WORLDCRAFT: PERSISTENT OBJECT-STATE CONTROL IN CAMERA-NAVIGABLE VIDEO WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent video world models enable interactive camera navigation but provide limited control over individual objects. We study *persistent object-state control* in autoregressive video world models, where an object-trajectory action must remain effective across viewpoint changes, generation chunks, and intervals of temporary invisibility. This setting introduces an action-memory conflict because autoregressive memory records the last observed object state, whereas an intervening action may update that state off-screen. We present **WorldCraft**, the first camera-navigable autoregressive video world model to jointly compose camera navigation and object-trajectory control. Given a user-selected object and a 2D motion path, Normalized World Trajectory (NWT) anchors the path in a normalized world coordinate system and projects it into each camera view, producing a consistent object-action signal throughout the autoregressive rollout. Trajectory-Anchored State Persistence (TASP) combines persistent NWT guidance with selective memory refresh to maintain action-updated object states through off-screen intervals and re-entry. A pathway-selective LoRA introduces object control while retaining the pretrained camera controller. WorldCraft achieves accurate composable camera-object control and preserves camera fidelity. We further find that it maintains action-updated object states across extended intervals of complete target invisibility without off-screen trajectory supervision.

1 INTRODUCTION

A world model (Hafner et al., 2020; Ha & Schmidhuber, 2018) is a learned simulator that predicts future states given the current state and an action. Recent video-based world models realize this idea directly in pixel space, such as Genie 3 (Parker-Holder & Fruchter, 2025) and WorldPlay (Sun et al., 2026) have made impressive progress on *camera-level* interaction: users can navigate viewpoints and the model generates coherent visual continuations. Yet their action space stops at the camera. Real interaction is inherently object-centric. A robot must predict the consequence of pushing a cup along a tabletop (Wu et al., 2023; Yang et al., 2024b). A driving simulator must model a pedestrian entering the road (Wang et al., 2024a; Hu et al., 2023). An interactive game must allow users to move entities rather than only the viewpoint. In all these cases, the action is a continuous trajectory attached to a specific object. Without such object-level actions, world models are more like passive scene observers than manipulable environments.

Object-level actions in an autoregressive world model differ from trajectory guidance within a fixed generation window (Chu et al., 2025; Wu et al., 2024; Wang et al., 2024b). Fixed-window methods represent a trajectory as a sequence of visible spatial conditions. We define **persistent object-state control** as the ability to maintain an action-induced state transition across viewpoint changes and autoregressive chunk boundaries, including intervals in which the target is outside the current observation. This requirement creates two coupled problems. First, *observation-dependent coordinates*: screen-space motion entangles object displacement with camera ego-motion, so the same intended object action changes under a moving viewpoint. Second, *off-screen object-state updates* create an action-memory conflict. While the target is fully outside the current view, an object-level trajectory may continue moving it from its last observed location to a new state. Autoregressive visual memory, however, remains anchored to the pre-exit state. Upon re-entry, the model must render the same object at its trajectory-specified location rather than restore the stale location stored in memory.

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.

Table 1: **Comparison of control and generation settings.** "Off-screen object-state update" denotes changing a controlled object's state through an object-level action while the target is fully outside the camera view.

Method	Setting	Camera	Object	AR rollout	Off-screen object-state update
Motion-controlled generation and editing					
DragAnything (Wu et al., 2024) (ECCV)	Fixed-window generation	-	✓	-	-
Wan-Move (Chu et al., 2025) (NeurIPS)	Fixed-window generation	-	✓	-	-
MotionCtrl (Wang et al., 2024b) (SIGGRAPH)	Fixed-window generation	✓	✓	-	-
Free-Form Motion Control (Shuai et al., 2025) (ICCV)	Fixed-window generation	✓	✓	-	-
Edit-by-Track (Lee et al., 2026) (CVPR)	Fixed-window editing	✓	✓	-	-
Interactive video world models					
Yume (Mao et al., 2025)(Arxiv)	Camera-navigable world model	✓	-	✓	-
Matrix-Game 2.0 (He et al., 2025)(Arxiv)	Camera-navigable world model	✓	-	✓	-
GameCraft (Li et al., 2025a)(Arxiv)	Camera-navigable world model	✓	-	✓	-
Genie 3 (Parker-Holder & Fruchter, 2025)	Camera-navigable world model	✓	-	✓	-
WorldPlay (Sun et al., 2026) (ICML 2026)	Camera-navigable world model	✓	-	✓	-
WorldCraft (ours)	Persistent object-state control	✓	✓	✓	✓

We introduce **WorldCraft**, a framework for persistent object-state control in camera-navigable autoregressive video world models. Given a user-selected object and a 2D motion path, WorldCraft composes object-trajectory control with continuous camera navigation. Normalized World Trajectory (NWT) anchors the user path in a normalized reference coordinate system and re-projects it under each camera pose, separating intended object motion from camera-induced image displacement across viewpoints and autoregressive chunks. After each chunk, NWT updates its visual anchor and depth estimate from the latest generated observation to limit accumulated re-projection drift. Trajectory-Anchored State Persistence (TASP) uses NWT as global object-state guidance during off-screen intervals and filters stale pre-exit memories at re-entry, preserving object appearance and scene context while enforcing the updated position. We implement trajectory adaptation with Spatial-Pathway LoRA (SP-LoRA), which equips the backbone with object-level trajectory control while preserving its pretrained camera-navigation capability.

Table 1 contrasts WorldCraft with fixed-window motion-control methods and camera-navigable world models. To our knowledge, WorldCraft is the first camera-navigable autoregressive video world model to jointly compose camera navigation and object-trajectory control within a single rollout. Importantly, its training data contain no trajectories that move a controlled object while it is fully off-screen. We therefore evaluate *zero-shot* transfer to this unseen regime using a dedicated benchmark, where the trajectory updates an invisible target and the returning camera tests whether it reappears at the specified location. Together, NWT propagates the trajectory-updated spatial state and TASP reconciles it with visual memory at re-entry, maintaining the update through up to four invisible generation chunks. Our contributions are:

1. **Joint camera-object control in autoregressive world models.** To our knowledge, WorldCraft is the first camera-navigable autoregressive video world model to jointly compose camera navigation and object-trajectory control within a single rollout.
2. **Autoregressive camera-decoupled trajectory control.** NWT decouples object motion from camera-induced displacement and iteratively updates its anchor and depth across autoregressive chunks.
3. **Zero-shot persistent object-state control.** Without training trajectories that move fully off-screen targets, NWT and TASP jointly propagate trajectory-updated states through invisible intervals and recover controlled objects at the specified locations upon re-entry.

2 Related work

Interactive video world models and state-evolution benchmarks. Recent video world models make controllable simulation in pixel space increasingly practical. GameNGen (Valevski et al., 2025) and GameGen-X (Che et al., 2024) demonstrate action-conditioned game video generation (Yu et al., 2025), while Yume (Mao et al., 2025), GameCraft (Li et al., 2025a), Matrix-Game 2.0 (He et al., 2025), and WorldPlay (Sun et al., 2026) advance long-horizon camera navigation and scene

revisitation. WorldPlay is our direct backbone because it combines camera-action conditioning with autoregressive visual memory. Such memory can preserve previously observed scene content during camera excursions. However, these models do not expose an object-level action that changes a selected object's state while it is outside the current view. WorldCraft adds this object-level state-update capability while retaining camera navigation and scene memory. MIND (Ye et al., 2026) evaluates memory consistency and action control through closed-loop scene revisitation, while STEVO-Bench (Ma et al., 2026) tests whether naturally evolving processes continue under camera lookaway. These benchmarks establish unobserved state evolution as a distinct challenge for video world models. Our OSS benchmark targets a complementary trajectory-conditioned setting: a user trajectory updates a selected object's position while it is fully outside the view, and the same autoregressive camera rollout must recover that object at the action-specified state upon re-entry.

Camera and object motion control. Trajectory-guided methods provide object control through motion paths (Yin et al., 2023; Wu et al., 2024) or displaced first-frame features (Chu et al., 2025). MotionCtrl (Wang et al., 2024b) jointly controls camera poses and object trajectories in fixed-window generation, Free-Form Motion Control (Shuai et al., 2025) uses explicit 6D camera and object poses, and Edit-by-Track (Lee et al., 2026) edits a source video with paired 3D point tracks. These methods establish camera-object control for fixed-duration generation or editing under inputs ranging from 2D paths to explicit 3D constraints. WorldCraft instead studies persistent object-state control in a camera-navigable autoregressive world model. From a user-selected object and 2D motion path, NWT produces view-dependent conditions that evolve with the camera and generated state across chunks. The action must compose with camera navigation, remain effective through long-horizon memory, and update a fully invisible target. Our quantitative comparisons accordingly use trajectory methods with a matched 2D-path interface and camera-controlled world models that support autoregressive long-video evaluation.

