DUAL-SIM: INTERLOCKED PHYSICAL-GENERATIVE WORLD SIMULATOR FOR ROBOTIC MANIPULATION

Anonymous authors
Paper under double-blind review

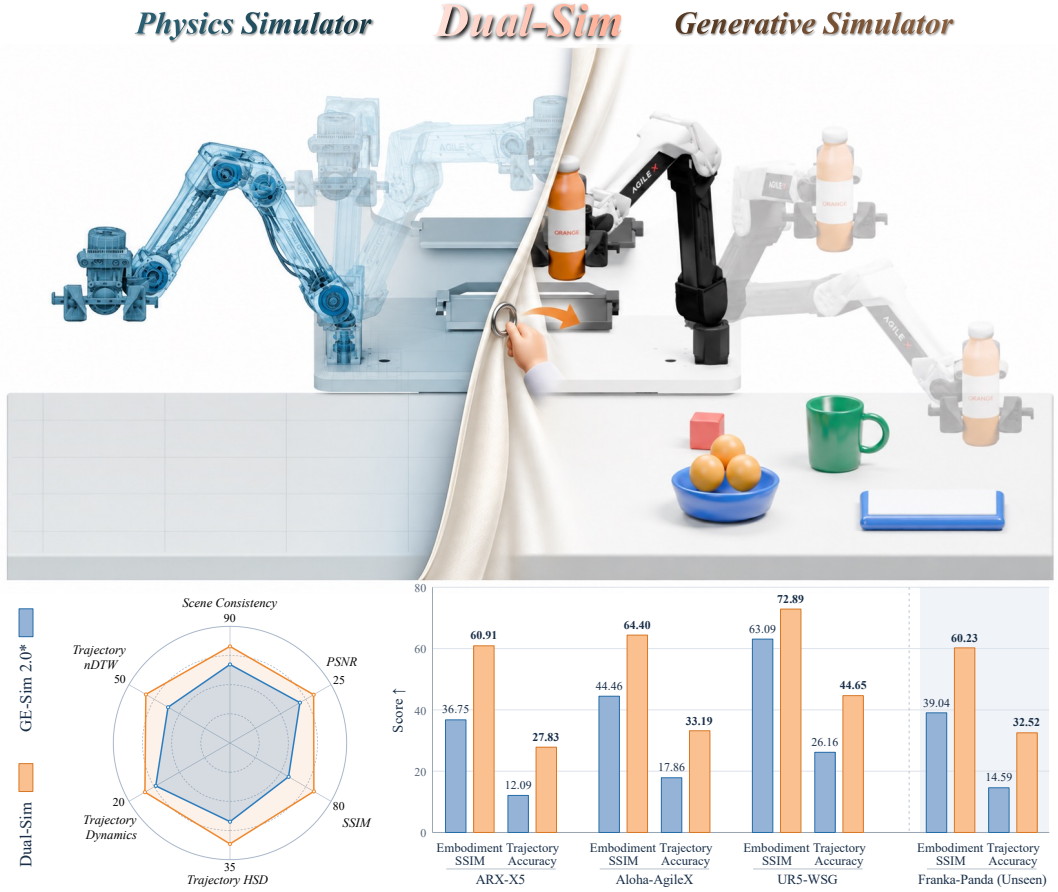

Figure 1: We introduce ***Dual-Sim***, a world simulator that interlocks a physics simulator and a generative simulator, combining kinematic and visual fidelity simultaneously. Dual-Sim achieves superior action following and accurate proprioception states, and can even generalize zero-shot to unseen embodiments, serving as a faithful world simulator for policy evaluation and training.

ABSTRACT

Action-conditioned video world models, *i.e.*, world simulators, produce the future video of a robot executing given actions, offering a photorealistic environment for robotic manipulation. However, three limitations remain in current literature. (1) The generated robot motion gradually drifts from the given actions as the horizon grows. (2) Every new embodiment requires a separately fine-tuned model. (3) The proprioception state that a policy needs for training and evaluation is missing or inaccurate. In contrast, a physics simulator can compensate for these limitations, since it computes robot motion directly. Therefore, we devise Dual-Sim, a world simulator that interlocks a physics simulator and a generative simulator, combining the kinematic fidelity of the physics simulator with the visual fidelity of the generative simulator. Specifically, Dual-Sim casts world prediction as a conditional transport from the robot kinematics to the video and proprioception state. This transport is learned by a video-state Mixture-of-Transformers, whose video

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.

expert and state expert exchange features along one shared flow and predict the video and the state jointly. We further propose kinematic seeding, which replaces the Gaussian source of the flow with an inversion of the rendered robot kinematics. The generative simulator thus refines the robot and completes the scene rather than imaging the future from scratch. Dual-Sim surpasses existing methods in action following, cross-embodiment generalization, and proprioception state accuracy. Closed-loop experiments further establish it as a faithful world simulator.

1 INTRODUCTION

Robot foundation models, VLA (Kim et al., 2024) and WAMs (Kim et al., 2026; Team et al., 2026b) are rapidly advancing robotic manipulation. However, model training and evaluation remain bound to the physical world, leading to slow, costly, and hard to reproduce. Physics simulators (*e.g.*, Isaac Sim) (Makoviychuk et al., 2021) have long offered a way out, yet the rendered world falls short of the real world, *e.g.*, deformable objects and soft contact simulation, leading to a sim-to-real gap.

Recently, world simulators, *i.e.*, action-conditioned video world models, have emerged as an alternative (Qiu et al., 2026; Gu et al., 2026; Team et al., 2026a; Wang et al., 2026). It takes an initial observation and actions to generate the future video of the robot. Generally, world simulators are trained on large-scale videos and thereby acquire visual knowledge of the world. Thus, the generated videos are realistic in appearance, and there is no need to build assets/scenes by hand. However, current world simulators still fall short in three aspects. (1) *Long-horizon action following.* The generated video needs to ensure that the robot motion follows the given actions, yet existing methods, whose robot motion is learned from data, fail to precisely maintain this consistency, especially over long horizons. (2) *Cross-embodiment adaptation.* Existing methods cannot handle different embodiments with a single model, since embodiments differ in degrees of freedom, kinematics, and geometry, and the action-to-motion mapping learned for one embodiment does not hold for another. Thus, they need to fine-tune separate models for each embodiment, and cannot be applied zero-shot to unseen embodiments. (3) *Proprioception state.* Existing methods produce video alone, whereas a VLA/WAM policy needs the proprioception state as input to roll out inside the simulator. Although recent methods (Qiu et al., 2026) can predict the state, since the robot motion exists only in pixel space, they merely estimate the state from the generated video. Thus, the estimate is usually inaccurate. In fact, these limitations share a single cause. For existing methods, the robot has no explicit representation, and its motion exists only in pixel space and is learned purely from data.

In contrast, a physics simulator models the robot explicitly. It takes the embodiment as a kinematic structure and can directly compute the motion according to the given action. Therefore, the motion follows the actions precisely at every step and would not drift over long horizons. Also, it can apply to any embodiment whose kinematics are specified, with no per-embodiment training. It further directly yields the proprioception state of the robot rather than reading off the pixels.

Based on these insights, we introduce **Dual-Sim**, a world simulator that interlocks a physics simulator and a generative simulator (Fig. 1). Intuitively, the physics simulator governs the robot side. It provides sim-video and sim-state, *i.e.*, a rendered video of the robot and the proprioception state at every frame, which carries robot motion and structure. The generative simulator governs the world side, which renders the scene, providing photorealism, object dynamics, and refining the appearance of the robot. In conclusion, Dual-Sim combines the kinematic fidelity of the physics simulator with the visual fidelity of the generative simulator in a single world simulator.

More technically, Dual-Sim reformulates world prediction from a generation into an editing problem. Existing methods transport Gaussian noise to the future video. Dual-Sim instead treats the sim-video and the sim-state as a strong kinematic prior and directly transports them to the results. The generative simulator is designed via a video-state Mixture-of-Transformers, in which a video expert produces the video and a state expert produces the proprioception state. The two experts keep separate weights, exchange features in every layer, and share a single flow, so that the video and the state are predicted jointly. Both experts start from this prior rather than from noise, conditioned on the observation history, the state history, and the actions. We propose kinematic seeding to build the transportation sources. For the video expert, it inverts the sim-video inside the robot mask with a small amount of noise and fills the outside with pure noise. The small amount of noise leaves room to refine the texture and appearance of the robot, and to adjust the motion where contact changes it.

For the state expert, it inverts the sim-state to the same noise level. The flow therefore starts with the priors and generates what the physics simulator leaves out, *i.e.*, objects, background, and their interaction with the robot.

Experimentally, Dual-Sim achieves higher visual quality and action consistency than baselines, *e.g.*, trajectory nDTW is 17% higher. A single Dual-Sim can produce multiple embodiments, and it even generalizes to unseen embodiments, with a trajectory accuracy $2.2\times$ that of the baseline. The proprioception state of Dual-Sim is more accurate, reducing per-joint MAE by 8.5% relative to baseline. As a policy evaluator, Dual-Sim yields policy success rates that correlate with those in the real world at a Pearson coefficient of 0.92 across five tasks and three VLA policies. As a source of training data, Dual-Sim raises the success rate of π_0 from 37.7% to 79.5% in the real world.

Our main contributions are summarized as follows. (1) We introduce Dual-Sim, a world simulator that interlocks a physics simulator with a generative simulator through the sim-video and the sim-state, and casts world prediction as a conditional transport from the simulated kinematics to the results. (2) We design a video-state Mixture-of-Transformers framework, in which a video expert and a state expert produce a joint velocity field via shared flow, so that the two predictions are made jointly and remain consistent. (3) We devise a kinematic seeding mechanism, which replaces the Gaussian source of the flow with a mask-gated inversion of the sim-video and the sim-state. The generative simulator learns to refine the robot and complete the scene. (4) Extensive experiments show that Dual-Sim surpasses existing methods in action following, cross-embodiment generalization, and proprioception state prediction. Close-loop VLA experiments further confirm its value.

2 Related Works

Physics Simulators have been the primary platform for robot learning. MuJoCo (Todorov et al., 2012), PyBullet (Coumans & Bai, 2016), and SAPIEN (Xiang et al., 2020) integrate rigid-body dynamics with contact. Isaac Gym (Makoviychuk et al., 2021), ManiSkill3 (Tao et al., 2025), and Genesis (Authors, 2024) add high-fidelity rendering and deformable-body simulation. Also, they are manipulation benchmarks built on simulators, *e.g.*, LIBERO (Liu et al., 2023) and RoboTwin (Chen et al., 2026a). However, in simulation, both the rendered appearance and the contact dynamics still deviate from the real world, leading to the sim-to-real gap. Dual-Sim instead restricts the physics simulator to its strength (robot kinematics) and delegates scene generation and contact dynamics to the generative simulator, so that the two parts complement each other.

Action-Conditioned World Models predict the future observation of a robot according to the given actions. Existing methods mainly differ in how the actions are injected into the generation model. (1) *Numerical vectors.* Methods encode end-effector poses or joint values as vectors and inject them through cross-attention or adaptive normalization, as in IRASim (Zhu et al., 2025) and Ctrl-World (Guo et al., 2026). (2) *Semantics.* DreamGen (Jang et al., 2025) conditions on text instructions and ORV (Yang et al., 2026) on 4D occupancy. (3) *2D visual prompts.* EnerVerse-AC (Jiang et al., 2025) and GE-Sim 2.0 (Qiu et al., 2026) projects end-effector poses onto the image, VAP (Wang et al., 2025) renders skeletons, BridgeV2W (Chen et al., 2026b) and Kinema4D (Xu et al., 2026) render robot masks and pointmaps, and MVA (Alzayer et al., 2026) and GeniWorld (Gu et al., 2026) condition on visual actions. Dual-Sim instead conditions on a physics simulator, which executes the actions and provides the simulated robot as a strong prior. World prediction is thereby reformulated as a conditional transport from the simulated robot to the video and state.

