representations provide a complementary form of persistence by binding content to structured entities (Locatello et al., 2020; Kipf et al., 2022), and their video extensions model the dynamics of these entities through time (Wu et al., 2023). Collectively, these mechanisms expose historical evidence to subsequent state transitions and improve temporal consistency, but



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.

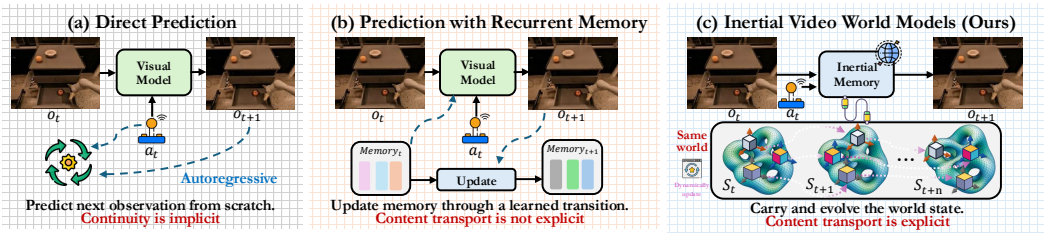



Figure 1: **From memory to inertia.** All three paradigms predict $\mathbf{o}_{t+1}$ from $\mathbf{o}_t$ and action $\mathbf{a}_t$. (a) Direct prediction reconstructs each next observation, leaving persistence implicit. (b) Recurrent memory retains the past as predictive context, but a generic transition need not use predicted motion to advance stored content. (c) INERTIALWORLD actively evolves its memory: action-conditioned coordinate motion carries bound content before gated prediction updates the state.

retention alone does not specify how the retained content should evolve. When memory serves as context for a generic transition, the transition remains free to synthesize the next state without explicitly advancing the stored content. Transformation-based predictors demonstrate the value of warping observations or features (Jaderberg et al., 2015; Finn et al., 2016), yet warping does not by itself make transport the recurrent evolution rule of persistent memory across model-predicted steps. Thus, memory determines which past evidence remains available, whereas inertia additionally specifies what persists and where it moves. This distinction motivates the central question of this work: ***How can a video world model evolve the world it already remembers, rather than repeatedly reconstructing its persistent content?***

To address this question, we introduce INERTIALWORLD, an Inertial Video World Model that makes inertia a structural prior for autoregressive prediction: existing scene evidence is carried across time rather than reconstructed at every step. As illustrated in Figure 1, its central component, *Inertial Memory*, turns retained predictive context into an evolving video state with two complementary views: a dense Eulerian grid preserves scene-aligned evidence, while persistent Lagrangian carriers bind local content to learned canonical coordinates. At each step, the model transports the carrier content according to action-conditioned motion, predicts the complete next video state from the transported carriers and dense grid, and uses a learned gate to combine transported and newly predicted content into the memory for the next step. This carry-before-write transition establishes persistence as the default path of state evolution while preserving the capacity to model changes that transport alone cannot explain, without requiring object annotations or calibrated physical parameters. We evaluate INERTIALWORLD on RT-1 robot manipulation (Brohan et al., 2022), Counter-Strike 3D game simulation (Huang et al., 2026), and RECON open-world navigation (Shah et al., 2021). Across all three interactive domains, INERTIALWORLD produces more accurate and stable autoregressive predictions than competitive video world model baselines.

Our contributions are threefold:

- We formulate *inertial video world modeling*, a state-evolution view of video world models in which existing scene evidence is inherited by default rather than repeatedly reconstructed.
- We introduce *Inertial Memory*, a hybrid Eulerian–Lagrangian video state that preserves dense scene evidence while transporting persistent regional content through action-conditioned coordinate motion before a learned gate updates the memory.
- We conduct extensive autoregressive prediction evaluations across robot manipulation, 3D game simulation, and open-world navigation, demonstrating consistently more accurate and stable rollouts than competitive video world model baselines.

## 2 RELATED WORK

**Video World Models.** Action-conditioned video prediction learns how future observations evolve under control (Oh et al., 2015; Finn et al., 2016). Early approaches use stochastic, recurrent, and physics-inspired dynamics to model uncertainty and temporal dependence (Denton & Fergus, 2018;

Wang et al., 2017; Le Guen & Thome, 2020; Babaeizadeh et al., 2021). Learned discrete representations (van den Oord et al., 2017; Esser et al., 2021) enable scalable autoregressive and masked video prediction (Yan et al., 2021; Yu et al., 2023; 2024; Wang et al., 2024). These formulations support interactive simulation across games and robotics (Bruce et al., 2024; Valevski et al., 2025; Wu et al., 2024; Yang et al., 2024; Zhou et al., 2025a; 2024), while diffusion-based models provide an alternative generative backbone (Alonso et al., 2024; Huang et al., 2026). Reinforcement learning and alignment objectives further refine predicted rollouts (Wu et al., 2025a; 2026). In parallel, latent world models learn compact predictive states for planning and imagined policy learning (Hafner et al., 2019; 2021; 2025), sometimes prioritizing task-relevant dynamics over frame reconstruction (Schrittwieser et al., 2020; Hansen et al., 2024). Rather than changing the video generator or its training objective, our work changes how the recurrent video state evolves by introducing transport-structured persistence that is compatible with different generative backbones.

**Memory and Structured Video State.** Existing memory mechanisms primarily determine *what* a video world model retains. Recurrent states and token histories preserve temporal context (Wang et al., 2017; 2018); spatial and persistent memories preserve scene-aligned content (Wu et al., 2025b; Zhou et al., 2025b); and object-centric representations bind content to persistent entities (Burgess et al., 2019; Greff et al., 2019; Locatello et al., 2020), with temporal extensions supporting tracking and prediction (Kipf et al., 2022; Wu et al., 2023; Kipf et al., 2020). Relational models expose interactions between state elements (Battaglia et al., 2016; Kipf et al., 2018; Battaglia et al., 2018), while Hamiltonian and graph-based simulators impose additional structure on state transitions (Greydanus et al., 2019; Sanchez-Gonzalez et al., 2020). These approaches enrich what persists, but they do not by themselves specify *how* retained content should evolve. A generic recurrent update can encode motion without requiring that motion to relocate content already in memory. Transformation-based predictors demonstrate the value of warping observations or features (Finn et al., 2016; Jaderberg et al., 2015), but warping alone does not make transport the recurrent evolution rule of a persistent state. Inertial Memory instead treats persistent canonical evidence as the transported quantity: it maintains a dense scene field, binds selected local content to learned coordinate frames, and uses their predicted evolution to transport the bound content before the memory is revised. Its defining property is therefore the explicit coupling of coordinate evolution and content transport within a hybrid dense–regional video state, rather than recurrence, slots, or warping in isolation.

# 3 Method

## 3.1 Problem Definition

Figure 2 summarizes the complete prediction pipeline. At time $t$, a video world model receives the current frame $\mathbf{o}_t$ and action $\mathbf{a}_t$ and predicts $\widehat{\mathbf{o}}_{t+1}$. A multi-step rollout observes $\mathbf{o}_t$ only once; all subsequent frames are generated from the model's own state and the action sequence. We use a frozen codec that represents the observed scene with a rollout-level Context code $\mathbf{C}$ and each predicted frame with a Dynamic code $\mathbf{D}$. We encode $\mathbf{C} = \mathrm{Enc}_C(\mathbf{o}_t)$ and initialize the recurrent video state as $\mathcal{S}_t = \mathrm{Init}(\mathbf{C})$. Each causal prediction step is then

$$(\widehat{\mathbf{D}}_{\tau+1}, \mathcal{S}_{\tau+1}) = \mathcal{F}_\theta(\mathcal{S}_\tau, \mathbf{a}_\tau; \mathbf{C}), \qquad \widehat{\mathbf{o}}_{\tau+1} = \mathrm{Dec}(\mathbf{C}, \widehat{\mathbf{D}}_{\tau+1}). \tag{1}$$

The Context code remains fixed and anchors the observed scene throughout the rollout. The predicted Dynamic code is a complete code for the next frame, not a residual correction to a ground-truth frame. At the first transition, there is no previous Dynamic code; all observed information enters through $\mathbf{C}$ and the initialized state. Future frames define training targets $\mathbf{D}^\star_{\tau+1}$, but neither those targets nor their encoded features are fed back into the recurrence. This causal interface defines the prediction problem but does not specify how the state should evolve. A generic recurrent state can retain past information while the transition still rewrites its content at every step. We instead make persistence structural: predicted motion first acts on stored content, and prediction updates the state only after this content has been carried forward.