3 METHOD

WorldCraft converts a transient trajectory condition into persistent object-state control for an autoregressive video world model. Figure 1 summarizes the pipeline. After describing the backbone and in-context trajectory injection (§3.1-3.2), we present NWT for camera-decoupled trajectory representation and iterative updates across chunks (§3.3), followed by TASP for reconciling off-screen object-state updates with autoregressive visual memory (§3.4). We then describe Spatial-Pathway LoRA for backbone-preserving trajectory adaptation (§3.5). The method proceeds from action representation to state persistence: NWT determines the trajectory coordinates across views and chunks, and TASP reconciles the resulting action state with autoregressive memory. Spatial-Pathway LoRA provides a lightweight implementation for adapting the backbone to this trajectory condition.

3.1 PRELIMINARIES AND NOTATION

WorldCraft builds on WorldPlay (Sun et al., 2026), an autoregressive video DiT (Peebles & Xie, 2023; Dosovitskiy et al., 2021) based on HunyuanVideo-1.5 (Wu et al., 2025). Its input concatenates a 32-channel noisy latent $\mathbf{z}_t$, a 32-channel image condition $\mathbf{c}_{\text{img}}$, and a task mask $\mathbf{m}$. WorldPlay generates video chunk by chunk, conditions camera actions through the action encoder `action_in` and per-block ProPE projections $\texttt{prope_proj}^{(l)}$ (Li et al., 2025b), and attends to cached key-value memories from previous chunks (Yin et al., 2025). We denote frame-t camera intrinsics and world-to-camera extrinsics by $\mathbf{K}_t \in \mathbb{R}^{3\times3}$ and $\mathbf{E}_t \in \mathrm{SE}(3)$, and perspective projection by $\pi(\mathbf{K}, \mathbf{E}, \mathbf{P})$.

3.2 IN-CONTEXT TRAJECTORY CONDITIONING

We inject trajectory information by replacing the image-conditioning channels (32-64) with first-frame latent features displaced to the target positions. Given the first-frame latent $\mathbf{z}_0 \in \mathbb{R}^{C\times1\times H\times W}$ and N point trajectories $\{\mathbf{p}_t^{(n)}\}_{n,t}$, the trajectory condition $\hat{\mathbf{c}}_{\text{traj}}$ is:

$$\hat{\mathbf{c}}_{\text{traj}}[:, t, h_t^{(n)}, w_t^{(n)}] \leftarrow \mathbf{z}_0[:, 0, h_0^{(n)}, w_0^{(n)}], \quad \forall\, n, t \tag{1}$$

where $(h_t^{(n)}, w_t^{(n)})$ is the latent-space coordinate of track n at frame t. Unassigned positions remain zero. The model input becomes $[\mathbf{z}_t ; \hat{\mathbf{c}}_{\text{traj}} ; \mathbf{m}]$, preserving full compatibility with the pretrained

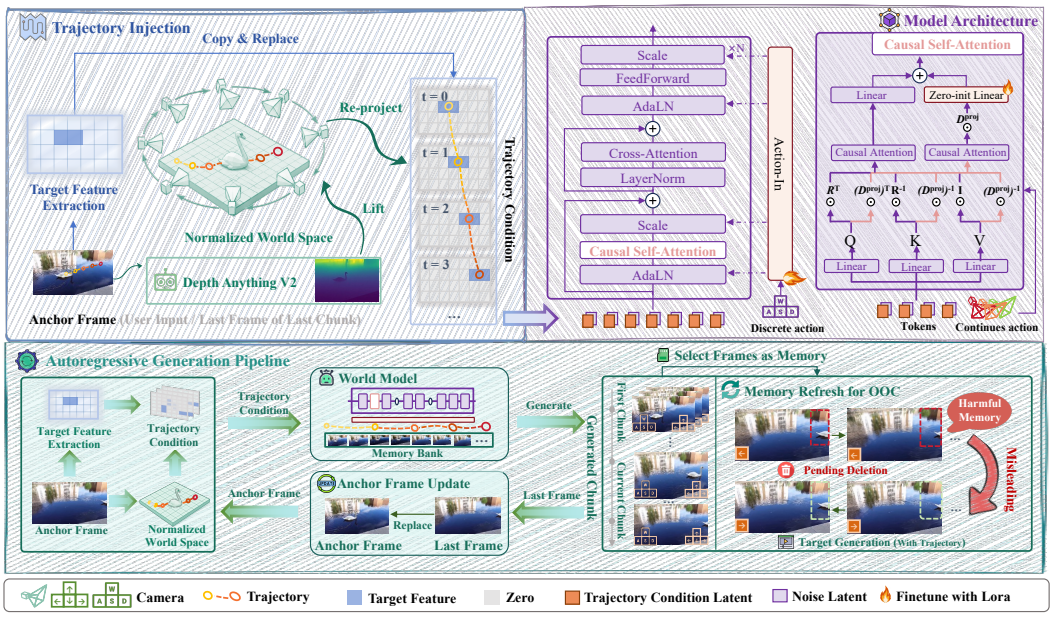


Figure 1: **WorldCraft overview. (Top-left)** WorldCraft lifts a user-specified 2D trajectory into a camera-decoupled normalized world space and re-projects it into per-frame trajectory conditions under the given camera actions. **(Top-right)** The trajectory and camera controls are injected through a lightweight pathway-selective LoRA on the spatial-control pathway, while the backbone attention and MLP layers remain frozen. **(bottom)** During autoregressive generation, WorldCraft updates the anchor frame and memory bank across chunks, and refreshes outdated memories to support long-horizon out-of-camera object reasoning.

PatchEmbed (Wu et al., 2025). This design yields an informative prior even *before* trajectory-specific training: displaced first-frame features serve as positional cues that the base model's spatial understanding partially decodes, producing coarse trajectory following at zero shot.

3.3 Normalized World Trajectory

At inference, the user clicks the target object in the initial frame and sketches a 2D path specifying its desired motion. NWT acts as a lightweight *action compiler*, converting this input into a camera-aware control signal that remains meaningful beyond the current image. A raw screen-space path entangles the object's intended displacement with apparent motion caused by camera ego-motion. We therefore anchor the trajectory to the first-frame camera coordinate system and dynamically re-project it under the current camera pose. For autoregressive generation, NWT further updates the visual anchor and target depth after each chunk, limiting accumulated re-projection drift while retaining the same user-specified world-space trajectory.

Definition. Given a user-specified screen trajectory $\{\mathbf{p}_t^{\text{user}}\}$ and a reference depth d, we first lift each point to a normalized coordinate on the first-frame reference plane:

$$\mathbf{q}_t = \begin{pmatrix} (x_t - c_x)/f_x \\ (y_t - c_y)/f_y \end{pmatrix}, \tag{2}$$

where f_x, f_y, c_x, c_y are the first-frame intrinsic parameters. Intuitively, $\mathbf{q}_t$ describes the object's position on a first-frame-anchored reference plane, rather than its instantaneous screen-space location. At each generation step, we re-project this anchored coordinate into the current camera view:

$$\mathbf{p}_t^{\text{anchored}} = \pi\Big(\mathbf{K}_t,\ \mathbf{E}_t\mathbf{E}_0^{-1} \cdot \text{lift}(\mathbf{q}_t, d)\Big), \tag{3}$$

where $\mathbf{E}_t$ denotes the world-to-camera extrinsic matrix and $\text{lift}(\mathbf{q}_t, d)$ maps the normalized coordinate to a 3D point on the reference depth plane. This re-projection automatically folds camera-induced pixel displacement into the trajectory signal consumed by the model.

Algorithm 1 Inference with Normalized World Trajectory

Require: First frame $\mathbf{I}_0$; user screen trajectory $\{\mathbf{p}_t^{\text{user}}\}_{t=0}^{T}$; camera pose sequence $\{\mathbf{E}_t\}_{t=0}^{T}$; first-frame intrinsics $\mathbf{K}_0$; chunk size C.
Ensure: Generated video $\{\mathbf{I}_t\}_{t=1}^{T}$.
 # Stage 1: lift user trajectory into world space (once).
$d \leftarrow \mathcal{D}(\mathbf{I}_0)[\mathbf{p}_0^{\text{user}}]$ ▷ depth at initial click
for $t = 0, \dots, T$ **do**
  $\mathbf{q}_t \leftarrow \big((x_t^{\text{user}} - c_x)/f_x,\ (y_t^{\text{user}} - c_y)/f_y\big)$
end for
 # Stage 2: autoregressive generation with iterative depth refinement.
for chunk $k = 1, \dots, \lceil T/C \rceil$ **do**
  for t in chunk k **do**
    $\mathbf{p}_t^{\text{anchored}} \leftarrow \pi\big(\mathbf{K}_t,\ \mathbf{E}_t \cdot \mathbf{E}_0^{-1} \cdot \text{lift}(\mathbf{q}_t, d)\big)$
  end for
  $\mathbf{I}_{\text{chunk } k} \leftarrow \text{WorldModel}\big(\mathbf{I}_{<\text{chunk } k},\ \{\mathbf{p}_t^{\text{anchored}}\},\ \{\mathbf{E}_t\}\big)$
  $d \leftarrow \text{RefineDepth}(\mathbf{I}_{\text{chunk } k},\ \mathbf{p}_{t_{\text{last}}}^{\text{anchored}},\ d)$
end for
return $\{\mathbf{I}_t\}_{t=1}^{T}$

Composable camera-object control. Trajectories extracted from videos by point tracking are screen-space observations that naturally entangle object motion with camera motion:

$$\mathbf{p}_t = \pi\big(\mathbf{K}_t,\ \mathbf{E}_t,\ \mathbf{P}_{\text{world}}(t)\big), \tag{4}$$

At inference time, however, users typically specify the desired object motion without compensating for the camera trajectory, leading to a mismatch if the raw user path is used directly as screen-space conditioning. NWT closes this gap by anchoring the user trajectory in the first-frame coordinate system and re-projecting it under the current camera pose. As a result, user-specified object motion and model-driven camera motion can be composed without requiring the user to manually anticipate camera-induced screen displacement.