3 Method

3.1 Dual-Sim Framework

Problem Formulation. World simulators predict how the robot evolve under a sequence of actions. We set the current step to 0 and the future spans steps from 1 to T. The input is a context $c = (O_{\leq 0}, q_{\leq 0}, A)$, where $O_{\leq 0}$ is the observation history ending at the current observation O_0, $q_{\leq 0}$ is the proprioception state history ending at the current state q_0, and $A = [a_1, \ldots, a_T]$ is the action sequence for the next T steps. Given c, our goal is to build a world simulator $\mathcal{W}$ that jointly predicts the future video $V_{1:T}$ and the future proprioception state $q_{1:T}$,

$$(\widehat{V},\, \widehat{q}) = \mathcal{W}(O_{\leq 0},\, q_{\leq 0},\, A)\,, \tag{1}$$

where $\widehat{V}$ and $\widehat{q}$ denote the predictions of $V_{1:T}$ and $q_{1:T}$, which are required by VLA or WAM policies, which can therefore be trained and evaluated inside the world simulator as it would be on a real robot.

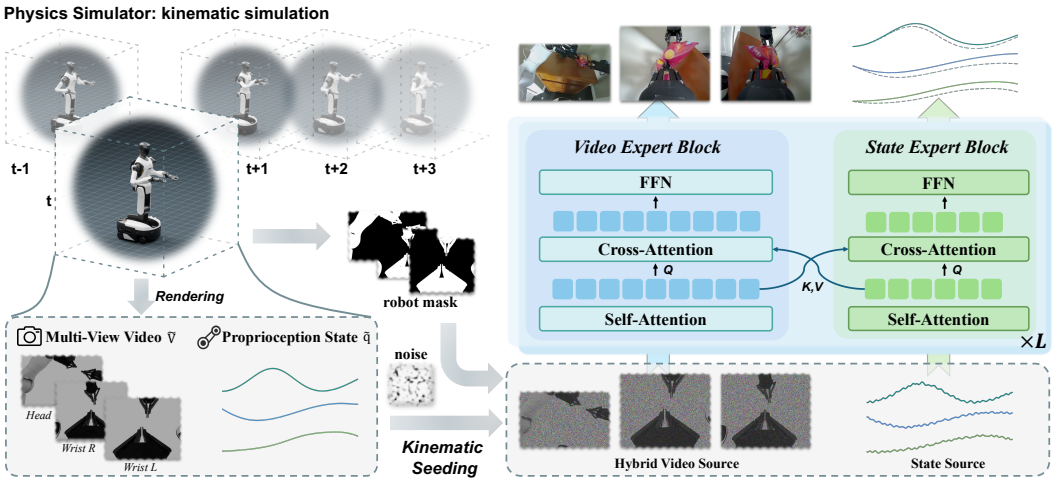

Figure 2: **Overview of Dual-Sim. Left:** The physics simulator executes actions and renders robot motion into the sim-video, sim-state, and robot mask. **Right:** The generative simulator transports the joint source to result videos and states via shared flow, conditioned on the observation history.

Framework Overview. Dual-Sim realizes the world simulator by coupling a physics simulator with a generative simulator, designed to complement each other (Fig. 2). Specifically, the physics simulator models the robot embodiment and executes the action sequence A. It outputs a video of the robot, which we term the sim-video $\widetilde{V}$, including precise robot trajectory and robot structure. It also produces the proprioception state of the robot at every frame, which we term the sim-state $\widetilde{q}$.

We design the generative simulator to map sim-video $\widetilde{V}$ and sim-state $\widetilde{q}$ to the future video $V_{1:T}$ and the future proprioception state $q_{1:T}$, conditioned on the context c. We formulate this mapping as a conditional flow whose source is built from $\widetilde{V}, \widetilde{q}$ (Sec. 3.2). We parameterize the flow with a video-state Mixture-of-Transformers, which predicts the video and the state jointly (Sec. 3.3). The video expert and state expert keep separate weights, exchange latent features, and share a single flow, keeping predicted video and state consistent. We propose a kinematic seeding mechanism to plant $\widetilde{V}, \widetilde{q}$ as the source of the generative flow. The flow preserves the given motion, completes the surrounding scene, and corrects robot appearance and contact-dependent motion. Thus, the predicted video and state inherit the accuracy of the physics simulator.

Physics Simulator. We instantiate physics simulator with Isaac Sim (Makoviychuk et al., 2021), which simulates the required embodiment. Given the context c and action sequence A, it executes each action on this embodiment from the current state q_0 and simulates results. It then renders the robot into three camera views, *i.e.*, a head view and two wrist views, with a blank background. This yields the sim-video $\widetilde{V}$, together with a binary robot mask M that marks the robot in every frame, as shown in Appendix A.3. Note that the mask is obtained easily via the simulator. Also, the sim-state $\widetilde{q}$ is obtained via kinematics computations. Thus, the sim-video carries the motion, structure, and appearance of the robot, and the sim-state carries its joint configuration.

Generative Simulator. Given the sim-video $\widetilde{V}$, the sim-state $\widetilde{q}$, and the robot mask M from the physics simulator, it predicts the future video $\widehat{V}$ and the future proprioception state $\widehat{q}$:

$$(\widehat{V}, \widehat{q}) = \mathcal{G}_\theta(\widetilde{V}, \widetilde{q}, M;\, c), \tag{2}$$

where θ denotes the learnable parameters. Specifically, we propose to cast world prediction $\mathcal{G}_\theta$ as a conditional transport (Sec. 3.2). We realize this transport with a video-state Mixture-of-Transformers (Sec. 3.3), where the video expert generates $\widehat{V}$, and the state expert generates $\widehat{q}$. Kinematic seeding plants the sim-video inside the robot mask as its starting point with a small amount of noise and leaves the rest to pure noise. We plant the sim-state as its starting point with the same noise level. The two experts maintain separate weights but exchange features throughout the network:

$$\widehat{V} = \mathcal{G}_\theta^V(\widetilde{V}, M;\, c,\, H_S), \qquad \widehat{q} = \mathcal{G}_\theta^S(\widetilde{q};\, c,\, H_V), \tag{3}$$

where H_V and H_S denote the intermediate features of the video expert and the state expert. Both experts take the sim-video and the sim-state as a strong prior and refine them rather than generate

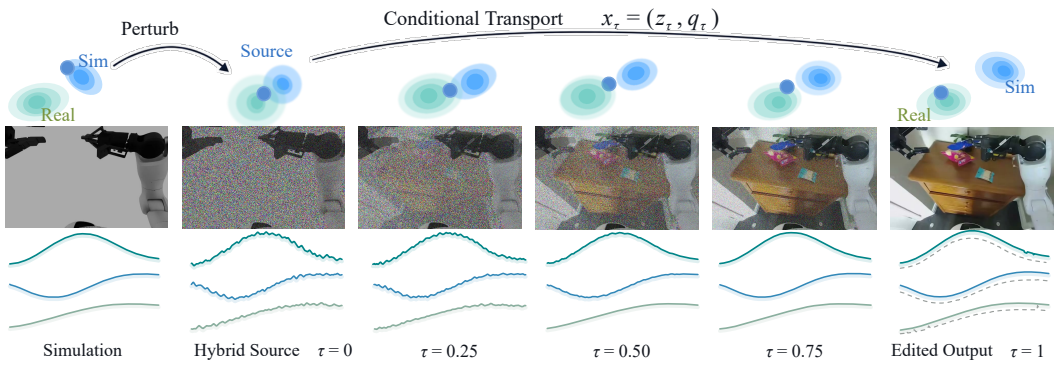

Figure 3: **World prediction as conditional transport.** The hybrid source of the transport is built from the sim-video and sim-state. Inside the robot mask M, the source keeps the rendered robot with a small amount of noise, and outside the mask it starts from noise. As τ runs from 0 to 1, the robot region moves from the rendered robot to its real appearance, the remaining region moves from noise to the surrounding scene, and the state moves from the sim-state to the real state.

from scratch. The video expert keeps the given robot motion, refines the robot appearance, and completes the objects and scenes, guaranteeing photorealism. The state expert keeps the joint configuration and refines it where the interaction alters the motion. Exchanging H_V and H_S ties the two refinements to the same interaction, so that the jointly predicted video and state stay consistent.

3.2 World Prediction via Kinematic Seeding

We now describe how the conditional transport in Eq. 2 is realized. Existing generative simulators transport Gaussian noise to the future video and inject the actions as conditioning. Dual-Sim instead transports a hybrid source built from the sim-video $\widetilde{V}$ and the sim-state $\widetilde{q}$. The transport thus takes $\widetilde{V}, \widetilde{q}$ as strong priors, and generates what the physics simulator leaves out, *i.e.*, objects, background, and their interaction with the robot. We adopt a frozen video encoder $\mathcal{E}$ and the decoder $\mathcal{D}$. The encoder maps the sim-video to $\widetilde{z} = \mathcal{E}(\widetilde{V})$, and the robot mask M is aligned to the grid of $\widetilde{z}$. For simplicity, the video feature and the state are formulated as one joint variable $x = (z, q)$.

Hybrid Source Construction. The source $x_{\text{src}} = (z_{\text{src}}, q_{\text{src}})$ is the joint variable at the start of the transport. We obtain it by inverting $\widetilde{V}$ and $\widetilde{q}$, *i.e.*, by adding noise. For the video, kinematic seeding inverts the sim-video inside the robot mask and fills the outside with pure noise,

$$z_{\text{src}} = M \odot \big(\alpha \widetilde{z} + (1-\alpha)\epsilon_V\big) + (1-M) \odot \epsilon_V, \qquad \epsilon_V \sim \mathcal{N}(0, I). \tag{4}$$

For the state, state editing inverts the sim-state to the same noise level,

$$q_{\text{src}} = \alpha \widetilde{q} + (1-\alpha)\epsilon_S, \qquad \epsilon_S \sim \mathcal{N}(0, I), \tag{5}$$

where ϵ_V and ϵ_S are independent and $\alpha \in (0, 1)$ is close to one, so that the source stays close to the outputs of the physics simulator. We invert rather than copy these outputs for two reasons. First, the inversion leaves room to correct the appearance of the robot. Second, it allows the motion to be adjusted where contact or interaction changes it.

Conditional Transport. As shown in Fig. 3, starting from the source, the joint variable $x_\tau = (z_\tau, q_\tau)$ evolves over the flow time $\tau \in [0, 1]$ under a conditional velocity field $u_\theta = (u_\theta^V, u_\theta^S)$,

$$\frac{\mathrm{d}z_\tau}{\mathrm{d}\tau} = u_\theta^V(x_\tau, \tau; c, M), \qquad \frac{\mathrm{d}q_\tau}{\mathrm{d}\tau} = u_\theta^S(x_\tau, \tau; c, M), \qquad x_{\tau=0} = x_{\text{src}}, \quad \widehat{x} = x_{\tau=1}. \tag{6}$$

The video velocity u_θ^V is produced by the video expert and the state velocity u_θ^S by the state expert (Sec. 3.3). Both read the current video z_τ and the current state q_τ, so the two variables shape each other along the whole transport rather than being predicted in isolation.

Training. We train on collected trajectories, in which every step provides both the context c and the ground-truth future video $V_{1:T}$ and state $q_{1:T}$. Each step thus provides a pair of a source x_{src}, built by Eq. 4 and Eq. 5, and a target $x = (z, q)$ with $z = \mathcal{E}(V_{1:T})$ and $q = q_{1:T}$. Conditional flow matching connects the pair by a straight path with constant velocity:

$$x_\tau^{\text{pair}} = (1-\tau)x_{\text{src}} + \tau x, \qquad u^* = x - x_{\text{src}}, \tag{7}$$

with video and state components $u^{*V} = z - z_{\text{src}}$ and $u^{*S} = q - q_{\text{src}}$. We regress the two velocity components of Eq. 6 onto u^* along this path,

$$\mathcal{L} = \mathbb{E}\Big[\tfrac{1}{2}\big\langle(\delta^V)^2\big\rangle_M + \tfrac{1}{2}\big\langle(\delta^V)^2\big\rangle_{1-M} + \big\langle(\delta^S)^2\big\rangle\Big], \tag{8}$$

where $\delta^V = u^V_\theta(x^{\text{pair}}_\tau, \tau; c, M) - u^{*V}$ and $\delta^S = u^S_\theta(x^{\text{pair}}_\tau, \tau; c, M) - u^{*S}$ are the velocity errors, $\langle\cdot\rangle_M$ and $\langle\cdot\rangle_{1-M}$ average over latent positions inside and outside the robot mask, and $\langle\cdot\rangle$ averages over the T steps. The expectation is over trajectories, the source noise (ϵ_V, ϵ_S), and $\tau \sim \mathcal{U}[0, 1]$. The visual error is averaged inside and outside the mask separately with equal weight, since robot covers a small part of the frame and its supervision would otherwise shrink with its image area.