## 3.2 Inertial Memory

Inertial Memory realizes this carry-before-write principle by coupling content with learned coordinates. Its state is $\mathcal{S}_\tau = (\mathbf{G}_\tau, \{\mathbf{r}_{\tau,k}\}_{k=1}^K, \mathbf{B}_\tau)$, where $\mathbf{G}_\tau$ is a dense Eulerian grid and

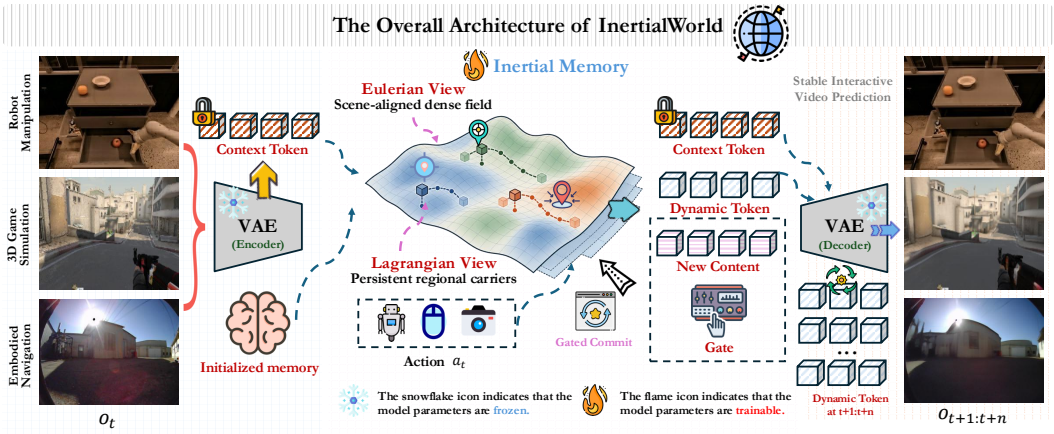



Figure 2: **Overall architecture of INERTIALWORLD.** A frozen encoder extracts the rollout-level Context code and initializes Inertial Memory. Its dense Eulerian grid preserves scene-aligned evidence, while action-conditioned Lagrangian carriers transport persistent regional content. A learned gate combines carried and newly predicted content, and a frozen decoder reconstructs future frames from the fixed Context code and predicted Dynamic codes. Snowflake and flame icons denote frozen and trainable components, respectively.

$\mathbf{r}_{\tau,k} = (\mathbf{A}_{\tau,k}, \boldsymbol{\alpha}_{\tau,k}, \boldsymbol{\xi}_{\tau,k})$ is a persistent Lagrangian carrier. The grid remains aligned with scene coordinates and preserves evidence across the full spatial domain. Each carrier instead moves with a local region: $\mathbf{A}_{\tau,k}$ stores canonical content, $\boldsymbol{\alpha}_{\tau,k}$ describes its support, and $\boldsymbol{\xi}_{\tau,k}$ specifies where that content is placed. The soft binding map $\mathbf{B}_\tau$ assigns evidence between the carriers and a dense fallback. The carriers are learned regions rather than supervised semantic objects, so this representation does not require object annotations or a complete object decomposition.

**Carrying persistent content.** Let $\boldsymbol{\xi}_\tau$ collect all carrier coordinates. The first part of a transition predicts their action-conditioned motion and applies it to the stored state:

$$\boldsymbol{\xi}_{\tau+1} = U_\theta(\mathcal{S}_\tau, \mathbf{a}_\tau), \qquad \widetilde{\mathcal{S}}_{\tau+1} = \mathrm{Carry}(\mathcal{S}_\tau; \boldsymbol{\xi}_\tau \to \boldsymbol{\xi}_{\tau+1}). \tag{2}$$

The coordinate update $U_\theta$ predicts bounded motion from the current memory and action. Carry re-renders each carrier's canonical content and support at its new coordinates, so local evidence moves with the carrier rather than being synthesized again at the destination. For the dense grid, the binding map combines carrier-induced motion with a learned fallback flow; normalized forward splatting propagates the evidence while accounting for ownership and visibility. The resulting $\widetilde{\mathcal{S}}_{\tau+1}$ is therefore the state inherited from the past. When the predicted motion is zero, this operation reduces to an identity carry. The coordinates are learned representational variables, not calibrated physical parameters.

**Predicting and updating.** Transport alone cannot explain newly visible content, deformation, or other changes not represented by the current memory. Let $\widetilde{\mathcal{M}}_{\tau+1}$ denote the writable content in the carried state. A predictor reads that state and jointly produces the complete next Dynamic code, a candidate memory update, and a learned write gate. Suppressing time subscripts within this transition and using $+$ for time $\tau+1$, this is

$$(\widehat{\mathbf{D}}^+, \mathcal{M}^{\mathrm{new}}, \mathbf{g}) = P_\theta(\widetilde{\mathcal{S}}, \mathbf{a}; \mathbf{C}), \qquad \mathcal{M}^+ = (1-\mathbf{g}) \odot \widetilde{\mathcal{M}} + \mathbf{g} \odot \mathcal{M}^{\mathrm{new}}. \tag{3}$$

The first output is decoded into the complete next frame, allowing the model to generate changes that cannot be obtained by transport alone. The second output proposes new writable memory content, while the gate determines which carried values should persist and which should be replaced. Together with the transported carrier coordinates, the updated content forms $\mathcal{S}_{\tau+1}$. Thus, persistence is the default path, but the memory can still adapt when the newly predicted state provides evidence for change. Because the proposal is generated from the model prediction rather than the target Dynamic code, repeated application of Eqs. equation 2–equation 3 remains strictly predicted-only. This ordered

transition—carry, predict, then update—is our operational definition of representational inertia; it does not impose a hard-coded Newtonian law.

### 3.3 Learning and Autoregressive Rollout

We train on the same predicted states used at inference and backpropagate through the complete rollout. Discrete internal decisions remain hard on the forward path and use a straight-through estimator for optimization. The recurrence is never refreshed with a target state or detached between autoregressive steps. We progressively increase the prediction horizon from 8 to 16 and then 32 steps so that the model learns to preserve and update its own state under repeated application.

The training objective combines prediction, memory-consistency, and regularization terms:

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda_{\text{mem}}\mathcal{L}_{\text{mem}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}. \tag{4}$$

Here, $\mathcal{L}_{\text{pred}}$ supervises the complete Dynamic token IDs, their finite scalar quantization digits (Mentzer et al., 2024), and the decoded frame's appearance and temporal change. $\mathcal{L}_{\text{mem}}$ provides consistency signals for transporting, binding, and updating memory from adjacent frozen-codec representations without object or contact annotations. $\mathcal{L}_{\text{reg}}$ regularizes carrier diversity and dense motion. Ground truth is used only to compute these losses; it never replaces $\mathcal{S}_\tau$ in the rollout. Training therefore matches the causal, predicted-only recurrence used at inference.

## 4 Experiments

Our experiments address four questions that follow directly from the proposed inertial principle: (i) does the complete video world model improve the accuracy and stability of autoregressive prediction across interaction domains, (ii) are its gains retained when evaluation focuses on action-induced motion, (iii) do its generated rollouts preserve behavioral differences between policies, and (iv) does performance degrade when Inertial Memory is removed? We answer these questions in the same order below.

### 4.1 Stable Autoregressive Video Prediction

We first evaluate autoregressive video prediction on RT-1 (Brohan et al., 2022). Every method is evaluated on the same 1,024 episodes and predicts 32 future frames using its native observation interface. All outputs are compared with the same $256 \times 320$ ground-truth frames. We evaluate the released checkpoints of RLVR-World (Wu et al., 2025a), Vid2World (Huang et al., 2026), WorldGym (Quevedo et al., 2025), and DriftWorld (Lu et al., 2026) without retraining. The frozen protocol is disjoint from our training and development episodes; because released methods use different native RT-1 partitions, it is not necessarily held out for every baseline. Under the released RLVR-World partition, 1,015 of the 1,024 evaluation episodes belong to its training subset. We therefore treat this as a descriptive released-checkpoint comparison, rather than a controlled estimate of held-out generalization.