Depth estimation and iterative anchor refinement. Eq. 3 requires the object depth d, which we initialize by querying the monocular depth map (Yang et al., 2024a) at the user's initial click position: $d_0 = \mathcal{D}(\mathbf{I}_0)[\mathbf{p}_0^{\text{user}}]$. In long autoregressive rollouts, however, keeping both the depth and the visual anchor fixed to the first frame can accumulate projection errors across chunks. We therefore apply *iterative anchor refinement*: after each generated chunk, we update the autoregressive anchor to the latest reliable frame, re-estimate the object depth from that frame, and use the updated anchor-depth pair for subsequent re-projection. This forms a closed-loop correction that keeps the trajectory condition aligned with the generated video, reducing geometric drift while preserving the normalized world-space trajectory as the global control signal.

3.4 Trajectory-Anchored State Persistence

A world model is partially observed: leaving the camera view removes an object from the observation, not from the simulated world. An *off-screen object-state update* occurs when an object-level trajectory moves a fully invisible target away from its last observed location. Autoregressive memory still encodes the pre-exit position, creating a conflict with the trajectory-specified state required at re-entry. TASP reconciles these two sources of state: NWT carries the updated object position through the invisible interval, while selective filtering prevents stale pre-exit memory from restoring the old position. The remaining memory continues to provide appearance, identity, and scene context.

(i) Persistent trajectory state. The normalized trajectory $\{\mathbf{q}_t\}$ remains defined independently of object visibility. During the camera-away interval (t_0, t_1), the trajectory continues updating the target state across autoregressive chunks even though the target is not observed. When the camera returns at t_1, Eq. 3 maps the current trajectory state into the returning view, providing the action-updated re-entry position.

(ii) Pre-exit memory filtering. When the camera turns away from an object at time t_0 and returns at t_1, the autoregressive memory $\mathcal{M}_{t_0}$ holds a *frozen state snapshot*: its keys and values encode the

object at its pre-departure screen location. At t_1, attention retrieval confidently reproduces that stale location even if the trajectory has moved the object elsewhere. We resolve this with *pending deletion with dynamic mask*: at each re-entry chunk, we identify memory frames in the pre-exit zone (the last k temporal latent before off-screen happens) and mask them from the retrieval set if their FOV similarity with the current chunk exceeds a threshold τ:

$$\mathrm{sim}_{\mathrm{FOV}}(\mathbf{V}_f, \mathbf{V}_{\mathrm{cur}}) > \tau \implies f \notin \mathcal{M} \tag{5}$$

where $\mathbf{V}_f$ and $\mathbf{V}_{\mathrm{cur}}$ are the view matrices of memory frame f and the current chunk's first frame. Frames outside the pre-exit zone or with dissimilar FOV are retained, preserving the appearance prior. The two mechanisms are complementary. The persistent trajectory supplies the updated location at re-entry, while pre-exit filtering suppresses stale positional evidence and retains the remaining memory as an appearance and scene prior.

3.5 Backbone-preserving trajectory adaptation

Camera actions and object trajectories differ in scope but both regulate where visual content appears. A camera action produces a scene-level transformation, whereas a trajectory moves a selected instance. We therefore hypothesize that trajectory adaptation should reuse the backbone's spatial-control pathway rather than broadly modify attention and feature-mixing layers. Full-parameter trajectory fine-tuning in Appendix §C supports this hypothesis: relative weight changes concentrate on the action encoder and ProPE projections, which change by $10\text{-}25\times$ more than attention and MLP layers. Motivated by this observation, *Spatial-Pathway LoRA* (Hu et al., 2022) applies low-rank updates only to `action_in` and the per-block `prope_proj` layers, while freezing the remaining backbone. The resulting $\sim$50M-parameter adapter (0.6% of the backbone) adds trajectory control while limiting interference with the pretrained camera controller. Appendix §C gives the complete weight-change analysis, update formulation, layer-selection ablation, and training-strategy comparison.

4 Experiments

4.1 Implementation details

Base model and training. We build on WorldPlay (Sun et al., 2026), an 8B-parameter video world model based on HunyuanVideo-1.5 (Wu et al., 2025). WorldCraft follows a three-stage progressive schedule. Stage 0 performs real-video domain adaptation. Stage 1 learns trajectory control from static-camera data using BI attention and SP-LoRA. Stage 2 transfers this capability to dynamic-camera data using AR attention. Details are provided in Appendix §A.

Evaluation data. All evaluation splits are disjoint from training. The **Trajectory Accuracy (TA)** set contains 50 static-camera clips with SAM 2 object masks (Ravi et al., 2025) and CoTracker trajectories (Karaev et al., 2024). The **Camera Fidelity (CF)** set contains 50 dynamic-camera clips without object actions, using per-latent intrinsics and extrinsics estimated by ViPE (Huang et al., 2025a). The **Composable** set contains 45 clips evenly divided among small ($< 15°$), medium (15-45°), and large ($\geq 45°$) camera rotations, with both camera and object controls. The **Off-Screen State (OSS)** set contains 50 held-out 253-frame rollouts with mirrored camera-away-and-return paths. The trajectory updates the fully invisible target before the reversed camera path reveals its trajectory-specified re-entry state; we evaluate one and four consecutive invisible chunks. No training trajectory moves a controlled target while fully off-screen.

Evaluation horizon and metrics. We evaluate trajectory control with Trajectory Error (TE), the mean CoTracker (Karaev et al., 2024) L2 pixel error between the tracked object and specified path, and visual quality with VBench++ (Huang et al., 2025b; 2024), PSNR/SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018). Camera control is measured by rotational, translational, and combined Relative Pose Error (RPE) after Sim(3) Umeyama alignment (Umeyama, 1991). For OSS, $\mathbf{RTE_f}$ measures final re-entry error relative to the trajectory-specified location, and $\mathbf{RID_{pre}}$ measures DINO (Oquab et al., 2024) identity similarity to the pre-exit target. Background-only **MEt3R** (Asim et al., 2026) and relative-depth consistency (**DepthC**) are computed on matched outbound/return views with the target's swept region excluded.

Table 2: **Trajectory control** under static camera (61 frames, 50 clips). All methods share the same first frame and trajectory condition. Best in bold.

	Visual Quality				VBench++			
Method	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**DINO↑**	**SubjC↑**	**BgC↑**	**Temp↑**	**TE↓**
DragAnything (Wu et al., 2024) (ECCV 2024)	15.97	0.600	0.468	0.777	0.896	0.913	0.938	39.86
Wan-Move (Chu et al., 2025) (NeurIPS 2025)	16.42	0.592	0.375	0.782	0.927	0.943	0.985	44.08
WorldCraft (ours)	**17.23**	**0.616**	**0.363**	**0.807**	**0.942**	**0.945**	**0.989**	**38.90**

Table 3: **Camera fidelity** on camera-only input. WorldCraft preserves camera accuracy at 61 frames and outperforms all methods at the 253-frame extended horizon.