3.3 Video-State Mixture of Transformers

As shown in Fig. 2, we adopt a Mixture-of-Transformers (MoT) architecture, in which the video expert and the state expert keep separate weights but exchange features in every layer. Both experts are conditioned on the same flow time τ, so that the two velocities are integrated together along one flow. The video expert receives the observation history and the evolving video latent, and the state expert receives the actions, the state history, and the evolving state,

$$H^{(0)}_V = P_V([\mathcal{E}(O_{\le 0}); z_\tau \oplus M]), \qquad H^{(0)}_S = P_S([A; q_{\le 0}; q_\tau]), \tag{9}$$

where P_V embeds video patches, P_S embeds action and state tokens, $[\cdot\,;\cdot]$ denotes concatenation along the sequence, and $\oplus$ denotes concatenation along channels. The robot mask M thus tells the video expert where the source carries the rendered robot.

Feature Exchange. Each expert is a stack of L transformer blocks. In block l, each expert first applies self-attention to its own tokens with its own weights, giving $\bar{H}^{(l)}_V$ and $\bar{H}^{(l)}_S$. The two experts then exchange features through residual cross-attention,

$$H^{(l)}_V = \bar{H}^{(l)}_V + \mathrm{CA}_V\Big(\bar{H}^{(l)}_V, \bar{H}^{(l)}_S\Big), \qquad H^{(l)}_S = \bar{H}^{(l)}_S + \mathrm{CA}_S\Big(\bar{H}^{(l)}_S, \bar{H}^{(l)}_V\Big), \tag{10}$$

where $\mathrm{CA}(H, H')$ attends from the tokens of H to the tokens of H', and each expert finally applies its own feed-forward layer. Through CA_S, the state expert reads the scene context from the video features and corrects the motion where contact alters it. Through CA_V, the video expert reads the corrected motion from the state features and follows it in the visual prediction. The features $H^{(l)}_V$ and $H^{(l)}_S$ are the H_V and H_S exchanged in Eq. 3. After the last block, two linear heads R_V and R_S read the features at the positions of z_τ and q_τ and output their velocities,

$$u^V_\theta = R_V\Big(H^{(L)}_V\Big), \qquad u^S_\theta = R_S\Big(H^{(L)}_S\Big). \tag{11}$$

Only z_τ and q_τ are transported. The history and action tokens are clean conditioning inputs and stay fixed. These velocities are integrated in Eq. 6 at inference and regressed onto u^* in Eq. 8 at training.

4 Experiments

We structure our evaluation around three key questions. (1) Can Dual-Sim generate visually faithful and action-consistent predictions? (2) Can a single Dual-Sim model generalize across robot embodiments, including an embodiment never observed during training? (3) Are Dual-Sim's rollouts accurate and informative to support the evaluation and training of real-world robot policies?

4.1 Video-State Generation Evaluation

Datasets and Baselines. We adopt the RichInteraction subset of AGIBOT WORLD 2026 (Team, 2026). We curate 100 challenging episodes for evaluation. We compare Dual-Sim with Ctrl-World (Guo et al., 2026), EnerVerse-AC (Jiang et al., 2025), DreamDojo (Gao et al., 2026), and GE-Sim 2.0 (Qiu et al., 2026). GE-Sim 2.0 and EnerVerse-AC are pretrained exclusively on large-scale AGIBOT embodiments. For DreamDojo, we use the officially released version post-trained on the AGIBOT embodiment. Further, we also fine-tune GE-Sim 2.0 on the same Rich-Interaction training split as Dual-Sim to better showcase our superiority, which is denoted as *GE-Sim 2.0**.

Evaluation metrics. Following (Yue et al., 2025; Liao et al., 2026; Zhang et al., 2026a), we measure both visual quality and action consistency (Tab. 1), where EWMScore is an overall metric. Appendix A.4 gives more metric definitions.

Table 1: Action-conditioned video generation on AGIBOT WORLD 2026 RichInteraction. We compare Dual-Sim against four existing methods.

Method	EWMScore ↑	Visual Quality			Action Consistency		
		Scene Consistency ↑	PSNR (dB) ↑	SSIM ↑	HSD ↑	Dynamics ↑	nDTW ↑
Ctrl-World	49.54	**88.94**	15.82	56.67	11.99	3.52	14.47
EnerVerse-AC	49.30	84.08	15.49	57.30	15.81	6.91	19.55
DreamDojo	44.99	76.57	14.51	54.97	12.84	8.40	16.93
GE-Sim 2.0	52.48	86.55	17.86	67.25	16.96	12.54	19.88
*GE-Sim 2.0**	59.11	85.10	18.84	67.38	31.74	14.70	38.40
Dual-Sim	**64.67**	87.42	**21.56**	**74.90**	**33.65**	**16.84**	**44.99**

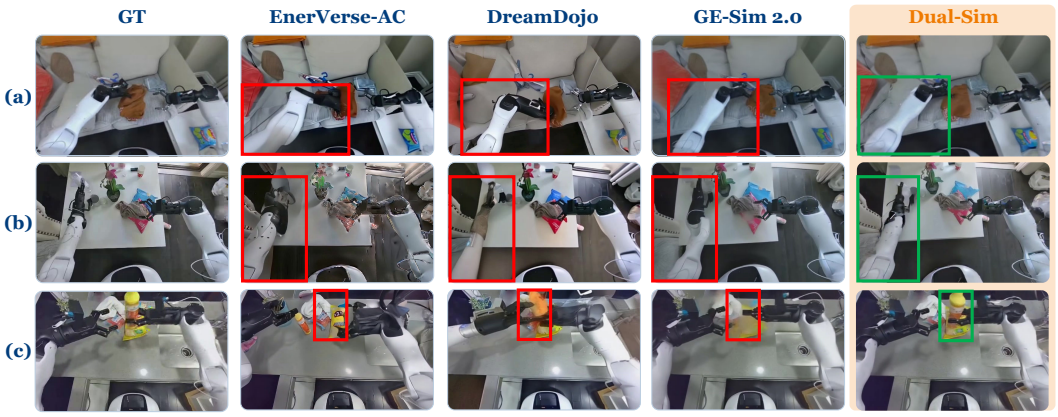

Figure 4: **Qualitative results of video generation.** Dual-Sim demonstrates (a) more accurate robot kinematics, (b) more consistent embodiment appearance, and (c) more plausible object interactions.

Comparisons. As shown in Tab. 1, Dual-Sim achieves the best EWMScore of 64.67, improving over GE-Sim 2.0* by 5.56% and DreamDojo by 19.68%. The substantial gains in PSNR and SSIM indicate that Dual-Sim achieves better scene appearance. The action consistency metrics demonstrate more accurate adherence to the given actions. As shown in Fig. 4(a), the embodiment generated by Dual-Sim remains closely aligned with the GT arm position, while the baselines exhibit noticeable positional deviations. Fig. 4(c), Dual-Sim correctly captures the grasped-object state, whereas the baselines struggle to model the object interaction.

Table 2: **Ablations.** All variants only differ in how the sim-video is incorporated into the generative model.

Method	PSNR ↑	SSIM ↑	Trajectory nDTW ↑
Input Cond.	19.90	70.50	40.13
Feature Cond.	19.95	68.94	38.40
Dual-Sim	**21.56**	**74.90**	**44.99**

Ablation on Kinematic Seeding. We compare three mechanisms for incorporating sim-video. *Input-Level Conditioning* concatenates the sim-video with the model input. *Feature-Level Conditioning* injects sim-video features into intermediate layers. Both treat simulated motion as an auxiliary condition to be interpreted by the generative model. In contrast, Dual-Sim uses the kinematic seeding mechanism to construct a hybrid source to conduct conditional transport. As shown in Tab. 2, this design performs best across all three metrics, indicating that directly anchoring the transport source better preserves the simulated motion prior while retaining flexibility for refinement.

Proprioceptive State Accuracy. We report mean absolute error (MAE), averaged uniformly over all 16 state dimensions and the same 100 evaluation episodes. Dual-Sim reduces the overall state MAE by 8.5% relative to *GE-Sim 2.0**, showing an advantage in long-horizon stability. As shown in Fig. 6, *GE-Sim 2.0** exhibits substantially stronger error accumulation over time, whereas Dual-Sim remains considerably more stable. These results indicate that Dual-Sim performs better in long-horizon action following and mitigates autoregressive drift.

Table 3: **Cross-embodiment evaluation on RoboTwin.** ARX-X5, Aloha-AgileX, and UR5-WSG are used for training, while Franka-Panda is held out entirely and used for zero-shot evaluation.

Method	ARX-X5		Aloha-AgileX		UR5-WSG		Franka-Panda (Unseen)	
	Emb. SSIM ↑	Traj. Acc. ↑	Emb. SSIM ↑	Traj. Acc. ↑	Emb. SSIM ↑	Traj. Acc. ↑	Emb. SSIM ↑	Traj. Acc. ↑
*GE-Sim 2.0**	36.75	12.09	44.46	17.86	63.09	26.16	39.04	14.59
Dual-Sim	60.91	27.83	64.40	33.19	72.89	44.65	**60.23**	**32.52**

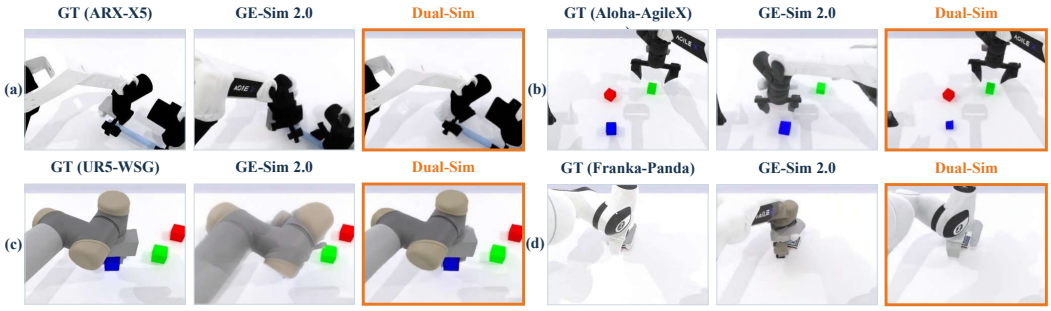

Figure 5: **Qualitative Comparisons of Cross-embodiment Generalization.** Dual-Sim better preserves embodiment-specific structure and action consistency across seen and unseen embodiments.

4.2 Cross-embodiment Evaluation

Protocol and Metrics. We investigate whether Dual-Sim enables a single model to generalize across multiple embodiments. We evaluate on three training embodiments and the held-out Franka-Panda embodiment on 13 RoboTwin tasks. Dual-Sim is jointly trained on the three embodiments and evaluated on the held-out Franka-Panda embodiment without any adaptation. We compare against *GE-Sim 2.0**, which is trained on the same data. Note that all evaluation episodes are unseen. *Embodiment SSIM* measures the visual fidelity, and *Trajectory Accuracy* measures action consistency.