For Counter-Strike 3D game simulation, we follow the CSGO benchmark and data partition used by Vid2World (Huang et al., 2026). From its 500 held-out episodes, we reserve the first 128 for validation and evaluate a fixed 70-episode subset of the remaining split. Each selected episode contributes one fixed anchor with at least 32 preceding frames, and every method predicts the same 32 adjacent targets at $256 \times 320$ resolution. We compare against the released checkpoints of Vid2World and DIAMOND (Alonso et al., 2024); each method retains its native conditioning interface, while conditioning frames are never scored. Appendix C specifies the frozen selection rule and the complete frame–action alignment.

We additionally evaluate video prediction for embodied navigation on RECON (Shah et al., 2021). We freeze 128 distinct test episodes from the official 2,367-episode held-out split after excluding the 17 trajectories used for validation. The common anchor in each episode provides the last observed frame $o_t$, and every method predicts the same 32 adjacent targets $o_{t+1:t+32}$ at $240 \times 320$ resolution; baselines retain their native four-frame context when required. We compare the released checkpoints of Vid2World, NWM (Bar et al., 2025), and RAE-NWM (Zhang et al., 2026). For all three dataset blocks in Table 1, we first average each frame-level metric over the 32 predicted frames of an episode.

Table 1: Autoregressive video prediction on RT-1, CSGO, and RECON using released baseline checkpoints. Every method predicts 32 future frames. Best results within each dataset are in **bold**, and second-best results are underlined. Pixel errors are measured on RGB values normalized to $[0, 1]$; the final column reports MSE for RT-1 and MAE for CSGO and RECON, as indicated in each block header.

| *Robot Manipulation: RT-1* | | | | |
|---|---|---|---|---|
| Model | PSNR ↑ | SSIM ↑ | LPIPS ↓ | MSE ↓ |
| RLVR-World | $21.95 \pm 2.36$ | $0.780 \pm 0.062$ | $0.168 \pm 0.050$ | $0.0086 \pm 0.0053$ |
| Vid2World | $21.32 \pm 2.35$ | $0.724 \pm 0.072$ | $0.244 \pm 0.050$ | $0.0101 \pm 0.0060$ |
| WorldGym | $21.72 \pm 2.45$ | $0.782 \pm 0.063$ | $0.205 \pm 0.050$ | $0.0094 \pm 0.0060$ |
| DriftWorld | $20.83 \pm 2.44$ | $0.731 \pm 0.072$ | $0.266 \pm 0.061$ | $0.0116 \pm 0.0064$ |
| **INERTIALWORLD (Ours)** | $\mathbf{24.37 \pm 2.60}$ | $\mathbf{0.839 \pm 0.054}$ | $\mathbf{0.147 \pm 0.049}$ | $\mathbf{0.0051 \pm 0.0036}$ |
| *3D Game Simulation: CSGO* | | | | |
| Model | PSNR ↑ | SSIM ↑ | LPIPS ↓ | MAE ↓ |
| DIAMOND | $17.04 \pm 3.20$ | $0.513 \pm 0.108$ | $0.576 \pm 0.080$ | $0.1187 \pm 0.0525$ |
| Vid2World | $16.69 \pm 3.14$ | $0.487 \pm 0.108$ | $0.518 \pm 0.084$ | $0.1234 \pm 0.0509$ |
| **INERTIALWORLD (Ours)** | $\mathbf{18.36 \pm 3.02}$ | $\mathbf{0.565 \pm 0.097}$ | $\mathbf{0.516 \pm 0.086}$ | $\mathbf{0.0934 \pm 0.0358}$ |
| *Embodied Navigation: RECON* | | | | |
| Model | PSNR ↑ | SSIM ↑ | LPIPS ↓ | MAE ↓ |
| NWM | $13.03 \pm 1.46$ | $0.410 \pm 0.082$ | $0.507 \pm 0.051$ | $0.1623 \pm 0.0359$ |
| RAE-NWM | $14.66 \pm 1.56$ | $0.424 \pm 0.080$ | $0.458 \pm 0.046$ | $0.1285 \pm 0.0280$ |
| Vid2World | $15.59 \pm 1.74$ | $0.488 \pm 0.080$ | $0.416 \pm 0.054$ | $0.1099 \pm 0.0262$ |
| **INERTIALWORLD (Ours)** | $\mathbf{16.70 \pm 1.67}$ | $\mathbf{0.546 \pm 0.076}$ | $\mathbf{0.401 \pm 0.050}$ | $\mathbf{0.0895 \pm 0.0221}$ |

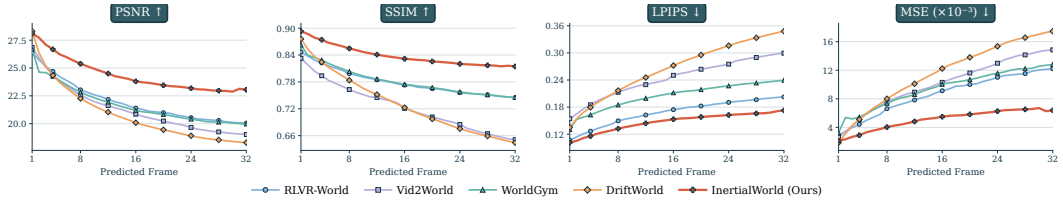



Figure 3: **Prediction quality throughout the 32-step RT-1 rollout.** Each curve reports the episode-level mean at every predicted frame under the same 1,024-episode protocol as Table 1. Higher is better for PSNR and SSIM, whereas lower is better for LPIPS and MSE.

We then report the mean and sample standard deviation across 1,024 RT-1 episodes, 70 CSGO test episodes, and 128 RECON test episodes, respectively; conditioning frames are never scored.

Figure 3 resolves the aggregate RT-1 results over the complete autoregressive rollout. INERTIALWORLD achieves the highest PSNR and SSIM and the lowest LPIPS and MSE at every predicted frame. Its advantage is retained as the prediction horizon increases, showing that the main-table gain reflects consistently more stable video evolution rather than an improvement confined to the first few transitions.

Figure 4 provides a complementary visual comparison under the same 32-step protocols. On RT-1, INERTIALWORLD preserves the articulated drawer geometry and the spatial relation between the drawer and robot through the final prediction. On CSGO, it retains the corridor layout and action-induced scene evolution without the pronounced structural degradation visible in competing rollouts. On RECON, it better maintains the curved building boundary, ground plane, and surrounding vegetation as the camera advances. Across all three domains, the benefit is most apparent at later predicted frames and in the highlighted regions, where independently reconstructing persistent content tends to blur, duplicate, or displace scene structure.

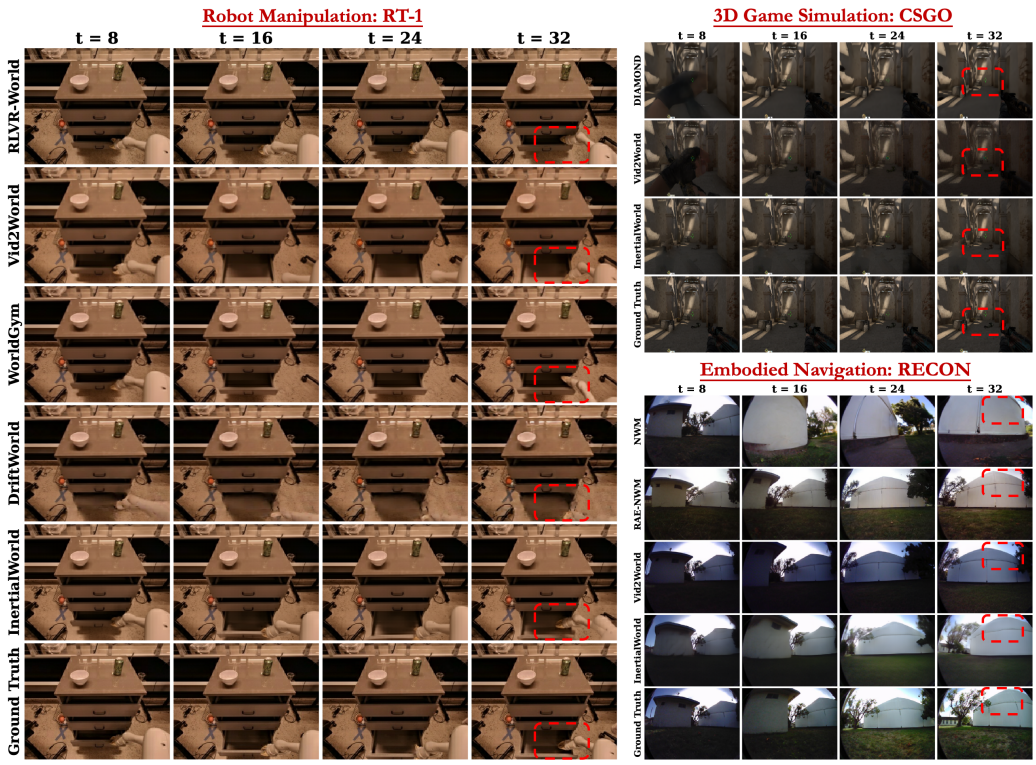



Figure 4: **Qualitative comparison across the three evaluation domains.** We show predicted frames at steps 8, 16, 24, and 32 for RT-1 robot manipulation, CSGO game simulation, and RECON embodied navigation. Each method uses its native conditioning interface, while all rows within a domain share the same episode, anchor, and action sequence. Red boxes mark structurally informative regions at the final step.