	Short-term (61 frames)						Long-term (253 frames)		
Method	**RPE_{rot}↓**	**RPE_{trans}↓**	**RPE_{cam}↓**	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**RPE_{rot}↓**	**RPE_{trans}↓**	**RPE_{cam}↓**
Yume (Mao et al., 2025)	0.261	0.0143	0.0169	12.39	0.2931	0.5718	0.374	0.0247	0.0285
Matrix-Game 2.0 (He et al., 2025)	0.342	0.0137	0.0196	12.96	0.3235	0.5326	0.162	0.0243	0.0261
GameCraft (Li et al., 2025a)	0.252	**0.0130**	**0.0157**	12.42	0.2861	0.5529	0.198	0.0243	0.0265
WorldPlay (base) (Sun et al., 2026)	**0.120**	0.0155	0.0165	13.77	0.3434	0.4700	0.130	0.0262	0.0276
WorldCraft (ours)	0.131	0.0161	0.0170	**13.95**	**0.3474**	**0.4621**	**0.123**	**0.0225**	**0.0233**

4.2 Quantitative results

Evaluation protocols. We report numerical comparisons only when the conditioning interface and evaluation protocol can be matched. For trajectory control, DragAnything (Wu et al., 2024), Wan-Move (Chu et al., 2025), and WorldCraft receive the same initial frame and 2D target trajectory (Table 2). For camera fidelity, we compare against the camera-controlled world models Yume (Mao et al., 2025), Matrix-Game 2.0 (He et al., 2025), GameCraft (Li et al., 2025a), and WorldPlay (Sun et al., 2026) under the same camera-action sequences (Table 3). Additional composable-control and DROID results are provided in Appendix §D.

Trajectory control quality. Under static camera with an identical first frame and object-trajectory condition, WorldCraft achieves the lowest trajectory error (TA) while simultaneously producing the best pixel fidelity (PSNR/SSIM (Wang et al., 2004)/LPIPS (Zhang et al., 2018)), semantic consistency (DINO), and VBench++ consistency scores across all 50 TA clips (Table 2).

Camera fidelity. On camera-only inputs, WorldCraft retains the camera-control capability of the base model. At 61 frames, its RPE_{rot} is 0.131, compared with 0.120 for WorldPlay, and far below the next-best external baseline (0.252). At 253 frames, a horizon $4\times$ longer than the main protocol, WorldCraft further reduces the error to 0.123, outperforming WorldPlay (0.130). These results show that object-level trajectory control preserves camera fidelity and slightly improves long-horizon stability (Table 3).

4.3 Qualitative results

We present qualitative comparisons along three axes: Trajectory control (Figure 2), long-horizon camera rollout with off-camera demonstration (Figure 3), and part-level control, multi-object control, long-term control (253-frame) of WorldCraft in Figure 4.

Trajectory control. Figure 2 compares WorldCraft with representative baselines. WorldPlay produces plausible scenes but does not support precise object-level trajectories. Under sparse trajectory signals, WorldCraft follows the prescribed path throughout the rollout while preserving the target and scene.

Off-screen state propagation. Figure 3 visualizes the action-memory conflict quantified in Table 4. The goose moves right while the camera pans left, remains fully outside the observation, and later returns along the mirrored camera path. Camera-only baselines can revisit the scene but cannot execute the object action. WorldCraft instead carries the trajectory-conditioned state through the invisible interval and renders the same object at its updated position upon re-entry. The model receives no matched supervision for the off-screen/re-entry regime, while the intended motion remains specified by the input trajectory.

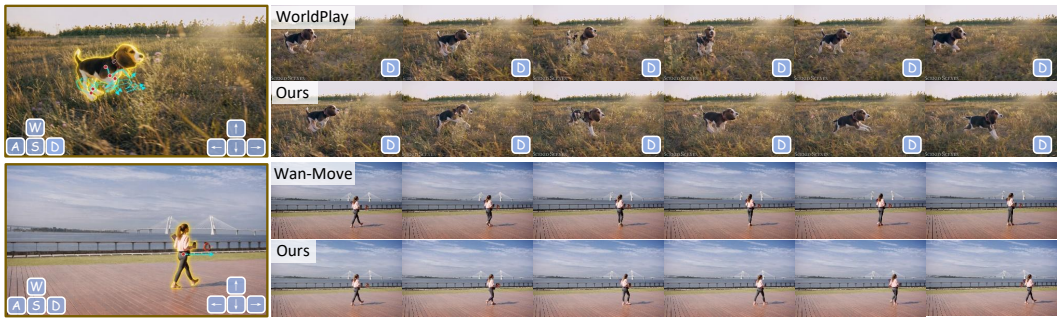


Figure 2: **Qualitative comparison of trajectory control. WorldCraft** achieves precise and composable controllability, jointly controlling camera motion and target-object trajectories.

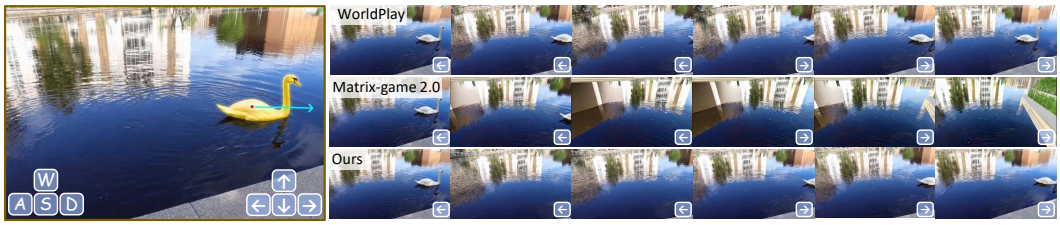


Figure 3: **Long-horizon comparisons with off-camera motion.** Given the same initial frame and camera actions, the goose moves right while the camera pans left and then returns. **WorldCraft** maintains scene consistency and, via TASP, recovers the goose at the correct off-camera-updated position when it re-enters view, whereas baselines either lose scene consistency or cannot track the off-camera object state.

WorldCraft also supports (i) *Part-level control* (Figure 4, top) and (ii) *Multi-object control* simultaneously (Figure 4, middle), even in (iii) *Long-horizon generation* (Figure 4, bottom), extending to 253 frames ($\sim$10.5 s) with composable camera-object control.

4.4 Zero-shot off-screen object-state propagation

The OSS benchmark tests whether object control learned from visible trajectories transfers to fully unobserved motion. Because no training trajectory moves a controlled target while it is fully off-screen, this evaluation is zero-shot with respect to the tested state-update regime. With NWT and TASP, increasing target invisibility from one to four chunks changes full-rollout TE from 38.48 to 40.03 and final re-entry error from 32.70 to 35.11 (Table 4(a)). Identity and background consistency are more sensitive, so our robustness claim is limited to the evaluated four-chunk range.



Figure 4: **Extended capabilities. Part**: part-level control-the shield follows the trajectory while the body stays still. **Multi**: multi-object control-three objects steered simultaneously along independent trajectories. **Long**: 253-frame autoregressive rollout with long trajectory ($\sim$10.5 s at 24 fps).

Table 4: **Trajectory-conditioned off-screen object-state propagation** on the 50-video OSS benchmark (253 frames). (a) Full TASP under a fourfold increase in target invisibility. (b) Matched memory ablation at four off-screen chunks.

Setting	$TE_{mean}\downarrow$	$RTE_f\downarrow$	$RID_{pre}\uparrow$	MEt3R$\downarrow$	DepthC$\downarrow$	$RPE_{rot}\downarrow$	$RPE_{trans}\downarrow$	$RPE_{cam}\downarrow$
(a) Robustness to off-screen duration (Full TASP)								
1 off-screen chunk	38.48	32.70	0.805	0.173	0.0665	0.126	0.0165	0.0194
4 off-screen chunks	40.03	35.11	0.717	0.196	0.0755	0.137	0.0182	0.0217
(b) Action-memory reconciliation at 4 off-screen chunks								
Default WorldPlay memory	52.10	48.20	0.618	0.202	0.0765	0.139	**0.0171**	**0.0215**
TASP selective reconciliation	**40.03**	**35.11**	**0.717**	**0.196**	**0.0755**	**0.137**	0.0182	0.0217

Table 5: **NWT representation ablation.** Trajectory error on the composable set (61 frames, 45 clips) across camera-rotation magnitudes. World-space trajectories with iterative depth refinement perform best, especially under large rotations.

Representation	TE (small rot.) $\downarrow$	TE (mid rot.) $\downarrow$	TE (large rot.) $\downarrow$
Pixel space (raw user traj.)	35.79	40.61	45.33
World space + single-shot depth	33.82	37.69	41.28
World space + iterative depth	**30.63**	**32.10**	**34.65**

Table 4(b) isolates the contribution of TASP under four invisible chunks. Both variants receive the same NWT trajectory state and differ only in memory handling. Default WorldPlay memory retains stale pre-exit evidence, whereas TASP reconciles memory with the trajectory-updated state. TASP reduces final re-entry error from 48.20 to 35.11 (27.2%), improves re-entry identity from 0.618 to 0.717, and largely preserves background geometry and camera control. Together, the duration study and matched ablation show that NWT and TASP generalize visible-object control to zero-shot off-screen state propagation.

4.5 Ablation Studies

Normalized world-space trajectory. Under the same click-and-sketch interface, we isolate two design choices within NWT: (i) pixel-space versus world-space coordinates, and (ii) single-shot versus iterative monocular depth estimation. Table 5 reports trajectory error across three camera-rotation magnitudes. Compared with raw pixel-space conditioning, world-space trajectories consistently improve trajectory accuracy, showing that anchoring trajectories in 3D space better composes object motion with camera. Iterative depth refinement provides an additional gain, especially under large camera rotations, where repeated re-estimation helps correct projection drift accumulated during autoregressive rollout.