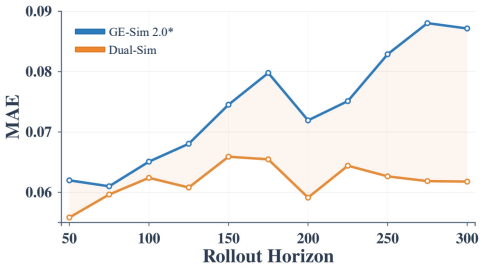

Figure 6: **Proprioception Drift.** Dual-Sim exhibits substantially less error accumulation than the baseline over long horizons.

Comparisons. As shown in Tab. 3, Dual-Sim improves all metrics across all embodiments. Importantly, on the unseen Franka-Panda embodiment, Emb.SSIM increases from 39.04% to 60.23%, and Traj.Acc improves from 14.59% to 32.52%. Baseline confuses the visually similar ARX-X5 and Aloha-AgileX embodiments in Fig. 5(a), produces distorted robot geometry for the UR5-WSG embodiment in Fig. 5(b), and fails to generate results for the unseen Franka-Panda in Fig. 5(c). In contrast, Dual-Sim preserves embodiment-specific structure and generates motion consistent with the input actions across both seen and unseen embodiments.

4.3 Policy Evaluation with Dual-Sim

We assess whether Dual-Sim can serve as a reliable simulator for real-world closed-loop policy evaluation. To this end, we compare policy success rates in Dual-Sim against those measured on the real robot across five tasks. Experiments are conducted on a dual-arm YAM Ultra1 robot, with the setup illustrated in Fig. 7. Detailed experiment settings are provided in Appendix A.8.

Policies and Tasks. We evaluate π_0, $\pi_{0.5}$, and Diffusion Policy (DP) on five tasks: *Place Bread*, *Place Banana*, *Pass Spatula*, *Flip Bread*, and *Open Lid & Place Bread*. Task definitions and examples appear in Appendix A.8.1 and Fig. 15. Both π_0 and $\pi_{0.5}$ are trained across tasks and receive language instructions, and DP is trained separately for each task.

We compare real-world and simulated success rates across 15 task-policy pairs using Pearson correlation, ordinary least-squares regression, and mean absolute error (MAE).

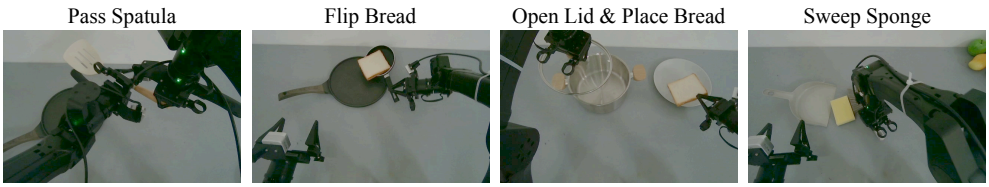

Figure 7: Real world tasks demonstration.

Success-rate Correlation. Fig. 8 shows a positive correlation between real-world and Dual-Sim success rates, with Pearson $r = 0.9168$. Compared with GE-Sim 2, Dual-Sim achieves higher Pearson correlation (0.9168 vs. 0.8122) and lower success-rate MAE (13.3 vs. 20.3 percentage points), as detailed in Appendix A.8.1. These results indicate that Dual-Sim effectively reflects real-world performance trends, supporting its value for closed-loop policy evaluation.

4.4 Improving Policies with World Simulator

We assess whether trajectories generated by Dual-Sim can serve as effective training data for real-world policies. To this end, we post-train π_0 with real and generated trajectories and evaluate it on the real robot across four tasks. The results are shown in Fig. 9

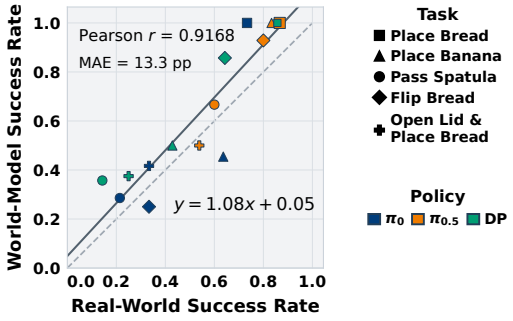

Figure 8: **Policy Evaluation in Dual-Sim.**

Tasks and Setting. We adopt four tasks: *Place Apple*, *Place Pomegranate*, *Place Bun*, and *Sweep Sponge*. *Real only* is π_0 fine-tuned using only real demonstrations, with 20 demonstrations per task. Its success rate is measured by evaluating this policy in the real world.

Then, we train π_0 using the same real demonstrations together with trajectories generated in GE-Sim2 or Dual-Sim, obtaining *Real + GE-Sim2* and *Real + Dual-Sim*, respectively. Each Sim generates 30 trajectories per task. Policies are jointly trained on tasks from the same checkpoint and then evaluated in the real world.

We conduct 15 trials per task and policy, with success judged manually. For each task, we keep the initial scene configuration and object placement as consistent to ensure a fair comparison. Task definitions, training details, and evaluation examples appear in Appendices A.8.3 and A.8.4.

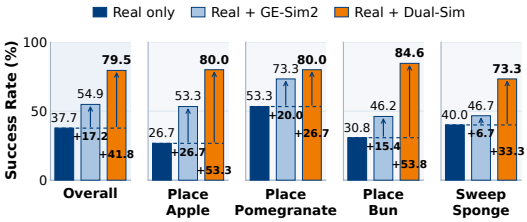

Figure 9: **Policy Training with World Simulator.** Overall denotes the average performance.

Results. π_0 under *Real+Dual-Sim* yields a mean real-world SR of 79.5%, compared with 54.9% for *Real+GE-Sim2* (Fig. 9). *Real+Dual-Sim* achieves higher SR on all four tasks, showing consistent training benefits across fruit placement, container interaction, and tool use. In contrast, the gains from GE-Sim2 are more limited on some tasks, particularly *Sweep Sponge*. On *Place Bun*, which requires placing the bun in the air-fryer basket and closing its drawer, SR reaches 84.6% with Dual-Sim data compared with 46.2% with GE-Sim2 data. These results indicate that Dual-Sim provides more effective trajectories for policy training.

5 Conclusion

We present Dual-Sim, a world simulator that interlocks a physics simulator and a generative simulator, combining kinematic and visual fidelity. Dual-Sim casts world prediction as a conditional transport from the robot kinematics to the results. This transport is learned by a video-state Mixture-of-Transformers, and its source is built by kinematic seeding. Experiments show that Dual-Sim surpasses existing methods in action following, cross-embodiment, and state accuracy. Closed-loop experiments establish it as a faithful policy evaluator and an effective training source.

REFERENCES

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.

Hadi Alzayer, Wenlong Huang, Haonan Chen, Christopher Luey, Lvmin Zhang, Maneesh Agrawala, Gordon Wetzstein, Li Fei-Fei, Yilun Du, Jiajun Wu, et al. Masked visual actions for unified world modeling. *arXiv preprint arXiv:2607.19343*, 2026.

Genesis Authors. Genesis: A universal and generative physics engine for robotics and beyond. *URL https://github. com/Genesis-Embodied-AI/Genesis*, 2, 2024.

Hritik Bansal, Zongyu Lin, Tianyi Xie, Zeshun Zong, Michal Yarom, Yonatan Bitton, Chenfanfu Jiang, Yizhou Sun, Kai-Wei Chang, and Aditya Grover. Videophy: Evaluating physical commonsense for video generation. In *International Conference on Learning Representations*, volume 2025, pp. 102075–102121, 2025.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first international conference on machine learning*, 2024.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=itonej9GIV.

Yixiang Chen, Peiyan Li, Jiabing Yang, Keji He, Xiangnan Wu, Yuan Xu, Kai Wang, Jing Liu, Nianfeng Liu, Yan Huang, et al. Bridgev2w: Bridging video generation models to embodied world models via embodiment masks. *arXiv preprint arXiv:2602.03793*, 2026b.

Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning, 2016.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. *Advances in Neural Information Processing Systems*, 37:91560–91596, 2024.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Naveed Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, Jiannan Xiang, Yuqi Xie, Ruijie Zheng, Dantong Niu, Pooya Jannaty, Jinwei Gu, Jun Zhang, Jitendra Malik, Pieter Abbeel, Ming-Yu Liu, Yuke Zhu, Joel Jang, and Linxi Fan. Dreamdojo: A generalist robot world model from large-scale human videos. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=FuvU7PTyED.

Chenghao Gu, Hanyang Yu, Jingbo Zhang, Haitao Lin, Wenyao Zhang, Jinghe Wang, Hanglei Jin, Shuzhao Xie, Jingyan Jiang, and Zhi Wang. Geniworld: A generalizable interactive world model for robotic manipulation via visual actions. *arXiv preprint arXiv:2608.06332*, 2026.

Yanjiang Guo, Lucy Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 6121–6138, 2026.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

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.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, Loïc Magne, Ajay Mandlekar, Avnish Narayan, You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Yinzhen Xu, Xiaohui Zeng, Kaiyuan Zheng, Ruijie Zheng, Ming-Yu Liu, Luke Zettlemoyer, Dieter Fox, Jan Kautz, Scott Reed, Yuke Zhu, and Linxi Fan. Dreamgen: Unlocking generalization in robot learning through video world models. In *9th Annual Conference on Robot Learning*, 2025. URL https://openreview.net/forum?id=3CnxNqmklv.

Yuxin Jiang, Shengcong Chen, Siyuan Huang, Liliang Chen, Pengfei Zhou, Yue Liao, Xindong He, Chiming Liu, Hongsheng Li, Maoqing Yao, et al. Enerverse-ac: Envisioning embodied environments with action condition. *arXiv preprint arXiv:2505.09723*, 2025.

Bingyi Kang, Yang Yue, Rui Lu, Zhijie Lin, Yang Zhao, Kaixin Wang, Gao Huang, and Jiashi Feng. How far is video generation from world model: A physical law perspective. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=DLlVjZQ7vD.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=ZMnD6QZAE6.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=wPEIStHxYH.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Si Liu, Jianlan Luo, Liliang Chen, et al. Genie envisioner: A unified world foundation platform for robotic manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 88446–88463, 2026.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Shaowei Liu, Zhongzheng Ren, Saurabh Gupta, and Shenlong Wang. Physgen: Rigid-body physics-grounded image-to-video generation. In *European Conference on Computer Vision*, pp. 360–378. Springer, 2024.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance GPU based physics simulation for robot learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=fgFBtYgJQX_.

Antonio Montanaro, Luca Savant Aira, Emanuele Aiello, Diego Valsesia, and Enrico Magli. Motioncraft: Physics-based zero-shot video generation. *Advances in Neural Information Processing Systems*, 37:123155–123181, 2024.

Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 948–958. IEEE, 2026.

Boxiang Qiu, Liliang Chen, Yue Liao, Nan Wang, Lintao Wang, Jiayi Luo, Wenzhi Zhao, Shengcong Chen, Di Chen, Ye Li, et al. Ge-sim 2.0: A roadmap towards comprehensive closed-loop video world simulators for robotic manipulation. *arXiv preprint arXiv:2605.27491*, 2026.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-Kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Nagaswamy Rajesh, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Demonstrating GPU Parallelized Robot Simulation and Rendering for Generalizable Embodied AI with ManiSkill3. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.021.

AgiBot World Team. Agibot world 2026. `https://huggingface.co/datasets/agibot-world/AgiBotWorld2026`, 2026.

GigaWorld Team, Angyuan Ma, Boyuan Wang, Bohan Li, Chaojun Ni, Guo Li, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, et al. Gigaworld-1: A roadmap to build world models for robot policy evaluation. *arXiv preprint arXiv:2607.02642*, 2026a.

MotuBrain Team, Chendong Xiang, Fan Bao, Haitian Liu, Hengkai Tan, Hongzhe Bi, James Li, Jiabao Liu, Jingrui Pang, Kiro Jing, et al. Motubrain: An advanced world action model for robot control. *arXiv preprint arXiv:2604.27792*, 2026b.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *International Conference on Learning Representations*, volume 2025, pp. 73754–73776, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Yixuan Wang, Rhythm Syed, Fangyu Wu, Mengchao Zhang, Aykut Onol, Jose Barreiros, Hooshang Nayyeri, Tony Dear, Huan Zhang, and Yunzhu Li. Interactive world simulator for robot policy training and evaluation. In *Robot World Models*, 2026. URL `https://openreview.net/forum?id=x4qnERKsNI`.