Table 1 summarizes the released-checkpoint comparison. On RT-1, INERTIALWORLD obtains the strongest values across pixel fidelity, structural similarity, and perceptual distance. Against the strongest baseline for each metric, PSNR increases from 21.95 to 24.37 dB, SSIM from 0.782 to 0.839, LPIPS decreases from 0.168 to 0.147, and MSE from 0.0086 to 0.0051. On the 70-episode CSGO evaluation, INERTIALWORLD ranks first on all four metrics. Against DIAMOND, PSNR increases from 17.04 to 18.36 dB, SSIM from 0.513 to 0.565, and MAE decreases from 0.1187 to 0.0934; against Vid2World, LPIPS decreases from 0.518 to 0.516. On the held-out RECON protocol, INERTIALWORLD again ranks first on every metric. Against Vid2World, PSNR increases from 15.59 to 16.70 dB, SSIM from 0.488 to 0.546, LPIPS decreases from 0.416 to 0.401, and MAE from 0.1099 to 0.0895. Overall, INERTIALWORLD ranks first in all 12 reported dataset–metric combinations, supporting the cross-domain accuracy and stability of the complete video world model. Because the evaluated checkpoints retain their native input interfaces and training recipes, the table establishes their relative video prediction quality under the stated protocols but does not by itself attribute every difference to a particular architectural choice.

### 4.2 MOTION FIDELITY UNDER ACTION

Aggregate appearance metrics can be dominated by the largely static regions of a manipulation video. We therefore complement the RT-1 results in Table 1 with three fixed diagnostics targeted at action-induced change. Because RT-1 does not provide optical-flow annotations, we apply the same frozen RAFT (Teed & Deng, 2020) estimator to each predicted and reference rollout. *Flow EPE* is the mean endpoint discrepancy between their estimated flows over all pixels and all 32 transitions; it is not an error against measured optical-flow ground truth. *Flow Cosine* measures directional agreement on reference-dynamic pixels, defined by a reference-flow magnitude of at least one pixel. *Dynamic-region LPIPS* applies spatial LPIPS (Zhang et al., 2018) only within the corresponding

Table 2: **GT-grounded motion fidelity on RT-1.** Values are episode-level mean ± standard deviation over the same 1,024 episodes and 32 predicted frames as the RT-1 block of Table 1. Best results are in **bold**; second-best results are underlined. Flow EPE is measured in pixels per frame.

| Model | Flow EPE ↓ | Flow Cos. ↑ | Dyn. LPIPS ↓ |
|---|---|---|---|
| RLVR-World | 1.110±0.869 | 0.466±0.183 | 0.403±0.074 |
| Vid2World | 1.334±1.002 | 0.538±0.191 | 0.436±0.068 |
| WorldGym | 1.549±1.214 | 0.381±0.177 | 0.444±0.066 |
| DriftWorld | 1.366±1.024 | 0.448±0.172 | 0.501±0.053 |
| **INERTIALWORLD** | **1.032±0.926** | **0.584±0.186** | **0.332±0.083** |

reference-motion mask, reducing the influence of static background. We first aggregate each quantity within an episode and then report statistics across episodes.

Table 2 shows that INERTIALWORLD ranks first on all three diagnostics. Against the strongest baseline for each measure, flow EPE decreases from 1.110 to 1.032, flow-direction cosine increases from 0.538 to 0.584, and dynamic-region LPIPS decreases from 0.403 to 0.332. The gains therefore persist when evaluation is concentrated on where and how the scene changes, rather than on the static pixels that dominate the frame. This behavior is consistent with the intended role of Inertial Memory: preserving stored frame content while action-conditioned motion transports it through the predicted video state.

These measures are protocol-specific diagnostics instantiated from frozen RAFT and LPIPS models, rather than an external leaderboard. All methods use the same estimators, thresholds, target-derived masks, episode set, and aggregation order; no metric or horizon is selected separately for an individual method.

### 4.3 REAL2SIM POLICY EVALUATION

We next test whether INERTIALWORLD can expose behavioral differences between frozen control policies, rather than merely reconstructing held-out videos. Starting from an observation in the frozen RT-1 test protocol, the policy receives the frame generated by the video world model, and its predicted action conditions the next model transition. Each closed-loop trajectory contains 40 frames including the initial observation, with no future image or action supplied by the dataset. Task success is determined by blind manual inspection of the resulting trajectory.

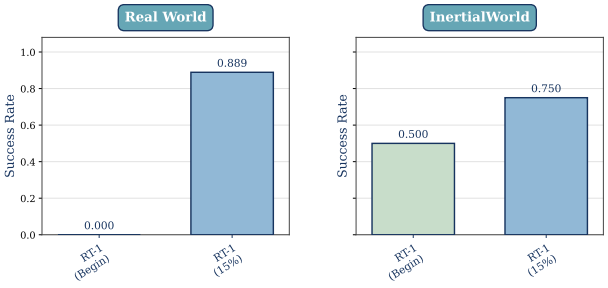



Figure 5: **Preliminary Real2Sim evaluation on close-drawer tasks.** The left panel shows the published real-world references from SIMPLER (Li et al., 2025); the right panel reports blind success labels for 40-frame rollouts generated by INERTIALWORLD. Each displayed policy–task cell contains eight trajectories. The two panels use different initial states and evaluation lengths and are therefore compared only at the level of policy behavior, not as paired success estimates.

Figure 5 reports a preliminary analysis of the first 72 entries in the fixed blind-review order. We focus here on two early RT-1 checkpoints; the complete four-checkpoint analysis requires the full blind set. For the displayed close-drawer trajectories, RT-1 (Begin) succeeds on $4/8$ and RT-1 (15%) succeeds on $6/8$, yielding observed rates of 0.50 and 0.75. The ordering agrees with the published real-world reference, where the same checkpoints obtain 0.00 and 0.889, respectively; the RT-1 (Begin) real-world evaluation was stopped after two trials for safety. This result indicates that the generated environment distinguishes the less-trained policy from the stronger checkpoint under closed-loop interaction. Because the current labels cover only 72 of 144 generated trajectories and the real-world numbers come from an unpaired evaluation with a different interaction length, we interpret the figure as evidence of qualitative policy discrimination rather than agreement in absolute success rate.

### 4.4 Ablation Study

We assess the complete proposed design against a capacity-controlled *Plain AR* baseline that is deliberately slightly larger. Plain AR removes Inertial Memory and directly predicts the next Dynamic state from the fixed Context, the current action, and the rollout step. Its first transition uses a learned beginning-of-sequence state because no previous Dynamic code exists; subsequent transitions additionally receive the previous prediction. Both models use the same data, optimizer, horizon curriculum, training budget, and evaluation protocol. Plain AR is slightly larger than the full model, preventing additional capacity from explaining the result. We further probe the trained full model with two inference-time interventions. *Eulerian only* disables the Lagrangian carrier and transport paths while preserving the dense state, whereas *bias-only gates* remove state and action dependence from the learned update gates by zeroing their inputs while retaining their checkpoint biases.