Backbone-preserving adaptation. Full fine-tuning slightly reduces TE from 38.90 to 37.20 but increases rotational RPE from 0.131 to 0.237, whereas conventional Q/K/V+MLP LoRA uses $\sim 4\times$ more trainable parameters and performs worse on both metrics. Spatial-Pathway LoRA therefore provides the best control-preservation trade-off; complete layer and curriculum ablations are reported in Appendix §C.

5 Conclusion

We introduced WorldCraft for persistent object-state control in camera-navigable autoregressive video world models. The central problem is maintaining an action-induced state transition as observations change and visual memory retains stale evidence. NWT represents a lightweight 2D trajectory as a camera-aware action across viewpoints and chunks. TASP preserves useful visual context while suppressing memories that conflict with the updated object position, and SP-LoRA retains the inherited camera controller. Without matched off-screen training supervision, WorldCraft propagates object state across up to four invisible chunks and recovers the target at its updated location. These results extend interactive video world models from viewpoint navigation to persistent object-state control under partial observation.

ETHICS STATEMENT

WorldCraft is intended for research on controllable video world models. Its generated rollouts are not physically validated predictions and should not be used directly for safety-critical decision making. As with other generative video systems, the model may reproduce biases or artifacts in its training data and may be misused to create misleading visual content.

REPRODUCIBILITY STATEMENT

The main paper specifies the model architecture, trajectory conditioning, inference procedure, evaluation protocols, and metrics. Appendix §A reports checkpoint initialization and training configurations; Appendix §B describes data curation; Appendix §C provides implementation details and ablations; and Appendix §D gives supplementary evaluation protocols and results.

AI USE STATEMENT

Generative AI tools, including OpenAI Codex, were only used to aid and polish the writing, including language editing, organization, and presentation. The authors reviewed all AI-assisted text and take full responsibility for the final content of this work.

REFERENCES

Mohammad Asim, Christopher Wewer, Thomas Wimmer, Bernt Schiele, and Jan Eric Lenssen. Met3r: Measuring multi-view consistency in generated images, 2026. URL https://arxiv.org/abs/2501.06336.

Haoxuan Che, Xuanhua He, Quande Liu, Cheng Jin, and Hao Chen. GameGen-X: Interactive open-world game video generation. *arXiv preprint arXiv:2411.00769*, 2024.

Ruihang Chu, Yefei He, Zhekai Chen, Shiwei Zhang, Xiaogang Xu, Bin Xia, Dingdong Wang, Hongwei Yi, Xihui Liu, Hengshuang Zhao, Yu Liu, Yingya Zhang, and Yujiu Yang. Wan-Move: Motion-controllable video generation via latent trajectory guidance. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2512.08765.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

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

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

Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, Baixin Xu, Hao-Xiang Guo, Kaixiong Gong, Cyrus Wu, Wei Li, Xuchen Song, Yang Liu, Eric Li, and Yahui Zhou. Matrix-Game 2.0: An open-source, real-time, and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025.

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

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 (ICLR)*, 2022.

Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, Jiawei Ren, Kevin Xie, Joydeep Biswas, Laura Leal-Taixe, and Sanja Fidler. ViPE: Video pose engine for 3d geometric perception. *arXiv preprint arXiv:2508.10934*, 2025a.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818, 2024.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b. doi: 10.1109/TPAMI.2025.3633890.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker: It is better to track together. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

Alexander Khazatsky et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2403.12945.

Yao-Chih Lee, Zhoutong Zhang, Jiahui Huang, Jui-Hsien Wang, Joon-Young Lee, Jia-Bin Huang, Eli Shechtman, and Zhengqi Li. Generative video motion editing with 3d point tracks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18306–18318, 2026.

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*, 2025a.

Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding, 2025b. URL https://arxiv.org/abs/2507.10496.

Ziqi Ma, Mengzhan Liufu, and Georgia Gkioxari. Out of sight, out of mind? evaluating state evolution in video world models, 2026.

Xiaofeng Mao, Shaoheng Lin, Zhen Li, Chuanhao Li, Wenshuo Peng, Tong He, Jiangmiao Pang, Mingmin Chi, Yu Qiao, and Kaipeng Zhang. Yume: An interactive world generation model. *arXiv preprint arXiv:2507.17744*, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Jack Parker-Holder and Shlomi Fruchter. Genie 3: A new frontier for world models. Google DeepMind blog post, August 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, 2023. doi: 10.1109/ICCV51070.2023.00387.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations (ICLR)*, 2025.

Xincheng Shuai, Henghui Ding, Zhenyuan Qin, Hao Luo, Xingjun Ma, and Dacheng Tao. Free-form motion control: Controlling the 6d poses of camera and objects in video generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12449–12458. IEEE, 2025.

Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. WorldPlay: Towards long-term geometric consistency for real-time interactive world modeling. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2512.14614.

Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991. doi: 10.1109/34.88573.

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

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. DriveDreamer: Towards real-world-drive world models for autonomous driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 55–72, 2024a. doi: 10.1007/978-3-031-73195-2_4.

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.

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. MotionCtrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024b. doi: 10.1145/3641519.3657518.

Bing Wu, Chang Zou, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Jack Peng, Jianbing Wu, Jiangfeng Xiong, Jie Jiang, et al. HunyuanVideo 1.5 Technical Report. *arXiv preprint arXiv:2511.18870*, 2025.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. DayDreamer: World models for physical robot learning. In *Proceedings of the 6th Conference on Robot Learning*, volume 205, pp. 2226–2240. PMLR, 2023.

Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. DragAnything: Motion control for anything using entity representation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything V2. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a.

Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations (ICLR)*, 2024b. Outstanding Paper Award.

Yixuan Ye, Xuanyu Lu, Yuxin Jiang, Yuchao Gu, Rui Zhao, Qiwei Liang, Jiachun Pan, Fengda Zhang, Weijia Wu, and Alex Jinpeng Wang. MIND: Benchmarking memory consistency and action control in world models, 2026.

Shengming Yin, Chenfei Wu, Jian Liang, Jie Shi, Houqiang Li, Gong Ming, and Nan Duan. DragNUWA: Fine-grained control in video generation by integrating text, image, and trajectory. *arXiv preprint arXiv:2308.08089*, 2023.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22963–22974, 2025. doi: 10.1109/CVPR52734.2025.02138.

Jiwen Yu, Yiran Qin, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. GameFactory: Creating new games with generative interactive videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11590–11599, 2025.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

Yang Zhou et al. OmniWorld: A multi-domain and multi-modal dataset for 4d world modeling. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2509.12201.

Table 6: Training schedule and checkpoint initialization. Batch size is reported per GPU, with gradient accumulation over eight steps.

Stage	Attention/data	Base initialization	Updated parameters	Steps	LR	Frames
0	BI / real-domain	WorldPlay-BI	Full backbone	2K	5×10^{-7}	97
1	BI / static-camera	Stage-0 BI backbone	Spatial-Pathway LoRA only	40K	5×10^{-6}	97
2	AR / dynamic-camera	WorldPlay-AR	Transferred Spatial-Pathway LoRA only	60K	5×10^{-6}	97

A TRAINING AND CHECKPOINT DETAILS

Trajectory control and camera navigation rely on the same spatial-control pathway identified in §3.5. Naïvely training trajectory control therefore risks *catastrophic interference* with the base model's camera capabilities.

Checkpoint initialization and three-stage pipeline. WorldPlay provides distinct bidirectional (BI) and autoregressive (AR) checkpoints. Stage 0 initializes from the 8B HunyuanVideo-1.5-based WorldPlay-BI checkpoint for real-domain adaptation, and Stage 1 retains the resulting BI backbone for static-camera trajectory learning. Stage 2 loads the corresponding WorldPlay-AR base weights for dynamic-camera autoregressive training and transfers the Spatial-Pathway LoRA learned in Stage 1. The base checkpoint therefore matches the attention regime at each stage, while the trajectory adapter remains continuous from Stage 1 to Stage 2. We set the LoRA rank to 32 and otherwise follow the training configuration of WorldPlay. All experiments are conducted on 8 NVIDIA H200 GPUs.

Stage 0 adapts WorldPlay-BI to the real-video domain through full-parameter fine-tuning at a low learning rate. No trajectory conditioning is used (`trajectory_rate`$=0$), so this stage does not introduce camera-trajectory interference. **Stage 1** freezes the Stage-0-adapted BI backbone and trains Spatial-Pathway LoRA on static-camera trajectory data. Static cameras remove the screen-space entanglement of Eq. 4, yielding a clean trajectory $\rightarrow$ object-motion mapping. **Stage 2** loads the WorldPlay-AR base checkpoint, keeps its backbone frozen, attaches the Spatial-Pathway LoRA learned in Stage 1, and continues optimizing that adapter on dynamic-camera data with AR attention. Stages 1 and 2 use a per-GPU batch size of one and gradient accumulation over eight steps. The AR-native initialization isolates the adapter transfer from changes in the backbone attention regime.