Yuang Wang, Chao Wen, Haoyu Guo, Sida Peng, Minghan Qin, Hujun Bao, Xiaowei Zhou, and Ruizhen Hu. Precise action-to-video generation through visual action prompts. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12713–12724. IEEE, 2025.

Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11094–11104. IEEE, 2020.

Mutian Xu, Tianbao Zhang, Tianqi Liu, Zhaoxi Chen, Xiaoguang Han, and Ziwei Liu. Kinema4d: Kinematic 4d world modeling for spatiotemporal embodied simulation. *arXiv preprint arXiv:2603.16669*, 2026.

Qiyao Xue, Xiangyu Yin, Boyuan Yang, and Wei Gao. Phyt2v: Llm-guided iterative self-refinement for physics-grounded text-to-video generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18826–18836. IEEE, 2025.

Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=sFyTZEqmUY`.

Xiuyu Yang, Bohan Li, Shaocong Xu, Nan Wang, Chongjie Ye, Zhaoxi Chen, Minghan Qin, Yikang Ding, Zheng Zhu, Xin Jin, Hang Zhao, and Hao Zhao. ORV: 4d occupancy-centric robot video generation. In *Conference on Computer Vision and Pattern Recognition 2026*, 2026. URL https://openreview.net/forum?id=G8lNJF6W2s.

Hu Yue, Siyuan Huang, Yue Liao, Shengcong Chen, Pengfei Zhou, Liliang Chen, Maoqing Yao, and Guanghui Ren. Ewmbench: Evaluating scene, motion, and semantic quality in embodied world models. *arXiv preprint arXiv:2505.09694*, 2025.

Jie Zhang, Xiaoyue Chen, Anzhe Chen, Dayiheng Liu, Deqing Li, Gengze Zhou, Hale Yin, Haoqi Yuan, Haoyang Li, Jiahao Li, et al. Qwen-robotworld technical report: Unifying embodied world modeling through language-conditioned video generation. *arXiv preprint arXiv:2606.17030*, 2026a.

Ke Zhang, Cihan Xiao, Yiqun Mei, Jiacong Xu, and Vishal M Patel. Think before you diffuse: Llms-guided physics-aware video generation. *arXiv e-prints*, pp. arXiv–2505, 2025.

Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y Feng, Changxi Zheng, Noah Snavely, Jiajun Wu, and William T Freeman. Physdreamer: Physics-based interaction with 3d objects via video generation. In *European Conference on Computer Vision*, pp. 388–406. Springer, 2024.

Xiangdong Zhang, Jiaqi Liao, Shaofeng Zhang, Fanqing Meng, Xiangpeng Wan, Junchi Yan, and Yu Cheng. Videorepa: Learning physics for video generation through relational alignment with foundation models. *Advances in Neural Information Processing Systems*, 38:122647–122676, 2026b.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: A fine-grained world model for robot manipulation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9834–9844. IEEE, 2025.

A Appendix

A.1 More Related Works about Video World Models.

World models predict the future of an environment according to control signals (Ha & Schmidhuber, 2018; Hafner et al., 2023). Built on video generation models such as Wan (Wan et al., 2025) and Cosmos (Agarwal et al., 2025), video world models inherit visual and motion priors from internet-scale videos. They now can simulate interactive games (Bruce et al., 2024; Valevski et al., 2025), driving scenes (Hu et al., 2023; Gao et al., 2024), and robot manipulation (Yang et al., 2024; Zhu et al., 2025; Gao et al., 2026). Despite their visual quality, the generated videos frequently violate physical principles (Bansal et al., 2025; Motamed et al., 2026; Kang et al., 2025). Recently, PhysGen (Liu et al., 2024), PhysDreamer (Zhang et al., 2024), and MotionCraft (Montanaro et al., 2024) guide generation with a physics information. PhyT2V (Xue et al., 2025) and DiffPhy (Zhang et al., 2025) refine prompts with LLM reasoning at inference time, and VideoREPA (Zhang et al., 2026b) and Phantom align the generative model with latent physical representations. Dual-Sim is an action-conditioned world model for robotic manipulation, which integrates the physics simulator with the video world model. The physics simulator provides the exact kinematics of the robot, and the video world model provides the appearance and dynamics of the world.

A.2 Implementation Details.

We initialize the video expert from Cosmos 2.0 (Ali et al., 2025). It contains 28 Transformer layers, each with 16 attention heads of dimension 128. The state expert contains 28 Transformer blocks of width 512, with 16 attention heads of dimension 32. Isaac Lab serves as the physical simulator for robot simulation and rendering. We optimize Dual-Sim with AdamW using a learning rate of 10^{-4}, a batch size of 32, and 8 NVIDIA A100 GPUs. At inference time, we integrate the coupled video-state flow with 20 Euler steps.

A.3 Physical Simulated Video Example

Fig. 10 gives a concrete example of the simulated video described in Section 3.1. The displayed sim-video is rendered from the head-camera viewpoint and its robot mask is obtained directly from the simulator.

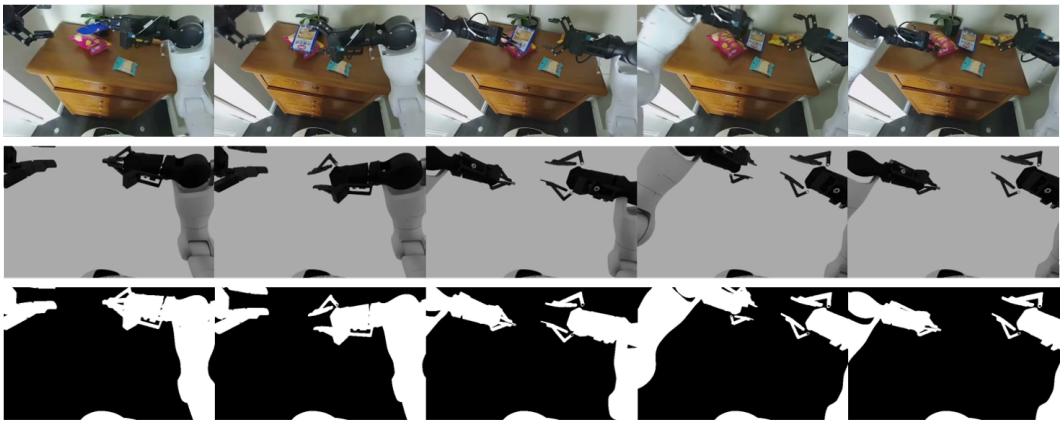

Figure 10: **Example outputs of the physics simulator.** From top to bottom: frames from the real head-camera video, the aligned sim-video $\widetilde{V}$ rendered against a blank background, and the corresponding binary robot mask M. The sim-video provides the robot motion and structure, while objects, background, and interaction effects are left to the generative simulator.

A.4 Evaluation Metrics

We summarize the evaluation metrics used in Tab. 1 and Tab. 3. PSNR is reported in dB, while all other metrics are reported on a 0–100 scale. Higher values indicate better performance.

Scene Consistency. Following EWMBench, Scene Consistency measures temporal appearance stability using DINOv2 features. For each generated frame, feature similarity is computed with both the preceding frame and the first frame, capturing short-term temporal smoothness and long-term appearance consistency.

PSNR and SSIM. PSNR and SSIM are computed between temporally aligned generated and ground-truth frames. PSNR measures pixel-level reconstruction fidelity and is reported in dB, while SSIM measures structural similarity and is scaled to $[0, 100]$.

Trajectory HSD, Dynamics, and nDTW. Following EWMBench, these metrics evaluate action consistency by comparing YOLO-World-extracted gripper trajectories from generated and ground-truth videos. Trajectory HSD is defined as the reciprocal of the symmetric Hausdorff distance between the two trajectories. Trajectory Dynamics compares their motion statistics using inverse Wasserstein distances between the distributions of gripper speed and speed change, weighted by the ratios of their respective ranges. Trajectory nDTW measures temporal trajectory agreement: given total alignment distance D and alignment path π, the raw score is the reciprocal of the mean alignment distance, $|\pi|/D$. The three raw scores are clipped and normalized to $[0, 100]$ using reference maxima of 24.979, 22.519, and 50.202, respectively.

EWMScore. Following the AGIBOT World Challenge, EWMScore aggregates Scene Consistency, normalized PSNR, and Trajectory nDTW into a single metric by taking their arithmetic mean, with all three components normalized to the 0–100 scale.

Embodiment SSIM. Embodiment SSIM measures structural fidelity specifically within the robot region. SAM3 is used to segment the robot foreground in the ground-truth frame, and the resulting mask is applied to both the generated and ground-truth frames. We then compute the SSIM map and average similarity only over foreground pixels. The resulting score is scaled to $[0, 100]$.

Trajectory Accuracy. Following WorldArena, Trajectory Accuracy evaluates gripper motion in normalized image coordinates using SAM3-based tracking. Among the left and right ground-truth grippers, we select the trajectory with the larger spatial extent and compare it with the corresponding generated trajectory using fastDTW. Given total alignment distance D and alignment path π, the raw score is $|\pi|/D$; invalid generated trajectories receive a score of zero. The final metric is normalized as

$$100 \ \mathrm{clip}\left(\frac{|\pi|/D}{40.854},\, 0,\, 1\right).$$

A.5 More Discussion on Video-State Generation

Interaction-Aware Refinement of the Simulation Prior. A potential concern with kinematic seeding mechanism is that it may overly constrain the generated embodiment to follow the simulated trajectory, leaving limited flexibility for interaction-dependent corrections. To examine this, we consider a representative bottle-grasping sequence in which Dual-Sim must adapt the idealized simulation prior to the inferred physical interaction. As shown in Fig. 11, before contact, at frames 60 and 80, the simulated, predicted, and ground-truth gripper states closely agree. Once grasping occurs, however, the simulated gripper closes completely, whereas the real gripper remains partially open around the bottle. Dual-Sim departs from the simulated trajectory and predicts a gripper opening substantially closer to the ground truth. At frame 100, the predicted openness is 0.475, compared with 0.469 for GT and 0 for Sim; at frame 120, the corresponding values are 0.516, 0.465, and 0. The shaded region highlights this interaction-induced correction relative to the simulation trajectory. This transition from agreement before contact to adaptive correction during grasping shows that kinematic seeding serves as an editable physical prior rather than a fixed constraint, allowing Dual-Sim to reconcile simulated kinematics with real scene interactions.

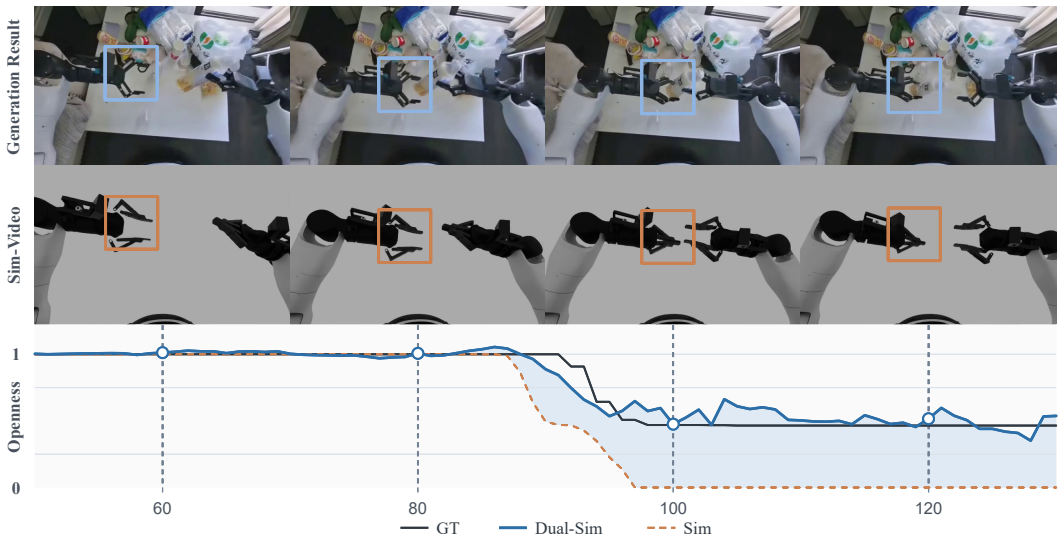

Figure 11: **Dual-Sim edits the simulated gripper trajectory.** Generated (top) and simulated (middle) frames align with the gripper openness trajectories below. Dual-Sim predicts a partial opening closer to GT, while Sim closes fully. Blue shading highlights the edit relative to Sim.

