PHYSCANVAS: A VISION-TACTILE WORLD MODEL FOR ROBOT POLICY EVALUATION AND LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Evaluating and post-training robot policies through physical interaction is costly and difficult to scale, motivating action-conditioned world models as interactive simulators. However, visually plausible predictions can misrepresent contact dynamics and task outcomes, providing unreliable feedback for policy evaluation and learning. We introduce PHYSCANVAS, a state-consistent visuo-tactile world model for contact-rich robot manipulation. PHYSCANVAS organizes multi-view RGB, tactile images, marker displacement, and task state into a unified multimodal canvas. All modalities share a video VAE and pretrained generative backbone, allowing optical touch to reuse visual and temporal priors. Motion images derived through forward kinematics and camera projection provide spatially aligned action conditioning, while an initial RGB anchor and updated reference observations guide sequential generation. Our consistency-aware post-training uses the frozen Scene Reward Model, Modality Reward Model, and State Reward Model to supervise scene preservation, visual–tactile compatibility, and agreement with real task outcomes. The resulting world model supports closed-loop policy rollouts for evaluation and provides imagined interactions and rewards for GRPO policy updates. Across four real-world manipulation tasks, when policies are rolled out in the world model, State Reward Model supervision increases the arithmetic mean of task-wise Pearson correlations between world-model and real-world success rates from 0.602 to 0.885. Policy post-training in the reward-guided world model further increases $\pi_{0.5}$'s mean real-world success rate from a 59.1% SFT baseline to 70.8%. These results demonstrate the value of unified multimodal modeling and consistency-aware post-training for robot policy evaluation and learning.

1 INTRODUCTION

Robot world models predict how environments evolve in response to actions, offering low-cost, reproducible interactions for policy evaluation and learning (Ha & Schmidhuber, 2018; Wu et al., 2023; Wang et al., 2026). Contact-rich manipulation requires both visual observations of scene dynamics and tactile feedback about contact changes (Yuan et al., 2017; Lambeta et al., 2020; Lee et al., 2020; Guzey et al., 2024). A world model that captures these complementary signals and their associated task outcomes could reduce the need for repeated physical trials.

Recent action-conditioned video models combine pretrained generative priors with action representations to support interactive prediction, policy evaluation, and learning in imagination (Wu et al., 2026; Wang et al., 2026; Yang et al., 2026a). Visuo-tactile world models further predict scene and contact dynamics jointly to support contact-aware control (Zheng et al., 2026; Ma et al., 2026). Yet better generation quality does not necessarily make a model reliable for decision-making: visual fidelity alone does not establish action adherence or accurate failure prediction, and rollout errors can mislead policy optimization (Yang et al., 2026b; Jiang et al., 2026). For contact-rich interaction, visual observations, tactile feedback, and predicted task outcomes must describe the consequences of the same action. For example, after an object slips, predicting stable tactile contact or eventual success provides misleading feedback despite visually plausible frames. The challenge is therefore to jointly predict multimodal futures while preserving scene, cross-modal, and task-outcome consistency for policy evaluation and learning.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