B SCALABLE DATA CURATION PIPELINE

No existing world-model dataset provides the structured supervision our method requires: each training sample must contain a video clip, camera intrinsics and extrinsics, a per-frame binary mask identifying the moving subject, and a multi-point trajectory describing its motion. We build an automatic pipeline that extracts these *(video, camera, mask, trajectory)* tuples from unlabeled video at scale, using only off-the-shelf vision models and physical-plausibility filtering. Figure 5 illustrates the end-to-end flow.

Camera estimation. For videos lacking camera annotations (e.g., WISA-80K), we run ViPE to recover per-frame intrinsics and $\mathrm{SE}(3)$ poses at metric scale. For SpatialVID-HQ, camera parameters are provided as normalized intrinsics and world-to-camera quaternion–translation pairs; we convert these to pixel-space intrinsics and world-to-camera 4×4 matrices, then Slerp-interpolate the sparse annotation frames (sampled at $\lfloor \mathrm{fps}/5 \rfloor$ intervals) to the full frame rate.

Subject discovery. The central challenge is identifying *which* object to track—videos may contain dozens of moving entities, and only a subset exhibit the kind of coherent, spatially significant motion suitable for trajectory training. We employ two complementary strategies depending on the data source:

- **VLM-guided discovery** (WISA-80K). We sample 5 frames uniformly from each clip and query a vision-language model (Qwen2.5-VL-8B) with a structured prompt asking it to identify the most salient moving subject and describe its appearance. The VLM response is parsed into a text query, which is passed to GroundingDINO to produce a bounding box localized in the video. This approach requires no category-specific priors and naturally adapts to open-vocabulary scenes.



Figure 5: Automatic data curation pipeline. Given unlabeled video, we extract camera parameters, discover the salient moving subject, track it with SAM2 to obtain per-frame masks, and run CoTracker to produce multi-point trajectories. Physical-plausibility filters remove degenerate samples. The pipeline adapts to two data sources with complementary strengths: WISA-80K contributes diverse real-world scenes via a fully automatic VLM-guided discovery; SpatialVID-HQ provides metric camera annotations that we combine with a novel tracklet-based subject selection.

- **Multi-frame tracklet matching** (SpatialVID-HQ). SpatialVID-HQ provides binary dynamic-region masks (`dyn_masks`) that mark *all* moving pixels per frame, but do not distinguish individual objects. In crowded scenes a single frame may contain 30–300 connected components of varying size; naïvely selecting the largest component yields multi-person merged blobs (area ratio > 0.3) rather than individual entities. We instead perform cross-frame association: on each annotated frame, we extract connected components filtered to 0.1%–30% of image area, then greedily match components across frames by centroid proximity (threshold: 15% of image diagonal). The resulting *tracklets* capture per-object temporal persistence; we score each tracklet by score $= n_{\text{frames}} \times s_{\text{area}} \times s_{\text{coherence}}$, where s_{area} peaks for objects occupying 0.5%–15% of the frame and $s_{\text{coherence}}$ penalizes erratic centroid jumps. The top-scoring tracklet yields the target frame, bounding box, centroid, and a per-pixel component mask for SAM2 initialization.

Tracking and trajectory extraction. Given the discovered subject (bounding box and, when available, centroid and component mask), we initialize SAM2 video segmentation with a compound prompt: per-pixel mask logits provide the strongest initialization signal, the centroid serves as a positive point to anchor identity, and the bounding box acts as a spatial fallback. This triple-prompt strategy substantially reduces identity switches in crowded scenes compared to box-only prompting. SAM2 propagates bidirectionally from the prompt frame, producing per-frame binary masks.

We then identify the optimal 97-frame window by sliding over the SAM2 output and selecting the interval with maximum mask coverage ($\geq 30\%$ of frames must contain a valid subject mask). Within this window, we verify that the subject exhibits meaningful displacement: centroid net displacement must exceed a minimum threshold, filtering out near-stationary objects whose "motion" is merely camera-induced parallax.

Finally, CoTracker3 tracks 20 query points seeded from the subject mask (one centroid plus 19 uniformly sampled interior points), producing dense multi-point trajectories over 97 frames. The

Table 7: Training data statistics. "Camera source" indicates whether camera parameters are estimated by our pipeline or provided by the dataset. "Subject method" indicates how the target object is identified.

Dataset	Raw	After curation	Camera source	Subject method	Traj. points
WISA-80K	80K	8.5K	ViPE (estimated)	VLM + GroundingDINO	20
SpatialVID-HQ	100K	24K	Provided (metric)	Tracklet matching	20
OmniWorld	—	2K	Provided	Provided	—
Total	—	**∼34.5K**	—	—	—

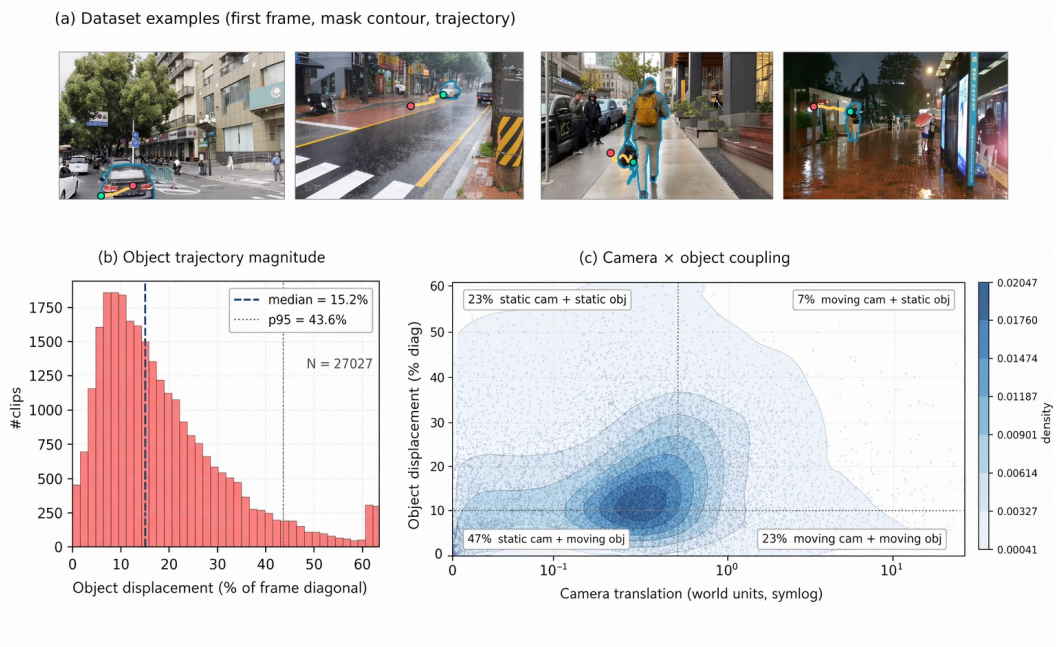


Figure 6: **Curated training set statistics** (N=27,027 clips after filtering). **(a)** Representative samples with the first frame, the SAM2 mask contour of the selected subject (blue), and the multi-point trajectory overlay (start in green, end in red, path in yellow). Subjects range from vehicles and pedestrians to pushed or carried objects under diverse weather and lighting. **(b)** Distribution of object displacement magnitude, measured as the net 2D displacement of the subject centroid across the 97-frame window, normalized by the frame diagonal. The distribution is right-skewed with median 15.2% and p_{95}=43.6%, covering small to large object motions. **(c)** Joint distribution of camera translation (world units, symlog axis) and object displacement (% diagonal). Using thresholds of 0.5 world-units for camera and 10% diagonal for object, four regimes partition the dataset: 47% static-cam / moving-obj (purely object-centric), 23% static-cam / static-obj, 7% moving-cam / static-obj (pure ego-motion), and 23% moving-cam / moving-obj, the WorldCraft-specific regime that demands composable control and is absent from most existing trajectory datasets.

center-point trajectory serves as the primary training signal; the remaining 19 tracks provide auxiliary supervision for spatial extent.

Training data summary. Table 7 summarizes the curated datasets. All videos are standardized to 30 fps and 97 frames (≈3.2 s). Camera parameters are converted to a unified format of per-frame 3×3 intrinsic and 4×4 world-to-camera matrices. Video latents are pre-cached through the HunyuanVideo VAE encoder to accelerate training. Figure 6 reports sample-level statistics of the 27,027 filtered clips: object-displacement magnitude and the joint distribution of camera and object motion.