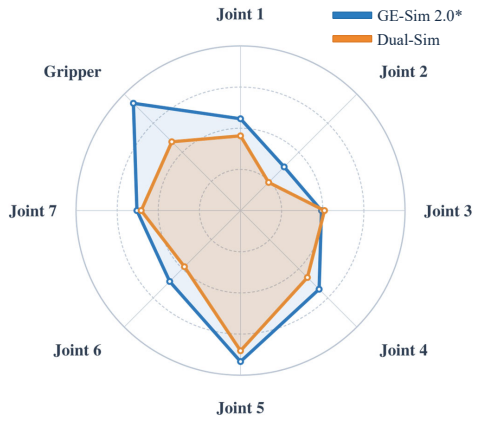

Figure 12: **Joint-wise proprioceptive state error.**

Joint-wise Proprioceptive State Accuracy. Beyond the results in Fig. 6, We further analyze proprioceptive state accuracy at the joint level, focusing on the latter half of each rollout where accumulated prediction errors become more evident. As shown in Fig. 12, Dual-Sim exhibits consistently reduced error across nearly all joints, indicating improved long-horizon state fidelity across the embodiment. The improvement is particularly pronounced for the gripper, whose state is tightly coupled with contact events and object interactions. This result further reflects Dual-Sim's improved modeling of interaction-dependent state evolution.

A.6 Additional Qualitative Results on AgiBot World

We provide additional qualitative comparisons in Fig. 13 to complement the results in the main paper. In the first two rows, Dual-Sim closely matches the ground-truth robot configuration, whereas the baselines exhibit noticeable kinematic deviations. In the middle rows, Dual-Sim preserves coherent embodiment geometry and appearance throughout the rollout, while the baselines progressively suffer from structural distortion and visual degradation. In the last three rows, Dual-Sim more faithfully captures the state of the manipulated object, whereas the baselines struggle to model the corresponding interactions. This improvement is consistent with our design of offloading embodiment kinematics to the simulator, allowing the generative world model to devote greater modeling capacity to scene interactions.

A.7 Additional Qualitative Results on Cross-Embodiment Generalization

Additional qualitative comparisons are provided in Fig. 14. Dual-Sim delivers consistent improvements over the baselines across all four embodiments. It faithfully preserves robot kinematics and embodiment appearance, achieving strong zero-shot performance even on the unseen Franka-Panda.



Figure 13: Qualitative results of video generation.

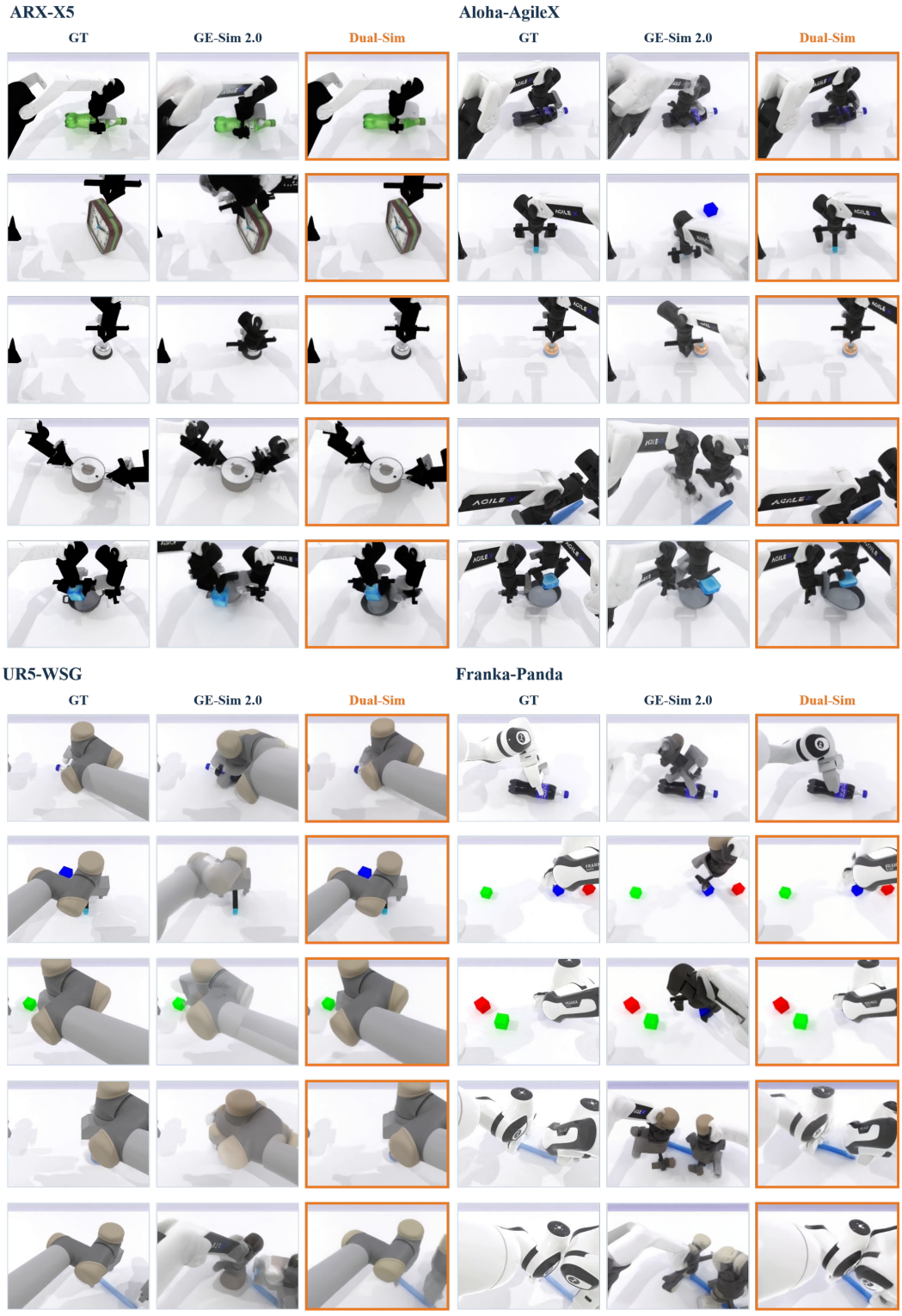

Figure 14: Qualitative results of cross-embodiment generalization

In contrast, the baselines frequently confuse visually similar embodiments, exhibit structural distortions and inaccurate action following, and degrade substantially on the unseen embodiment. These results suggest that kinematic seeding provides an effective interface between embodiment-specific physical priors and a shared generative world model, enabling stronger embodiment fidelity and cross-embodiment generalization.

A.8 Closed-loop Policy Training and Evaluation

We study two uses of Dual-Sim: evaluating policies through closed-loop interaction and generating trajectories for policy post-training.

Robot Setup. We conduct real-world experiments on a tabletop bimanual platform comprising two YAM Ultra1 robotic arms from I2RT, each with six actuated joints and a gripper. Three RealSense RGB cameras provide a head view of the workspace and one wrist-mounted view per arm. Videos are recorded at 640×480 resolution and a nominal frame rate of 30 fps. Robot observations also include joint positions and gripper states.

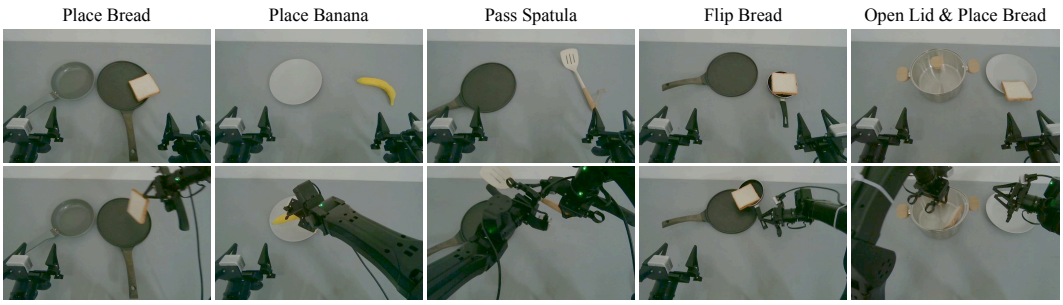

Figure 15: **Policy evaluation tasks on the YAM platform.** Real demonstrations of the five evaluation tasks, covering object placement, handover, and bimanual manipulation.

A.8.1 Closed-loop Policy Evaluation

Task definitions. We evaluate five manipulation tasks using the following language instructions: *Place Bread*: Move bread from the large pan to the small pan. *Place Banana*: Pick up a banana and place it on the plate. *Pass Spatula*: Hand a spatula from the right arm to the left arm, then place it. *Flip Bread*: Tilt the small pan to transfer bread into the large pan. *Open Lid & Place Bread*: Open the pot with the left arm, place bread inside with the right arm, and close the lid. Fig. 15 illustrates these five tasks.

Evaluation protocol. We evaluate π_0, $\pi_{0.5}$, and DP on the five tasks in Fig. 15, targeting 15 trials per task–policy pair in each environment. We use matched initial observations and robot states for execution in the real world and execution in Dual-Sim. Policy actions condition Dual-Sim to produce subsequent observations, which are fed back for closed-loop execution. Outcomes are manually labeled as success or failure based on completion of the instructed task.

Success-rate correlation and numerical agreement. The statistical unit is a task–policy pair. For task t and policy p, let x_{tp} and y_{tp} denote real-world and Dual-Sim success rates, respectively, on a $[0, 1]$ scale. Each of the 15 pairs receives equal weight. Pearson correlation measures linear association. Spearman correlation uses average ranks for tied values. We compute errors in percentage points as

$$\mathrm{MAE} = \frac{100}{N} \sum_{t,p} |y_{tp} - x_{tp}|, \qquad \mathrm{Bias} = \frac{100}{N} \sum_{t,p} (y_{tp} - x_{tp}), \quad N = 15.$$

Positive bias means a higher success rate in Dual-Sim than in the real world. The fitted line is an unweighted ordinary least-squares regression of y on x. $y = x$ represents exact numerical agreement. Tab. 4 reports the pooled and per-policy metrics.

Tab. 4 shows that real-world and Dual-Sim success rates are positively correlated across tasks for all three policies. Pearson correlation reaches 0.9900 for $\pi_{0.5}$ and 0.9780 for DP, while π_0 also shows a positive association ($r = 0.8277$). For DP, Spearman $\rho = 1.0000$ indicates that the five tasks have the same success-rate ordering in both environments. These results indicate that Dual-Sim captures task-dependent performance trends for both language-conditioned policies and single-task DP models, supporting closed-loop policy evaluation on the tested tasks.

Table 4: **Dual-Sim success-rate association and agreement.** The pooled row uses all 15 task–policy pairs. Individual-policy rows are computed across the five tasks. Bias is the success rate in Dual-Sim minus that in the real world. Errors are given in percentage points (pp).

Policy	N	Pearson r	Spearman ρ	MAE (pp)	Bias (pp)
All	15	0.9168	0.9278	13.3	+9.2
π_0	5	0.8277	0.8208	13.7	+3.1
$\pi_{0.5}$	5	0.9900	0.9747	10.7	+9.1
DP	5	0.9780	1.0000	15.4	+15.4

Comparison with GE-Sim2. We compare GE-Sim2 and Dual-Sim on the same five tasks and three policy families, using the same real-world success rates as the reference. GE-Sim2 success rates are computed from annotated rollout outcomes. All metrics give equal weight to the 15 task–policy pairs.