Table 3: Effect of Inertial Memory on RT-1. All results use the same 1,024-episode protocol with 32 predicted frames and raw, non-EMA checkpoints at 30k training steps using one training seed. Each entry is the mean and sample standard deviation across episodes after averaging the 32 predicted frames within each episode. Plain AR is trained independently with the matched recipe. † denotes an inference-time intervention on the full checkpoint rather than a retrained model. Best predictive results are shown in **bold**.

| Model | PSNR ↑ | SSIM ↑ | LPIPS ↓ | MSE ↓ ($\times 10^{-3}$) |
|---|---|---|---|---|
| Plain AR (w/o Inertial Memory) | $22.321 \pm 2.500$ | $0.8127 \pm 0.0578$ | $0.1777 \pm 0.0557$ | $8.016 \pm 4.778$ |
| Eulerian-only† | $20.494 \pm 2.614$ | $0.7827 \pm 0.0662$ | $0.2120 \pm 0.0663$ | $11.982 \pm 6.844$ |
| Bias-only gates† | $21.283 \pm 2.692$ | $0.7962 \pm 0.0621$ | $0.1934 \pm 0.0603$ | $10.726 \pm 6.048$ |
| **InertialWorld (Full)** | $\mathbf{23.367 \pm 2.566}$ | $\mathbf{0.8264 \pm 0.0560}$ | $\mathbf{0.1620 \pm 0.0536}$ | $\mathbf{6.296 \pm 4.078}$ |

As shown in Table 3, removing Inertial Memory yields a consistent degradation in the controlled 30k comparison. Compared with Plain AR, the full model increases PSNR from 22.321 to 23.367 dB and SSIM from 0.8127 to 0.8264, while decreasing LPIPS from 0.1777 to 0.1620 and MSE ($\times 10^{-3}$) from 8.016 to 6.296. It achieves these results with fewer parameters than Plain AR, supporting the benefit of inertial state evolution rather than additional capacity.

The inference-time interventions expose which operations the learned transition relies on. Disabling the Lagrangian path changes PSNR from 23.367 to 20.494 dB and MSE ($\times 10^{-3}$) from 6.296 to 11.982, showing that the learned transition relies on persistent carrier transport in addition to the dense path. Replacing the learned gates with their bias-only responses changes PSNR from 23.367 to 21.283 dB and MSE ($\times 10^{-3}$) from 6.296 to 10.726, demonstrating the importance of adaptive memory updates. Together, these results support the two complementary views and gated commit used by Inertial Memory. Because the † variants intervene on a trained model rather than retraining a new architecture, we interpret them as mechanistic probes, not capacity-controlled training baselines.

## 5 Conclusion

We introduce InertialWorld, an Inertial Video World Model that treats prediction as the evolution of an existing video state rather than its repeated reconstruction. Inertial Memory realizes this principle through two complementary views: a dense Eulerian grid preserves scene-aligned evidence, while persistent Lagrangian carriers bind and transport regional content under action-conditioned motion. The model then predicts the complete next state and uses learned gates to incorporate changes that transport alone cannot explain. This carry-before-write transition makes persistence the default while retaining the flexibility required for newly visible content and non-rigid change. Across paired 32-step evaluations on RT-1, CSGO, and RECON, InertialWorld achieves the strongest aggregate prediction results and improves diagnostics focused on action-induced motion. Controlled and inference-time ablations further support the roles of structured transport and state-dependent memory updates. Together, these results identify inertial state evolution as a useful design principle for interactive video world models. Extending the current deterministic, learned-coordinate formulation to multimodal futures and explicit three-dimensional scene representations remains an important direction.

### AI USE STATEMENT

We use AI to check for grammatical errors in the text; the AI is not involved in ideas, experimental design, or the actual writing of the paper.

### ETHICS STATEMENT

This work complies with the ICLR Code of Ethics. Human evaluations were conducted with informed consent from volunteers. All datasets and pretrained models are publicly available and used under their respective licenses. We have carefully designed our experiments to minimize risks of bias or discrimination. We uphold the principles of transparency and integrity throughout the work.

### REPRODUCIBILITY STATEMENT

The method section specifies the causal interface, recurrent video state, memory transition and readout, and training objective. The implementation records model configurations, fixed data splits, checkpoint-selection rules, and the sampling and optimizer states required to resume training.

## 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.

Mohammad Babaeizadeh, Mohammad Taghi Saffar, Suraj Nair, Sergey Levine, Chelsea Finn, and Dumitru Erhan. Fitvid: Overfitting in pixel-level video prediction. *arXiv preprint arXiv:2106.13195*, 2021.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Peter W. Battaglia, Razvan Pascanu, Matthew Lai, Danilo Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems*, 2016.

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Anthony Brohan et al. RT-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, et al. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623. PMLR, 2024.

Christopher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. MONet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

Chang Chen, Yi-Fu Wu, Jaesik Yoon, and Sungjin Ahn. TransDreamer: Reinforcement learning with transformer world models. *arXiv preprint arXiv:2202.09481*, 2022.

Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *International Conference on Machine Learning*, 2018.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems*, 2016.

Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Chris Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning*, 2019.

Sam Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems*, 2019.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, 2019.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with discrete world models. In *International Conference on Learning Representations*, 2021.

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

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.

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.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, 2015.

Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, 2018.

Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations*, 2020.

Thomas Kipf, Gamaleldin F. Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. In *International Conference on Learning Representations*, 2022.

Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Oier Mees, Karl Pertsch, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 3705–3728. PMLR, 2025. URL https://proceedings.mlr.press/v270/li25c.html.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, 2020.

Susie Lu, Haonan Chen, Weirui Ye, and Yilun Du. DriftWorld: Fast world modeling through drifting. *arXiv preprint arXiv:2607.15065*, 2026.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *International Conference on Learning Representations*, 2024.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *International Conference on Learning Representations*, 2023.

Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L. Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in Atari games. In *Advances in Neural Information Processing Systems*, 2015.

Julian Quevedo, Ansh Kumar Sharma, Yixiang Sun, Varad Suryavanshi, Percy Liang, and Sherry Yang. WorldGym: World model as an environment for policy evaluation, 2025. URL https://arxiv.org/abs/2506.00613.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning*, 2020.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588:604–609, 2020.

Dhruv Shah, Benjamin Eysenbach, Nicholas Rhinehart, and Sergey Levine. Rapid exploration for open-world navigation with latent goal models. In *5th Annual Conference on Robot Learning*, 2021. URL https://openreview.net/forum?id=d_SWJhyKfVw.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, 2020.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=P8pqeEkn1H.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, 2017.

Junke Wang, Yi Jiang, Zehuan Yuan, Binyue Peng, Zuxuan Wu, and Yu-Gang Jiang. OmniTokenizer: A joint image-video tokenizer for visual generation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 28281–28295. Curran Associates, Inc., 2024. doi: 10.52202/079017-0887.

Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S. Yu. PredRNN: Recurrent neural networks for predictive learning using spatiotemporal LSTMs. In *Advances in Neural Information Processing Systems*, 2017.

Yunbo Wang, Zhifeng Gao, Mingsheng Long, Jianmin Wang, and Philip S. Yu. PredRNN++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In *International Conference on Machine Learning*, 2018.

Hao Wu, Yuqi Li, Yuan Gao, Fan Xu, Fan Zhang, Kun Wang, Penghao Zhao, Qiufeng Wang, Yizhou Zhao, Weiyan Wang, Yingli Tian, Xian Wu, and Xiaomeng Huang. RoboAlign-R1: Distilled multimodal reward alignment for robot video world models. *arXiv preprint arXiv:2605.03821*, 2026.

Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. iVideoGPT: Interactive VideoGPTs are scalable world models. In *Advances in Neural Information Processing Systems*, 2024.

Jialong Wu, Shaofeng Yin, Ningya Feng, and Mingsheng Long. RLVR-World: Training world models with reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 125312–125350. Curran Associates, Inc., 2025a. doi: 10.52202/085713-4177.

Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory. In *Advances in Neural Information Processing Systems*, 2025b.

Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. SlotFormer: Unsupervised visual dynamics simulation with object-centric models. In *International Conference on Learning Representations*, 2023.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. VideoGPT: Video generation using VQ-VAE and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations*, 2024.

Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, et al. MAGVIT: Masked generative video transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Lijun Yu, José Lezama, Nitesh B. Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, et al. Language model beats diffusion—tokenizer is key to visual generation. In *International Conference on Learning Representations*, 2024.

Mingkun Zhang, Wangtian Shen, Fan Zhang, Haijian Qin, Zihao Pei, and Ziyang Meng. RAE-NWM: Navigation world model in dense visual representation space. *arXiv preprint arXiv:2603.09241*, 2026.

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.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135. PMLR, 2025a.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. RoboDreamer: Learning compositional world models for robot imagination. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61885–61896. PMLR, 2024.