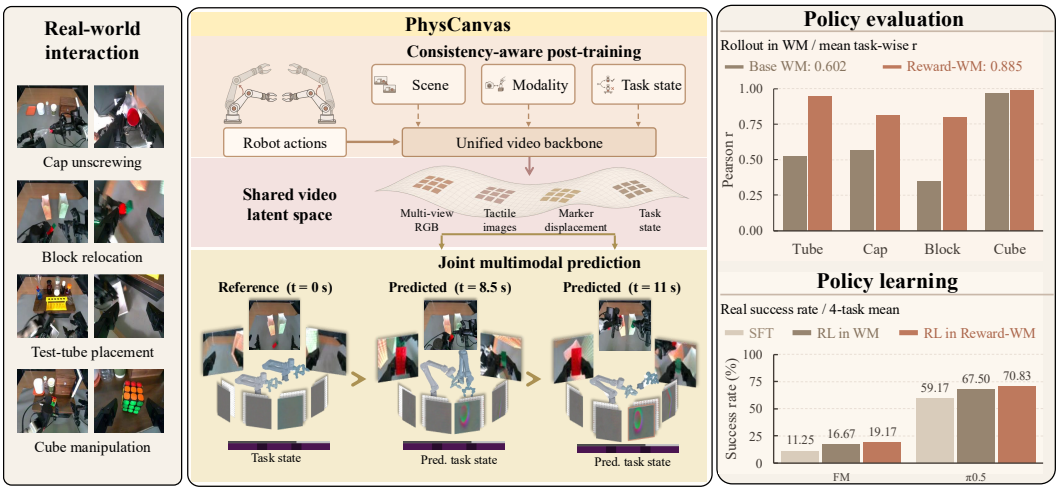


Figure 1: **PhysCanvas as a multimodal simulator.** Action-conditioned visual–tactile rollouts support policy evaluation and learning.

To address this challenge, we introduce PHYSCANVAS, a state-consistent visuo-tactile world model for robot policy evaluation and learning. PHYSCANVAS places multiple modalities in a shared generative space and uses consistency-aware post-training to preserve the interaction dynamics and task outcomes induced by actions, providing interactive, coherent observations.

Specifically, PHYSCANVAS places multi-view RGB, tactile images, marker displacement, and task state on a unified canvas predicted by a shared video VAE and generative backbone, allowing optical touch to reuse pretrained visual and temporal priors (Wan Team et al., 2025). Actions are encoded as spatially aligned motion images through forward kinematics and camera projection (Wu et al., 2026), while an initial RGB anchor and updated references guide sequential generation. During post-training, the frozen Scene Reward Model, Modality Reward Model, and State Reward Model enforce scene preservation, visual–tactile compatibility, and agreement with real outcomes, including both success and failure. The resulting model supports closed-loop evaluation and supplies imagined interactions and rewards for GRPO policy updates (Shao et al., 2024; Yu et al., 2026).

We evaluate PHYSCANVAS on four real-world manipulation tasks: test-tube placement, bottle-cap unscrewing, block relocation, and Rubik's Cube manipulation. When policies are rolled out in the world model, post-training improves the Pearson correlation between their world-model and real-world success rates on every task, increasing the arithmetic mean of task-wise correlations from 0.602 to 0.885. Each task includes nine evaluation batches spanning ACT, FM, and $\pi_{0.5}$ (Zhao et al., 2023; Lipman et al., 2023; Black et al., 2025); the batches are evaluated at different times, and each task receives equal weight in the summary. Policy optimization through reward-guided imagined interactions further improves the mean real-world success rate of $\pi_{0.5}$ by 11.7 percentage points over SFT, from 59.1% to 70.8% across the four tasks. These results demonstrate utility for reliable evaluation and policy improvement.

Our contributions are threefold: (1) a unified multimodal canvas that jointly models vision, touch, and task state using a shared video generation framework; (2) consistency-aware post-training that constrains generated interactions through scene, modality, and task-state supervision; and (3) a framework for policy evaluation and learning in imagination, evaluated on real robots.

2 RELATED WORK

2.1 ACTION-CONDITIONED WORLD MODELS FOR ROBOTICS

Action-conditioned video models have advanced robot planning and interactive prediction (Finn & Levine, 2017; Yang et al., 2024; Wu et al., 2024; 2023; Micheli et al., 2023; Bruce et al., 2024). Pre-trained video priors (Ho et al., 2022; Blattmann et al., 2023; Wan Team et al., 2025) and embodied action conditioning (Jiang et al., 2025; Wu et al., 2026) improve controllability, while unified archi-