C SPATIAL-PATHWAY LORA: MOTIVATION, DESIGN, AND ABLATIONS

Shared spatial-control pathway. Camera navigation and object trajectories act at different spatial scales but impose the same type of requirement on the video DiT: an external motion signal must change where visual content is rendered. Camera actions produce global scene transformations, whereas object trajectories produce local transformations of selected instances. In WorldPlay, camera actions are processed by the action encoder `action_in` and Projective Positional Encoding projections `prope_proj`. We therefore test whether trajectory control can reuse this existing spatial-control pathway rather than introduce broad changes to attention routing or channel mixing.

Evidence from full-parameter adaptation. We first fine-tune the full backbone for trajectory control and measure the relative change of each parameter group,

$$\Delta_{\text{rel}}^{(l)} = \frac{\|\mathbf{W}_{\text{ft}}^{(l)} - \mathbf{W}_{\text{base}}^{(l)}\|_F}{\|\mathbf{W}_{\text{base}}^{(l)}\|_F}. \tag{6}$$

The optimizer concentrates its largest changes on `action_in` and `prope_proj`: these modules change by 10-25$\times$ more than attention Q/K/V and MLP layers. Table 9 reports the highest-change parameters and shows that no attention Q/K/V, attention-output projection, or MLP parameter appears among the top 30. This pattern indicates that adapting the mapping from spatial conditioning to feature positions is more important than relearning general semantic routing.

Spatial-Pathway LoRA. Based on this observation, we restrict LoRA (Hu et al., 2022) updates to the spatial-control pathway:

$$\mathbf{W}'^{(l)} = \mathbf{W}^{(l)} + \mathbf{B}^{(l)}\mathbf{A}^{(l)}\frac{\alpha}{r}, \qquad l \in \{\texttt{action_in}, \texttt{prope_proj}^{(1..L)}\}, \tag{7}$$

while freezing attention, MLP, and all remaining backbone parameters. This produces a $\sim$50M-parameter adapter, approximately 0.6% of the 8B backbone. The frozen base weights retain the pretrained camera response, while the low-rank update learns how trajectory conditions perturb the same spatial pathway.

Layer selection and training curriculum. Table 8 evaluates the two design decisions that follow from this formulation. In Table 8(a), full fine-tuning obtains the lowest trajectory error but substantially degrades camera fidelity. Conventional LoRA on Q/K/V and MLP layers, and wider variants that additionally adapt V and MLP layers, use more trainable parameters without improving the TE-RPE trade-off. Spatial-Pathway LoRA provides the best balance between trajectory accuracy and inherited camera control. Table 8(b) then holds the adapter fixed and compares training schedules. Static-BI $\rightarrow$ Dynamic-AR best preserves both capabilities: static bidirectional training first establishes trajectory following, and dynamic autoregressive training transfers it to camera-controlled long-horizon generation.

Table 8: **Adaptation and training ablation.** (a) Spatial-Pathway LoRA, provides the best TE-RPE trade-off with the fewest trainable parameters. (b) The Static-BI $\rightarrow$ Dynamic-AR curriculum gives the best joint preservation of trajectory control and camera fidelity.

Configuration	#Params	TE ↓	RPE_{rot} ↓
(a) Layer selection, trained with Static-BI → Dynamic-AR			
Spatial-Pathway LoRA (ProPE + action)	$\sim$50M	38.90	**0.131**
Spatial pathway + V + MLP (blocks 28-42)	$\sim$120M	46.60	0.136
Spatial pathway + V + MLP (all blocks)	$\sim$180M	47.65	0.169
Q/K/V + MLP (conventional LoRA)	$\sim$200M	49.43	0.139
Full fine-tune	8B	**37.20**	0.237
(b) Training strategy, using Spatial-Pathway LoRA			
Dynamic data, AR attention	$\sim$50M	46.63	0.170
Dynamic BI $\rightarrow$ Dynamic AR	$\sim$50M	**38.25**	0.177
Static BI $\rightarrow$ Dynamic AR	$\sim$50M	38.90	**0.131**

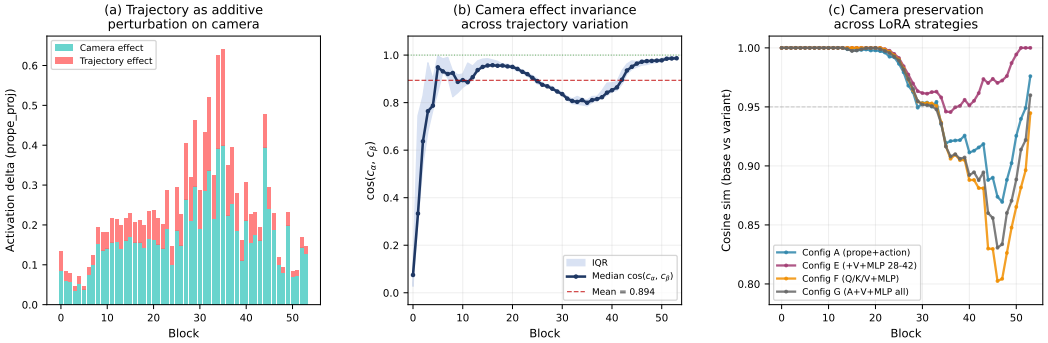


Figure 7: Activation-level analysis of camera - trajectory interaction. **(a)** Trajectory effect is a small additive perturbation on top of the camera signal across all blocks (stacked bars on `prope_proj`). **(b)** Counterfactual camera invariance: $\cos(\mathbf{c}_\alpha, \mathbf{c}_\beta)$ per block, where $\mathbf{c}$ is the camera effect vector measured under two different trajectories. Mean = 0.89, confirming that trajectory variation does not alter the camera effect direction. **(c)** Layer selection ablation on camera preservation: per-block cosine similarity averaged across all hooked layer types (`img_mod`, `prope_proj`, Q/K/V, MLP, `img_attn_proj`) between base model and each LoRA variant under camera-only input. Pathway-selective adaptation (prope+action) preserves camera activations far better than Q/K/V LoRA, which causes significant degradation in mid-to-late blocks.

C.1 Camera-trajectory interaction analysis

Beyond the weight-space and output-level results above, we analyze how trajectory adaptation interacts with the inherited camera representation. Rather than running full video generation, each activation experiment performs a single denoising forward pass and probes the `prope_proj` output across all 54 DiT double-stream blocks. Comparisons use the same denoising timestep and noise realization while varying only the camera and trajectory conditions. This controlled setup isolates their relative activation effects, and the reported trends remain stable across the tested timestep choices.

How large is the trajectory signal? We perform four forward passes with different input conditions (baseline, camera-only, trajectory-only, combined) and decompose the per-layer activation delta. Figure 7(a) shows the trajectory signal as a thin additive layer on top of the camera signal across all 54 blocks. The stacked-bar structure shows that LoRA preserves the dominant camera representation and adds a smaller trajectory-specific perturbation (mean trajectory-to-camera energy ratio $= 0.42$).

Is the camera effect direction preserved? To directly test whether camera control is preserved under trajectory conditioning, we employ a 2×2 counterfactual design: two camera poses (A, B) crossed with two trajectories (α, β). For each layer, the camera effect vector is $\mathbf{c}_\alpha = h(B, \alpha) - h(A, \alpha)$. Camera invariance is measured by $\cos(\mathbf{c}_\alpha, \mathbf{c}_\beta)$. A value near 1 indicates that the direction of the camera effect is unchanged by trajectory variation. Critically, this probe is evaluated on the `prope_proj` output, which encodes continuous camera pose (viewmat) and is the layer that shares parameters with the trajectory adapter. Corruption of the continuous camera pathway would rotate the camera-effect direction. A small but misaligned perturbation would preserve its magnitude while reducing directional agreement. Figure 7(b) shows that from block 5 onward, $\cos(\mathbf{c}_\alpha, \mathbf{c}_\beta)$ is consistently above 0.85, with a cross-block mean of 0.89. This demonstrates that the camera effect direction is highly stable regardless of trajectory input, confirming that pathway-selective LoRA achieves an asymmetric decoupling: camera control is preserved while trajectory control is added.