Fig. 16 shows the GE-Sim2 results. Dual-Sim achieves higher Pearson correlation than GE-Sim2 (0.9168 vs. 0.8122), indicating stronger agreement in success-rate variation across the task–policy pairs. Its MAE is also lower (13.3 vs. 20.3 percentage points), showing that the simulated success rates are closer to the real-world rates on average. The higher Spearman correlation (0.9278 vs. 0.8395) further indicates closer agreement in the ordering of these task–policy pairs. Together, these results support more accurate aggregate success-rate evaluation with Dual-Sim on the evaluated tasks and policies.

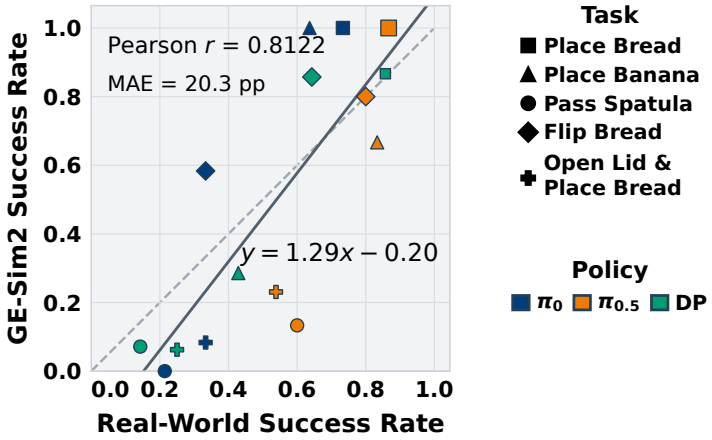

Figure 16: **Policy evaluation in GE-Sim2.** GE-Sim2 success rates are compared with the same real-world reference used for Dual-Sim across five tasks and three policies. The dashed line marks equal success rates, and the solid line is the least-squares fit.

A.8.2 Qualitative Policy Evaluation

Fig. 17 and 18 show execution in Dual-Sim for object placement and bimanual manipulation. In Place Bread and Place Banana, the object is grasped, transported, and released into the target container. Flip Bread shows bread moving into the receiving pan as the held pan tilts. Open Lid & Place Bread proceeds through lid opening, bread placement, and lid closing. These examples show that Dual-Sim can generate successive stages of a manipulation task within a single execution.

Multi-view observations at key interactions. Fig. 19 and 20 show head and wrist views from the same executions in Dual-Sim. For Place Bread, the right-wrist view shows the held bread and the left-wrist view shows the receiving pan. During placement in Open Lid & Place Bread, the left-wrist view reveals bread inside the pot while the arm partially occludes the head view. Together, the three views make grasping and placement visible when a single camera cannot show the full interaction.

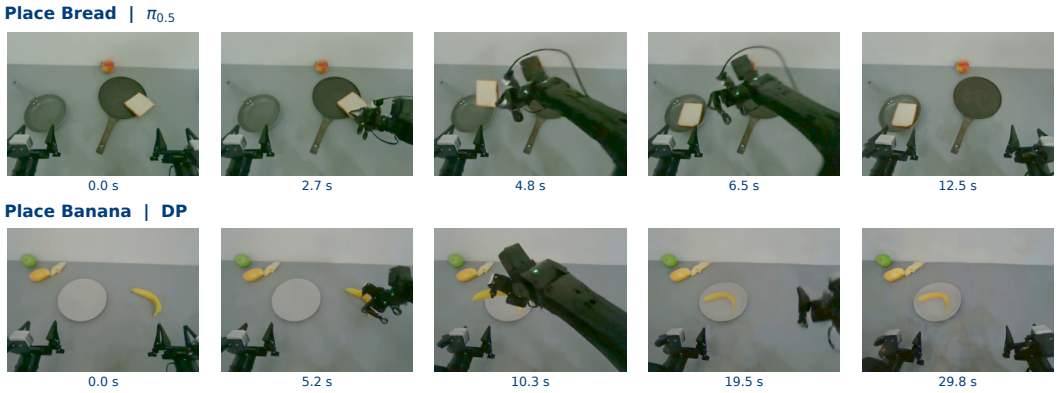

Figure 17: **Closed-loop policy execution in Dual-Sim.** Place Bread and Place Banana show the object being grasped, transported, and released into the target container.

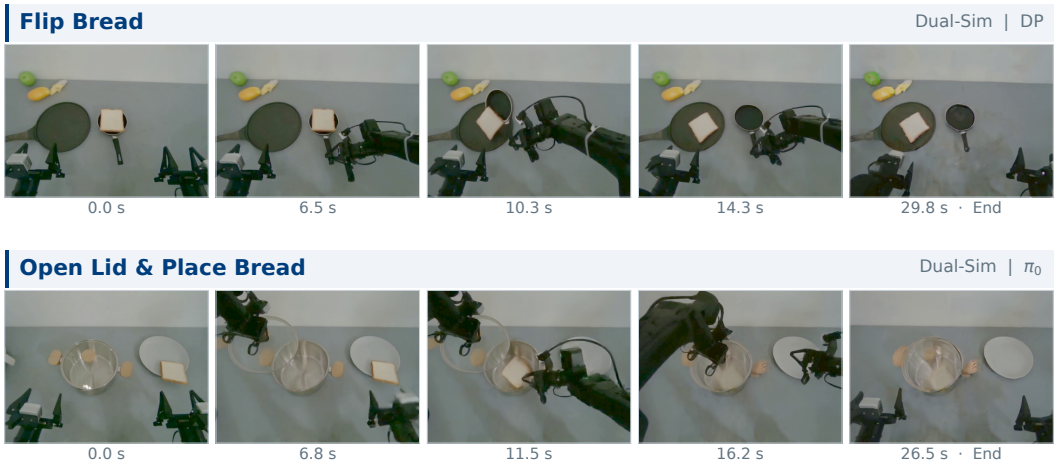

Figure 18: **Multi-stage execution in Dual-Sim.** Flip Bread shows bread transfer by tilting the pan. Open Lid & Place Bread shows lid opening, bread placement, and lid closing.

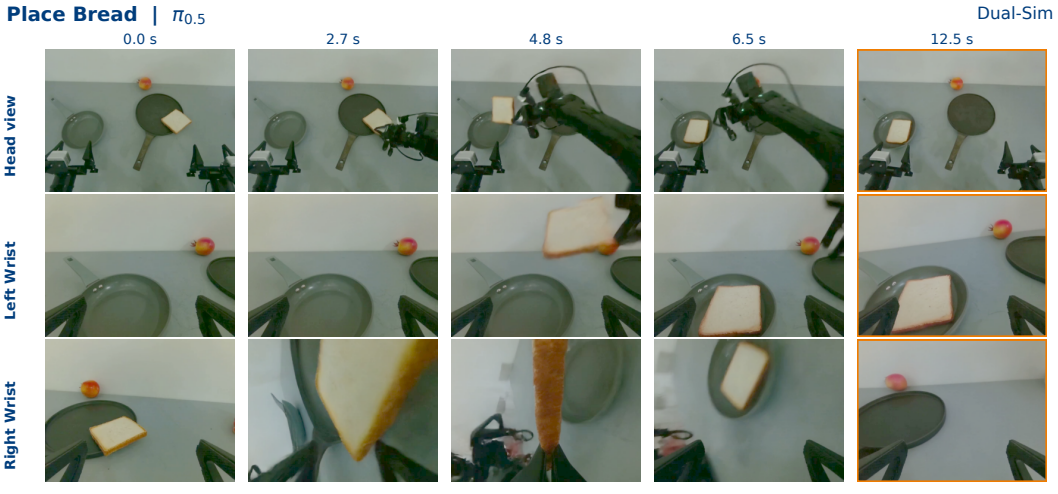

Figure 19: **Multi-view observations of bread transfer in Dual-Sim.** During execution in Dual-Sim with $\pi_{0.5}$, the wrist views reveal the held bread and the receiving pan during transfer.

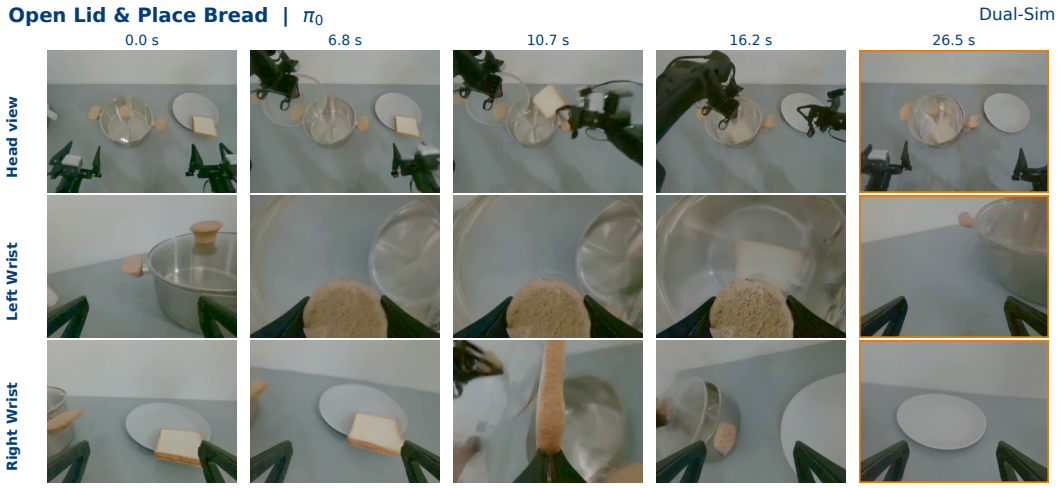

Figure 20: **Multi-view observations of lid manipulation in Dual-Sim.** During execution in Dual-Sim with π_0, the left-wrist view reveals bread inside the pot at 16.2 s despite partial occlusion in the head view.

A.8.3 Policy Post-training

We assess the utility of generated trajectories on four tasks: Place Apple, Place Pomegranate, Place Bun, and Sweep Sponge. *Real only* evaluates a π_0 policy fine-tuned using real demonstrations, with 20 demonstrations per evaluated task. On the other hand, we train π_0 using these real demonstrations together with trajectories generated in GE-Sim2 or Dual-Sim, obtaining *Real + GE-Sim2* and *Real + Dual-Sim*, respectively. The two policies are jointly post-trained on the four tasks. We evaluate all three policies in the real world.

Task definitions. *Place Apple*: pick up an apple and place it in the target dish. *Place Pomegranate*: move a pomegranate into the target bowl. *Place Bun*: place a steamed bun in the air-fryer basket and push the drawer closed. *Sweep Sponge*: use a brush to sweep a sponge into the dustpan. Fig. 21 shows the corresponding task examples.

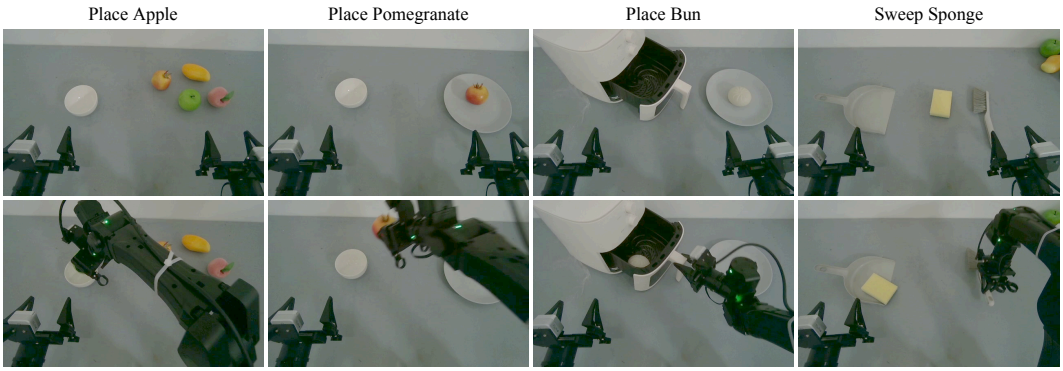

Figure 21: **Tasks for policy post-training.** Real demonstrations of Place Apple, Place Pomegranate, Place Bun, and Sweep Sponge. The tasks cover fruit placement, drawer closure, and tool use.