Siyuan Zhou, Yilun Du, Yuncong Yang, Lei Han, Peihao Chen, Dit-Yan Yeung, and Chuang Gan. Learning 3D persistent embodied world models. In *Advances in Neural Information Processing Systems*, 2025b.

Yixuan Zhu, Jiaqi Feng, Wenzhao Zheng, Yuan Gao, Xin Tao, Pengfei Wan, Jie Zhou, and Jiwen Lu. Astra: General interactive world model with autoregressive denoising. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=8UZpmrxoLG.

## APPENDIX CONTENTS

**A Complete Formulation of Inertial Memory** **15**

A.1 Frozen Video Representation . . . . . 15

A.2 Initialization and Dual-View Binding . . . . . 15

A.3 Action-Conditioned Carry . . . . . 15

A.4 Complete Prediction and Gated Commit . . . . . 16

**B Architecture and Training Details** **16**

B.1 Frozen Video Tokenizer . . . . . 16

B.2 World-Model Architecture and Objective . . . . . 17

**C Datasets and Frozen Evaluation Protocols** **17**

C.1 RT-1 . . . . . 18

C.2 CSGO . . . . . 18

C.3 RECON . . . . . 18

C.4 Canonical Targets and Leakage Checks . . . . . 19

**D Baseline Reproduction and Metric Definitions** **19**

D.1 Released-Checkpoint Rollout Contracts . . . . . 19

D.2 Frame Metrics and Statistical Aggregation . . . . . 20

D.3 Motion-Fidelity Diagnostics . . . . . 20

**E Ablation Protocol and Interpretation** **20**

# A Complete Formulation of Inertial Memory

This section expands the compact formulation in Section 3.2. We separate the state that implements the proposed inertial prior from auxiliary predictor context, and then describe a complete causal transition. Batch dimensions are omitted. Every recurrent variable is a function only of the initial observation, the action prefix, and previous model predictions.

## A.1 Frozen Video Representation

The frozen video tokenizer decomposes a video into a rollout-level Context code $\mathbf{C}$ and frame-level Dynamic codes $\mathbf{D}_\tau$. The Context code contains two $32\times40$ lanes and is computed once from $\mathbf{o}_t$. A Dynamic code contains four $8\times10$ lanes that represent base and detail components of motion and appearance. It is a complete discrete representation of a frame, rather than a residual relative to an observed image. During a rollout, $\mathbf{C}$ remains fixed and every future $\widehat{\mathbf{D}}_\tau$ is predicted by the model.

The recurrent video state is

$$\mathcal{S}_\tau = (\mathbf{C}, \mathcal{I}_\tau, \mathbf{H}_\tau, \mathbf{Q}_\tau), \qquad \mathcal{I}_\tau = (\mathbf{G}_\tau, \{\mathcal{R}_{\tau,k}\}_{k=1}^{K}, \mathbf{B}_\tau). \tag{5}$$

Here, $\mathcal{I}_\tau$ is Inertial Memory. Its Eulerian component $\mathbf{G}_\tau$ is a dense $16\times20$ grid aligned with scene coordinates. Its Lagrangian component contains $K = 16$ persistent regional carriers. Carrier $k$ maintains a descriptor, a canonical feature map and support, a translation–rotation–scale coordinate, persistent velocity, visibility and existence estimates, and a bounded local deformation. The soft binding map $\mathbf{B}_\tau$ assigns Eulerian evidence between the carriers and a dense background branch. The carriers are learned regions, not supervised semantic objects, and the background branch avoids imposing a complete object decomposition.

The implementation additionally keeps the four most recent *predicted* Dynamic states $\mathbf{H}_\tau$ and 64 generic interaction slots $\mathbf{Q}_\tau$. They provide predictor context inherited from the video backbone; the proposed persistent video state and its carry operation are defined by $\mathcal{I}_\tau$. This distinction is important: the method is not equivalent to retaining a longer image history as passive attention context.

## A.2 Initialization and Dual-View Binding

Learned projections of $\mathbf{C}$ initialize the dense grid and interaction slots. The first $K$ slots initialize the carrier descriptors. Carrier coordinates begin on a normalized $4\times4$ lattice with zero velocity; their canonical content is sampled from the initialized dense state. This constructs $\mathcal{S}_t = \mathrm{Init}(\mathbf{C})$ from the single observed frame, without an object detector or calibrated geometry.

At each recurrent step, a carrier is rendered into scene coordinates using its canonical content, support, coordinate, rigidity, and deformation. Its binding score at an Eulerian cell combines content compatibility, rendered support, visibility, and existence. A softmax over the $K$ carriers and the background branch gives $\mathbf{B}_\tau$. The dense view consequently retains globally available scene evidence, whereas the carrier view gives local content a persistent coordinate system in which it can move.

## A.3 Action-Conditioned Carry

The current action is combined with every carrier state, and a learned contact graph propagates action-dependent interactions between nearby carriers. The dynamics head predicts bounded changes to carrier velocity, coordinate, rigidity, existence, visibility, and deformation. Abstractly, the persistent coordinate update is

$$\mathbf{v}_{\tau+1,k} = \eta\mathbf{v}_{\tau,k} + \Delta\mathbf{v}_{\tau,k}, \qquad \boldsymbol{\xi}_{\tau+1,k} = \Pi(\boldsymbol{\xi}_{\tau,k} + \mathbf{v}_{\tau+1,k}), \tag{6}$$

where the learned increment is bounded component-wise and $\Pi$ enforces the valid coordinate range. Carrier motion is therefore accumulated through a persistent state rather than independently re-estimated from scratch.

Old and updated carrier coordinates induce a Lagrangian correspondence. The binding-weighted carrier flow $\mathbf{f}_\tau^L$ is combined with a dense Eulerian fallback flow $\mathbf{f}_\tau^E$ according to foreground mass $m_{\tau,i} = \sum_k B_{\tau,ik}$:

$$\mathbf{f}_{\tau,i} = (1 - m_{\tau,i})\mathbf{f}_{\tau,i}^E + m_{\tau,i}\mathbf{f}_{\tau,i}^L. \tag{7}$$

Table 4: Shared architecture of the frozen tokenizer and video world model. Parameter counts exclude the frozen tokenizer.

| Component | Configuration |
|---|---|
| Context code | Two $32 \times 40$ lanes; 2,560 tokens; fixed within a rollout. |
| Dynamic code | Four $8 \times 10$ lanes; 320 tokens per predicted frame. |
| Token quantization | Five FSQ digits with levels $(7, 5, 5, 5, 5)$. |
| Predictor backbone | Width 1,280; 20 attention heads; 15 transition layers; three appearance-refinement layers; two Context blocks; one interaction layer; MLP ratio 4; zero dropout. |
| Recurrent features | 320 dense scene tokens, 16 carriers of width 256, 64 interaction slots, and four recent predicted Dynamic states. |

The resulting correspondence carries owned source features with normalized bilinear forward splatting. Source ownership and destination coverage are kept separate: moved content is vacated at its source, while a learned reveal proposal fills newly uncovered locations. This prevents a feature from being duplicated at both its old and new positions. Zero predicted motion reduces to identity carry. We denote the complete inherited proposal by $\widetilde{\mathcal{S}}_{\tau+1}$.

### A.4 Complete Prediction and Gated Commit

The predictor reads the fixed Context code, carried dense state, rendered carriers, predicted-state history, interaction slots, and current action. It predicts all four lanes of $\widehat{\mathbf{D}}_{\tau+1}$. For each lane and cell, the readout assigns probability to three sources: newly generated content, stationary content from the previous predicted state, and content transported from a previous predicted location. Invalid persistence routes are masked. At evaluation time, the hard route is sampled deterministically from the learned posterior using an immutable key derived from the Context code, recurrent step, lane, and cell. No route has access to a future target.

Prediction and recurrent storage are deliberately distinct. Let $\mathcal{M}$ collect the writable fields of Inertial Memory and any predictor state updated by the transition. In the compact notation of Section 3.2,

$$(\widehat{\mathbf{D}}_{\tau+1}, \mathcal{M}^{\text{new}}_{\tau+1}, \mathbf{g}_{\tau+1}) = P_\theta(\widetilde{\mathcal{S}}_{\tau+1}, \mathbf{a}_\tau; \mathbf{C}), \qquad \mathcal{M}_{\tau+1} = (1 - \mathbf{g}_{\tau+1}) \odot \widetilde{\mathcal{M}}_{\tau+1} + \mathbf{g}_{\tau+1} \odot \mathcal{M}^{\text{new}}_{\tau+1}. \tag{8}$$

The predicted Dynamic code reconstructs the complete next frame. The learned gate then determines which transported values remain in memory and which are replaced by newly predicted content. Carrier descriptors are updated from their assigned dense evidence, and canonical content is updated by sampling the new dense state back into each carrier's coordinate system. The binding map, interaction slots, and predicted-state history are updated last.