Do camera and trajectory share a feature subspace? A deeper question is *why* trajectory control can be added without destructively interfering with the original camera-control ability. We analyze token-level activation updates induced by camera control, $\mathbf{u} = \mathbf{h}_{\text{cam}} - \mathbf{h}_{\text{base}}$, and trajectory control, $\mathbf{v} = \mathbf{h}_{\text{cam+traj}} - \mathbf{h}_{\text{cam}}$. Using PCA and cosine-based subspace overlap from principal angles, Figure 8 shows that the two updates are most aligned in the middle layers, where spatial layout and geometric control are primarily represented. Early and late layers show lower overlap, indicating that the

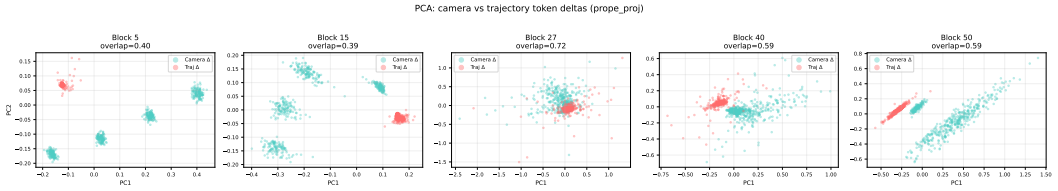


Figure 8: **Shared control subspace.** 2D PCA projection of token-level camera-control updates $\mathbf{u}$ (blue) and trajectory-control updates $\mathbf{v}$ (red) at the peak block. The two distributions are aligned along the same principal directions rather than forming orthogonal subspaces, indicating that trajectory control is injected within the camera-compatible spatial-control subspace.

two signals remain sufficiently distinguishable for low-level visual encoding and final rendering. Together with the preserved camera accuracy in Table 8, this supports our claim that WorldCraft adds object-level control by reusing the existing spatial pathway while avoiding destructive interference with camera control.

Table 9: Top-30 parameters ranked by relative weight change $\Delta_{\text{rel}} = \|\mathbf{W}_{\text{ft}} - \mathbf{W}_{\text{base}}\|_F / \|\mathbf{W}_{\text{base}}\|_F$ after full-parameter trajectory fine-tuning of WorldPlay (8B). The ranking is dominated by `action_in` (ranks 1-2), `prope_proj` (22 of the top 30 rows), and the `final_layer` adaLN modulation, confirming that the optimizer concentrates updates on the spatial-control pathway. No attention Q/K/V, attention-output projection, or MLP parameter appears in the top 30.

Rank	Δ_{rel}	$\|\Delta\mathbf{W}\|_F$	$\|\mathbf{W}_{\text{base}}\|_F$	Params	Category	Parameter
1	0.9619	0.2835	0.2947	4,194,304	action	`action_in.mlp.W`
2	0.1725	0.0037	0.0213	2,048	action	`action_in.mlp.b`
3	0.1371	0.1214	0.8860	4,194,304	prope	`blk53.prope.W`
4	0.0996	1.7929	18.0008	8,388,608	other	`final.adaLN.W`
5	0.0991	0.0017	0.0171	2,048	prope	`blk53.prope.b`
6	0.0945	0.1342	1.4196	4,194,304	prope	`blk52.prope.W`
7	0.0903	0.0018	0.0203	2,048	prope	`blk52.prope.b`
8	0.0883	0.0018	0.0206	2,048	prope	`blk51.prope.b`
9	0.0827	0.0018	0.0214	2,048	prope	`blk50.prope.b`
10	0.0824	0.1103	1.3388	4,194,304	prope	`blk3.prope.W`
11	0.0820	0.0981	1.1956	4,194,304	prope	`blk2.prope.W`
12	0.0768	0.1212	1.5794	4,194,304	prope	`blk51.prope.W`
13	0.0765	0.0018	0.0240	2,048	prope	`blk32.prope.b`
14	0.0759	0.1171	1.5435	4,194,304	prope	`blk5.prope.W`
15	0.0758	0.0018	0.0240	2,048	prope	`blk30.prope.b`
16	0.0755	0.1239	1.6413	4,194,304	prope	`blk6.prope.W`
17	0.0753	0.0019	0.0247	2,048	prope	`blk29.prope.b`
18	0.0753	0.0018	0.0240	2,048	prope	`blk31.prope.b`
19	0.0747	0.0020	0.0268	2,048	prope	`blk26.prope.b`
20	0.0745	0.1182	1.5865	4,194,304	prope	`blk4.prope.W`
21	0.0739	0.0019	0.0251	2,048	prope	`blk28.prope.b`
22	0.0735	0.1269	1.7267	4,194,304	prope	`blk8.prope.W`
23	0.0730	0.0020	0.0273	2,048	prope	`blk25.prope.b`
24	0.0730	0.1247	1.7075	4,194,304	prope	`blk7.prope.W`
25	0.0725	0.0021	0.0286	2,048	prope	`blk24.prope.b`
26	0.0723	0.0019	0.0266	2,048	prope	`blk27.prope.b`
27	0.0723	0.1304	1.8030	4,194,304	prope	`blk9.prope.W`
28	0.0713	0.0021	0.0298	2,048	prope	`blk23.prope.b`
29	0.0697	0.0017	0.0247	2,048	prope	`blk34.prope.b`
30	0.0697	0.1138	1.6325	4,194,304	prope	`blk50.prope.W`

D SUPPLEMENTARY EVALUATIONS

Protocol. The following tables report each baseline on the control outputs defined by its interface. "–" denotes an unavailable output. Trajectory methods are evaluated for object following, camera-

Table 10: Composable camera-object control at 61 frames, averaged over 45 held-out clips. "-" denotes that the method has no interface for the corresponding output. Joint success requires both control axes in the same rollout.

Type	Method	TE↓	DINO↑	PSNR↑	SSIM↑	LPIPS↓	RPE_{rot}↓	RPE_{trans}↓	RPE_{cam}↓	Joint
Traj.	DragAnything	39.76	0.7482	-	-	-	-	-	-	-
Traj.	Wan-Move	35.42	0.7516	-	-	-	-	-	-	-
Cam.	GameCraft	-	-	13.15	0.3180	0.5158	0.173	0.0155	0.0165	-
Cam.	WorldPlay	-	-	14.37	0.3651	0.4370	0.133	0.0159	0.0179	-
Joint	**WorldCraft**	**32.46**	**0.8799**	**14.46**	**0.3683**	**0.4306**	**0.114**	**0.0146**	**0.0163**	✓

Table 11: Long-horizon composable control at 253 frames. Fixed-window trajectory baselines do not define a streaming joint rollout and are therefore omitted.

Type	Method	TE↓	DINO↑	PSNR↑	SSIM↑	LPIPS↓	RPE_{rot}↓	RPE_{trans}↓	RPE_{cam}↓	Joint
Cam.	GameCraft	-	-	12.03	0.2964	0.6518	0.196	0.0213	0.0226	-
Cam.	WorldPlay	-	-	12.94	0.3385	0.5521	0.142	0.0229	0.0210	-
Joint	**WorldCraft**	**43.25**	**0.8028**	**13.23**	**0.3458**	**0.5378**	**0.134**	**0.0194**	**0.0186**	✓

controlled world models for navigation and visual reconstruction, and WorldCraft for both within the same rollout.

D.1 Composable camera-object control

The Composable set contains 45 held-out clips, evenly divided among small, medium, and large camera rotations. Each example specifies both a camera trajectory and a selected-object trajectory. Table 10 reports the 61-frame setting. The specialist baselines remain competitive on the axis they support, but only WorldCraft executes both requested actions in one rollout.

Table 11 extends the same joint protocol to 253 frames (≈ 10.5 seconds). Fixed-window trajectory baselines do not define a streaming camera-plus-object rollout at this horizon, whereas WorldCraft continues to expose both action channels. Relative to the camera-only WorldPlay baseline, WorldCraft retains competitive or improved camera metrics while additionally executing the object trajectory.

D.2 Held-out DROID robot manipulation

Protocol. We additionally evaluate transfer to robot manipulation using the DROID subset of OmniWorld (Khazatsky et al., 2024; Zhou et al., 2026). DROID videos are captured by a wrist-mounted camera on the robot end-effector, so camera and manipulated-object motion are coupled by construction. We annotate 2D trajectories for the robot arm and manipulated object with CoTracker, yielding paired object and camera ground truth for 50 videos of 125 frames each. The DROID evaluation videos are excluded from WorldCraft training, and we perform no DROID-specific adaptation.

Results. WorldCraft transfers to a held-out manipulation domain while jointly controlling the wrist-camera motion and selected entities. It lowers trajectory error relative to Wan-Move (37.77 versus 45.39) and improves identity consistency (0.797 versus 0.701). Despite executing the additional object action, it also improves all reported camera-pose and visual-quality metrics over WorldPlay.

Table 12: Held-out DROID robot-manipulation control (50 videos, 125 frames). Each method is evaluated on the outputs defined by its control interface. "-" denotes an unavailable output.

Type	Method	TE↓	DINO↑	RPE_{rot}↓	RPE_{trans}↓	RPE_{cam}↓	PSNR↑	SSIM↑	LPIPS↓
Traj.	Wan-Move	45.39	0.7011	-	-	-	-	-	-
Cam.	WorldPlay	-	-	0.145	0.0225	0.0203	12.95	0.4213	0.4703
Joint	**WorldCraft**	**37.77**	**0.7970**	**0.127**	**0.0194**	**0.0175**	**13.38**	**0.4433**	**0.4334**

The experiment demonstrates transfer to unseen manipulation scenes with coupled camera-object motion. It evaluates kinematic control and does not measure causal contact dynamics.