Data construction. We collect 30 real-world initial scenes per task, or 120 across the four tasks, to provide starting observations for trajectory generation. GE-Sim2 and Dual-Sim use these initial scenes to generate their respective trajectories. Generated trajectories are converted to the same training format as the real demonstrations, including three camera views, 14-dimensional robot states and action targets, and task instructions. Tab. 5 summarizes the resulting datasets.

Table 5: **Dataset composition for policy post-training.** Each augmentation condition combines the same real demonstrations with trajectories from the corresponding world model.

Task	Real demos.	GE-Sim2	Dual-Sim
Place Apple	20	30	30
Place Pomegranate	20	30	30
Place Bun	20	30	30
Sweep Sponge	20	30	30
Total	80	120	120

Training configuration. Both Real + GE-Sim2 and Real + Dual-Sim jointly post-train π_0 on the four tasks, starting from the same policy previously fine-tuned on multiple tasks. Training uses the standard flow-matching objective. Sampling is balanced across tasks. Within each task, frames are sampled from the combined real and generated data, so each source's contribution depends on its frame count. Each sample supervises a 30-step action chunk on a 30 Hz time grid, with observation-relative joint targets and absolute gripper targets.

Qualitative examples of generated training data. Fig. 22 and 23 show Dual-Sim trajectories used for policy post-training. Place Apple and Place Pomegranate show transport into a target container. Place Bun includes placement followed by drawer closure, and Sweep Sponge shows the brush moving the sponge into the dustpan. These trajectories provide action supervision for successive stages of object placement, container manipulation, and tool use.

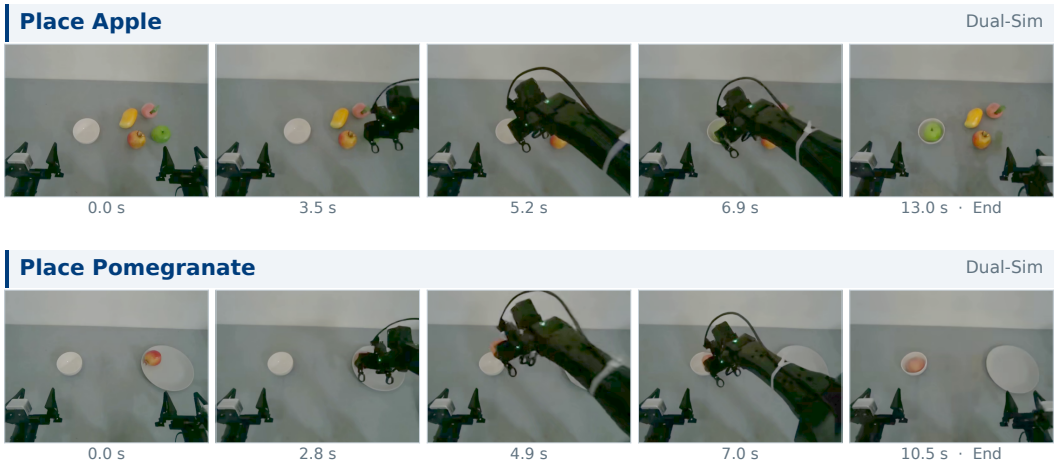

Figure 22: **Generated training trajectories for fruit placement.** Dual-Sim trajectories used for post-training show an apple and a pomegranate being moved into their target containers.

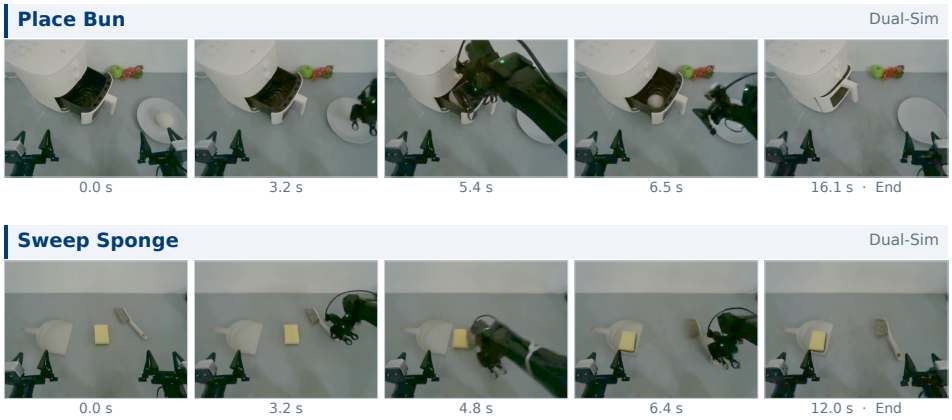

Figure 23: **Generated training trajectories for container and tool interaction.** Place Bun shows placement into the air-fryer basket and subsequent drawer closure. Sweep Sponge shows the brush moving the sponge into the dustpan. Both are Dual-Sim trajectories used for policy post-training.

A.8.4 Evaluation in the Real World after Post-training

Each policy is evaluated in 15 trials per task.Task completion is judged manually as success or failure. Table. 6 reports successful trials, denominators, and success rates from these annotations.

Table 6: **Evaluation of the three policies in the real world.** Each task entry shows successes / trials (success rate). Brackets give the 95% Wilson interval in percent. The mean is task-averaged.

Task	Real only	Real + GE-Sim2	Real + Dual-Sim
Place Apple	4/15 (26.7%) [10.9, 52.0]	8/15 (53.3%) [30.1, 75.2]	12/15 (80.0%) [54.8, 93.0]
Place Pomegranate	8/15 (53.3%) [30.1, 75.2]	11/15 (73.3%) [48.0, 89.1]	12/15 (80.0%) [54.8, 93.0]
Sweep Sponge	6/15 (40.0%) [19.8, 64.3]	7/15 (46.7%) [24.8, 69.9]	11/15 (73.3%) [48.0, 89.1]

In Fig. 9, Overall is the equally weighted mean of the four task success rates. Dashed lines mark the Real only reference, and arrows indicate absolute gains in percentage points.

The 95% Wilson intervals in Tab. 6 summarize trial-level sampling uncertainty for each task and evaluated policy. Real + Dual-Sim achieves higher success rates than Real + GE-Sim2 on all four tasks, with gains of 26.7, 6.7, 38.5, and 26.7 percentage points on Place Apple, Place Pomegranate, Place Bun, and Sweep Sponge, respectively. The largest gain occurs on Place Bun, where successful trials increase from 46.2% to 84.6% for a task requiring both placement and drawer closure. Across the four tasks, the mean success rate increases from 54.9% with GE-Sim2 data to 79.5% with Dual-Sim data. These results support the value of Dual-Sim trajectories for policy post-training under the evaluated setup, with benefits spanning object placement and multi-stage manipulation.

Fig. 24 and 25 show independent executions of the three policies in the real world. The apple examples distinguish placing the requested object from placing a distractor. The bun and sponge examples show the final actions required for success, including closing the drawer and moving the sponge into the dustpan. The policy post-trained with Dual-Sim data completes these task requirements in the displayed trials, showing successful target selection and task completion in the real world.

A.9 AI Use Statement

AI tools were used to polish the paper writing and assist with the selection of figure color. All AI-assisted revisions and color choices were approved by the authors. The authors take full responsibility for the accuracy and integrity of the final submission, including its technical descriptions, citations, experimental claims, and figures.



Figure 24: **Execution in the real world on Place Apple and Place Pomegranate.** In the displayed apple trials, Real only and Real + GE-Sim2 place a yellow distractor in the dish, while Real + Dual-Sim places the apple. All three pomegranate trials succeed.

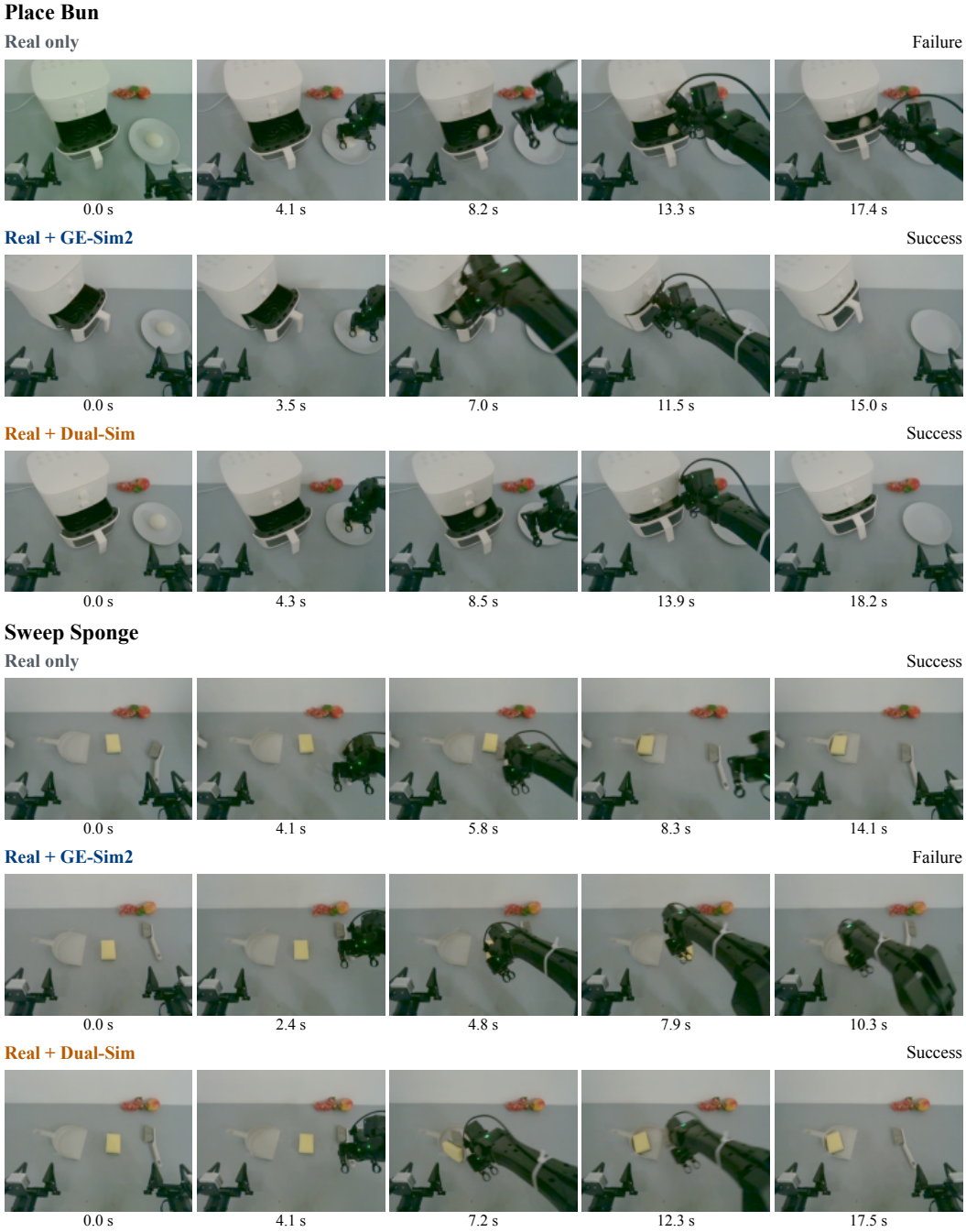

Figure 25: **Execution in the real world on Place Bun and Sweep Sponge.** In the displayed bun trials, Real only leaves the drawer open after placement, while both post-trained policies close it. In Sweep Sponge, Real + GE-Sim2 leaves the sponge outside the dustpan, while Real only and Real + Dual-Sim complete the transfer.