One transition therefore follows the fixed order *advance carriers* $\rightarrow$ *carry existing content* $\rightarrow$ *predict the complete next video state* $\rightarrow$ *commit memory*. This is the operational meaning of carry-before-write. Training targets define losses but never replace any field of $\mathcal{S}_\tau$. Hard Dynamic IDs are used on the forward path, straight-through surrogates provide gradients through discrete decisions, and the recurrence is not detached between steps. Training and evaluation thus execute the same predicted-only state evolution.

## B Architecture and Training Details

### B.1 Frozen Video Tokenizer

We train a separate tokenizer for each domain while keeping its learnable architecture, objective, and optimization recipe identical. The copies differ only in their data readers and image preprocessing. Each tokenizer contains 120.386M parameters, uses 384 hidden and 256 latent channels, and maps a $256 \times 320$ internal canvas to the shared interface in Table 4. Every token is represented by five finite-scalar-quantization digits with levels $(7, 5, 5, 5, 5)$, giving 4,375 joint states (Mentzer et al., 2024).

Table 5: Domain-specific preprocessing and video-world-model training. "Quick/full" gives the number of validation clips evaluated every 200/1,000 steps.

| Domain | Action | WM params | WM steps | Peak-LR hold | Quick/full |
|---|---|---|---|---|---|
| RT-1 | raw 13-D | 818.746M | 60k | 2k–32k | 64/1,024 |
| CSGO | raw 51-D | 818.795M | 30k | 2k–16k | 64/128 |
| RECON | local $\Delta$ 3-D | 818.733M | 30k | 2k–16k | 13/13 |

Tokenizer training uses 33-frame clips: one Context frame and 32 future frames. An update supervises the Context frame and eight deterministically distributed future frames, always including the first and last targets and covering all four temporal quarters; validation decodes all 32 targets. The objective combines pixel reconstruction, edge and high-frequency fidelity, two-level wavelet detail, SSIM, LPIPS-VGG (Zhang et al., 2018), quantization balance, and lane-specialization losses. Motion-region and late-frame terms are introduced at fixed stages after the initial reconstruction phase. We use no adversarial loss. This objective is identical across the three domains.

All tokenizers are trained from scratch for 60k optimizer steps with AdamW, learning rate $10^{-4}$, weight decay $0.02$, and $(\beta_1, \beta_2) = (0.9, 0.95)$. The learning rate is warmed up for 2k steps and cosine-decayed to zero. Training uses BF16, global batch size 32, and gradient-norm clipping at 1.0. The completed 60k tokenizer is then frozen for video-world-model training.

### B.2 World-Model Architecture and Objective

The three domain models share all architectural and loss settings except the action projection. RT-1 and CSGO consume their raw 13-D and 51-D actions. RECON consumes normalized local-frame displacement in $x$ and $y$ together with wrapped relative yaw. These input dimensions account for the small parameter-count differences in Table 5.

The primary objective supervises all digits of the four complete Dynamic lanes and their joint code. It additionally supervises the innovation, stationary, and transported routes. A frozen-decoder objective on two frames per update supplies RGB, edge, frequency, chroma, motion-region, and temporal change signals. Memory-consistency terms supervise binding, transport, background preservation, carrier rigidity, and contact consistency from adjacent frozen-tokenizer representations. These representations are used only to compute losses; none is inserted into the recurrent state. Transport, stationary-route, contact, and deformation terms are staged in during the first 12k steps so that an untrained persistence route cannot dominate the initial predictor. The complete set of coefficients is identical across domains.

All video world models use AdamW with learning rate $10^{-4}$, minimum learning rate $10^{-5}$, weight decay $0.05$, and $(\beta_1, \beta_2) = (0.9, 0.95)$. After a 2k-step warmup, the rate is held at its peak for the interval in Table 5 and then cosine-decayed to the minimum at the final step. The predicted-only rollout curriculum is shared: AR8 for steps $[0, 4\text{k})$, AR16 for $[4\text{k}, 12\text{k})$, and AR32 thereafter. We backpropagate through the entire active rollout without teacher forcing or truncated BPTT.

Formal training uses seed 0, BF16, PyTorch distributed data parallel over 32 GPUs, global batch size 32, no gradient accumulation, activation checkpointing, and gradient clipping at 1.0. Quick validation is run every 200 steps, full AR32 validation and checkpointing every 1,000 steps. The main results use the final raw-model checkpoint—60k for RT-1 and 30k for CSGO and RECON—rather than choosing a checkpoint on the test set. No exponential moving average is used. The ablation in Section 4.4 instead fixes every compared model to 30k steps.

## C Datasets and Frozen Evaluation Protocols

All protocols were materialized as ordered episode lists before metric aggregation. A protocol records the source episode, anchor, action slice, conditioning frames, and 32 reference targets. Every method receives the same anchor and the same actions for the 32 scored transitions. A baseline may additionally use frames and aligned actions strictly before the anchor when required by its released interface, but no method observes any scored target. Table 6 summarizes the resulting contracts.

Table 6: Frozen evaluation protocols. Context is excluded from all reported metrics.

| Domain | Evaluated episodes | Targets | Resolution | Common alignment |
|---|---|---|---|---|
| RT-1 | 1,024 | 32 | $256\times320$ | $\mathbf{o}_t, \mathbf{a}_{t:t+31} \mapsto \mathbf{o}_{t+1:t+32}$ |
| CSGO | 70 of 372 test | 32 | $256\times320$ | Same current anchor and raw 51-D action sequence. |
| RECON | 128 | 32 | $240\times320$ | Same current anchor and adjacent local-motion sequence. |

### C.1 RT-1

We construct a fixed 1,024-episode protocol from RT-1 (Brohan et al., 2022). Episodes used by our development validation protocol are first excluded. Of the remaining episodes, 1,142 admit both a 16-frame observation prefix and 32 adjacent future targets. We sample 1,024 distinct episodes without replacement using seed 0 and then sort the selected episode identifiers. The anchor within an episode is generated deterministically from the episode identifier and constrained to retain the preceding 15 frames and all 32 future frames. The common current observation is $\mathbf{o}_t$; the raw 13-D action $\mathbf{a}_{t+j}$ is paired with target $\mathbf{o}_{t+j+1}$. Our model observes only $\mathbf{o}_t$, whereas a released baseline may use frames from the reserved prefix according to its native interface.

This protocol is disjoint from the data used to train and validate INERTIALWORLD. The released baselines were trained with different RT-1 partitions, so it cannot simultaneously be a held-out split for every public checkpoint. In particular, 1,015 of the 1,024 episodes occur in the released RLVR-World training subset. We therefore use this block as a paired released-checkpoint comparison under a common rollout contract, not as a claim that every baseline is evaluated on its own held-out distribution.

### C.2 CSGO

We follow the 5,500-episode CSGO corpus and the 500-episode held-out partition used by Vid2World (Huang et al., 2026). The first 128 held-out episodes are reserved for validation and the remaining 372 define the test pool. For each test episode, a deterministic per-episode key selects an anchor uniformly from indices 32 through 967. Thus, every episode retains at least 32 earlier frames for native baseline context and exactly 32 future targets. All models use the same anchor and raw 51-D action interval.

The CSGO block reported in Table 1 is a fixed 70-episode slice of this 372-episode protocol: zero-based canonical request indices $0, 5, \ldots, 345$. We state this rule explicitly so that the subset can be reconstructed exactly.

The source frames are $150\times280$ BGR images. We convert them to RGB and apply one antialiased bilinear resize to $256\times320$. The same spatial transform is applied independently to every frame and does not alter the temporal pairing between $\mathbf{a}_t$ and $\mathbf{o}_{t+1}$.

### C.3 RECON

We use the official Vid2World RECON split over 11,835 trajectories (Shah et al., 2021; Huang et al., 2026): 9,468 for training and 2,367 held out. We exclude the 17 held-out trajectories used for development validation, retain the 1,154 trajectories that admit the required native history and 32 adjacent targets, and deterministically select 128 distinct test episodes with seed 0. A second deterministic key selects one valid anchor per episode. Every method receives the same anchor and sequence of normalized local $x$–$y$ displacement and relative-yaw actions.

RECON images are decoded at $480\times640$, resized with antialiased bilinear interpolation to $240\times320$, and reflection-padded by eight pixels above and below to form the $256\times320$ internal training canvas. The padding is removed before serialization and scoring, so all reported RECON metrics use the undistorted $240\times320$ field of view. Predictions from baselines with a different native image size are transformed once to this common scoring resolution.

Table 7: Native baseline interfaces used in the paired 32-frame evaluation. $Hk$ denotes $k$ conditioning frames available to a released model; it does not denote scored output.

| Domain | Method | Native context | 32-frame rollout |
|---|---|---|---|
| RT-1 | RLVR-World | released H7 state | Four 7-frame recurrent blocks followed by a final 4-frame output chunk. |
| RT-1 | Vid2World | initial H1, then H9 | One 15-frame generation, followed by 17 single-frame sliding-window generations. |
| RT-1 | WorldGym | H1 | Official one-step latent transition applied 32 times. |
| RT-1 | DriftWorld | H4 | Official four-frame sliding window applied autoregressively for 32 outputs. |
| CSGO | Vid2World | H9 | Official T10/H9 single-frame sliding rollout, repeated 32 times. |
| CSGO | DIAMOND | H4 | Official continuous four-frame-history rollout for 32 outputs. |
| RECON | Vid2World | H4 | Two consecutive 16-frame generations with predicted feedback. |
| RECON | NWM | H4 | Official one-step four-frame sliding rollout, repeated 32 times. |
| RECON | RAE-NWM | H4 | Official latent one-step rollout, repeated 32 times before decoding. |

### C.4 Canonical Targets and Leakage Checks

For each domain, the reference frames are decoded once through the frozen data preprocessing contract and used as the canonical target tensor for every method. We verify ordered episode identifiers, anchor indices, action bytes, frame count, color order, resolution, and value range before computing a metric. For RECON, where independently installed image libraries can produce small interpolation differences, baseline predictions are paired with this single canonical target tensor after confirming that their episode and anchor metadata match. The predicted bytes are not modified by this pairing.

The scored tensor always contains the 32 future frames $\mathbf{o}_{t+1:t+32}$. Native conditioning frames, the common anchor, and any preceding history are excluded. Autoregressive feedback contains only model predictions. All methods share $\mathbf{a}_{t:t+31}$ for the scored transitions. Native multi-frame baselines may additionally consume actions aligned with their observed prefix, and blockwise baselines receive the aligned subsequence for the frames they jointly generate. No method receives a reference image from the scored interval or an action after the final predicted transition.

## D Baseline Reproduction and Metric Definitions

### D.1 Released-Checkpoint Rollout Contracts

We preserve each baseline's released observation interface, checkpoint, and autoregressive decomposition instead of forcing all methods into our single-frame state interface. Table 7 records the native context and how 32 future frames are produced. In every case, the last conditioning frame is the common anchor $\mathbf{o}_t$, and all feedback after initialization is predicted by that method.

For Vid2World, we retain the released domain-specific samplers rather than transplanting the RT-1 schedule to CSGO or RECON. WorldGym receives the released 10-D control representation obtained from the common raw 13-D RT-1 actions. DriftWorld generates on its native square canvas and is transformed to the common RT-1 scoring geometry only after prediction. NWM and RAE-NWM run at their released native resolutions; their decoded predictions are resampled once to the common RECON geometry. These transforms affect only the exported pixels, not the action encoder or the recurrent latent state.

Table 8: Exact scope of the RT-1 ablations. All variants are evaluated at 30k steps on the same ordered 1,024-episode protocol.

| Variant | Training status | Intervention |
|---|---|---|
| Plain AR | Independently trained | Removes Inertial Memory. Predicts the next Dynamic state from fixed Context, current action, rollout step, and the preceding prediction; the first step uses a learned beginning-of-sequence state. |
| Eulerian-only† | Full checkpoint | Disables carrier strength, region transport, contact, deformation, and canonical carrier updates while retaining the dense state and generic predictor context. |
| Bias-only gates† | Full checkpoint | Zeros the complete inputs to the scene, interaction-slot, and carrier-update gate projections while retaining their checkpoint biases and validity masks. |
| Full | Independently trained | Executes the complete Eulerian–Lagrangian carry and adaptive gated commit. |

### D.2 Frame Metrics and Statistical Aggregation

All RGB tensors are converted to $[0, 1]$. For predicted frame $\widehat{\mathbf{o}}$ and target $\mathbf{o}$, MSE and MAE are means over all pixels and channels, and $\mathrm{PSNR} = 10 \log_{10}(1/(\mathrm{MSE} + 10^{-8}))$. SSIM uses an $11 \times 11$ Gaussian window with standard deviation 1.5 and is evaluated on RGB channels. LPIPS uses the VGG backbone and inputs mapped to $[-1, 1]$ (Zhang et al., 2018).

Metrics are computed independently for every predicted frame. We first average the 32 frame values within an episode and then take the arithmetic mean across episodes. The uncertainty after "$\pm$" is the sample standard deviation across episode-level means ($\mathrm{ddof} = 1$), not a standard error and not variation across training seeds. PSNR is therefore a macro-average of frame-level PSNR, rather than PSNR obtained from a globally pooled MSE. The fourth main-table metric is MSE on RT-1 and MAE on CSGO and RECON, as indicated in Table 1.

Distributed evaluation partitions complete episodes only. Episode records are restored to canonical request order and aggregated in double precision on the CPU; no metric is reduced in a rank-dependent order. Deterministic kernels are enabled and TF32 is disabled. These choices make the reported statistics independent of the number of evaluation workers.

### D.3 Motion-Fidelity Diagnostics

The three diagnostics in Table 2 use one frozen RAFT estimator for every predicted and reference rollout (Teed & Deng, 2020). Let $\widehat{\mathbf{u}}_\tau$ and $\mathbf{u}^\star_\tau$ be flows estimated from adjacent predicted and reference frames. Flow EPE averages $\|\widehat{\mathbf{u}}_\tau - \mathbf{u}^\star_\tau\|_2$ over pixels and transitions. It is a discrepancy between two estimated flows, not an error against measured flow annotations. Flow Cosine evaluates their directional agreement only where the reference forward-flow magnitude is at least one pixel. Dynamic-region LPIPS instead uses reference backward flow to define a target-coordinate motion mask with the same threshold. Both masks are fixed by the reference video and shared by all methods. Aggregation again proceeds within episode and then across the same 1,024 episodes.

These quantities are protocol-specific diagnostics designed to reduce the dominance of static background. They are not substituted for the main appearance metrics, and neither their threshold nor their reported transition range is selected separately for an individual method.

## E Ablation Protocol and Interpretation

Table 3 separates one controlled training ablation from two interventions on the trained full model. The distinction matters: the trained ablation tests whether the complete model is preferable under a matched optimization budget, whereas the interventions probe which pathways the learned full transition uses at inference.

Plain AR uses the same tokenizer, training episodes, action representation, optimizer, AR8→AR16→AR32 curriculum, 30k-step budget, global batch size, and evaluation code as the full model. It contains 824.450M parameters, compared with 818.746M for the full model, so the full model's advantage cannot be attributed to larger capacity. Both are trained with seed 0 and evaluated from raw, non-EMA 30k checkpoints. The resulting comparison supports the benefit of the complete inertial state transition, but it is a one-seed architectural comparison rather than an estimate of training-seed variance.

The two † variants do not introduce new learned weights and are not retrained after their pathways are disabled. Eulerian-only therefore asks whether a checkpoint that learned to use both views remains effective when regional persistence and transport are removed; it is not a separately optimized dense-memory baseline. Bias-only gates asks whether fixed average gate behavior can replace adaptive commits. Confidence, vacated-source, and canonical-evidence masks remain active so that the intervention isolates the learned gate inputs rather than breaking tensor validity.

The ordering in Table 3 supports three bounded conclusions. First, the matched Plain AR comparison attributes a consistent predictive gain to recurrent inertial state evolution rather than parameter count. Second, the Eulerian-only intervention shows that the trained model uses the complementary carrier path instead of relying exclusively on its dense state. Third, the bias-only intervention shows that adaptive commits are preferable to a fixed learned update rate. Because the latter two variants are distribution-shifting inference probes, their absolute drops should not be used to rank the intrinsic importance of components; the controlled claim is the full-versus-Plain-AR result.