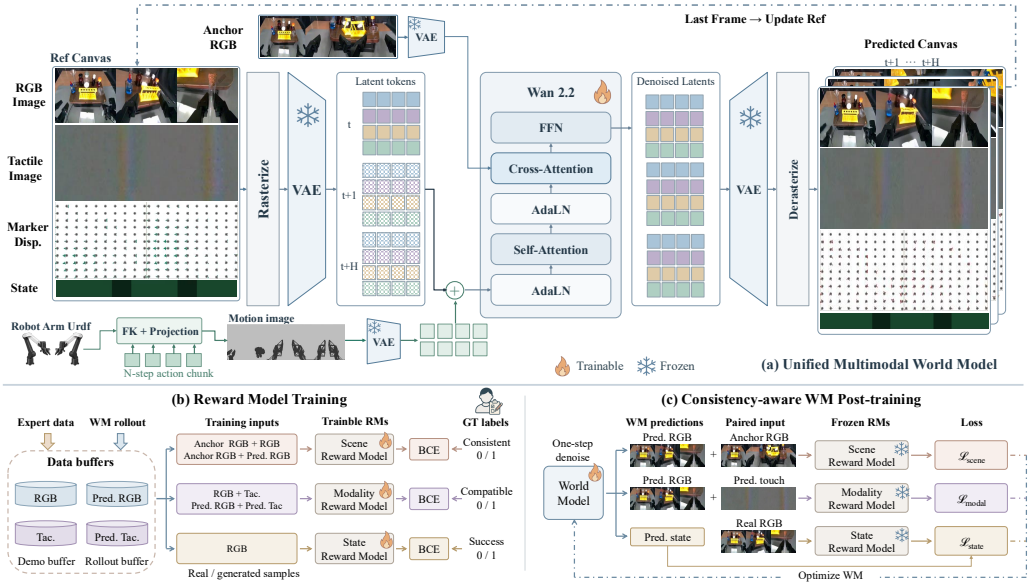


Figure 2: **Overview of PHYSCANVAS.** A unified canvas supports action-conditioned multimodal generation, with scene, state, and modality consistency guiding post-training.

tectures connect video prediction with policy modeling (Bi et al., 2026; Kim et al., 2026). Extending these visual simulators to contact-rich manipulation also requires predicting the tactile observations on which policies act.

Imagined rollouts also support policy evaluation and learning (Ha & Schmidhuber, 2018; Hafner et al., 2020; Hansen et al., 2024; Hafner et al., 2025; Wang et al., 2026; Jiang et al., 2026), making reliable action consequences and task outcomes essential. Recent work studies long-horizon consistency and simulator validity (GigaWorld Team et al., 2026; Yang et al., 2026b), or improves generation through reward-based optimization (Bardhan et al., 2026; Wang et al., 2025). PHYSCANVAS brings these requirements to joint visual–tactile prediction through scene and task-state consistency supervision, and evaluates generated rollouts against real policy outcomes.

2.2 VISUO-TACTILE WORLD MODELING

Touch exposes contact, deformation, and slip that can be ambiguous in RGB, motivating joint visual–tactile prediction (Yuan et al., 2017; Lambeta et al., 2020; Lee et al., 2020; Guzey et al., 2024). OmniVTA (Zheng et al., 2026) combines two-stream dynamics with contact-aware action generation; FeelWorld (Ma et al., 2026) predicts contact and slip states through contact-gated attention. DexTouch-WM (Qin et al., 2026) couples video and tactile experts, whereas DexTacWAM (Yuan et al., 2026) integrates finger-specific tactile latents into video diffusion. TACO (Liu et al., 2026) uses generated visuo-tactile corrections for policy post-training. Rather than add a separate tactile predictor, PHYSCANVAS renders tactile appearance and marker displacement as regions of the same image-native canvas as RGB and task status. The shared visual VAE and video denoiser then model aligned sensory futures, with explicit supervision of visual–tactile agreement.

3 METHOD

We first define the prediction problem (Section 3.1) and introduce our unified multimodal representation (Section 3.2). We then describe world-model post-training (Section 3.3) and its use for policy evaluation and policy post-training (Section 3.4).

3.1 Problem Formulation

We consider an offline dataset $\mathcal{D} = \{\tau_n\}_{n=1}^{N}$ of robot trajectories. Each trajectory is a time-aligned sequence of observation–action pairs,

$$\tau_n = ((\mathbf{o}_0, \mathbf{a}_0), \ldots, (\mathbf{o}_{T_n-1}, \mathbf{a}_{T_n-1})) , \tag{1}$$

where $\mathbf{o}_t = (\mathbf{o}_t^{\text{cam}}, \mathbf{o}_t^{\text{tac}}, \mathbf{s}_t)$ comprises synchronized multi-view camera observations, tactile sensor measurements, and task-state information $\mathbf{s}_t$. The action condition $\mathbf{a}_t$ is the target dual-arm joint configuration, including the gripper state, associated with time t. Thus, $\mathbf{a}_{t:t+H}$ contains the current configuration and H future target configurations, for a total of $H + 1$ states, and describes the commanded robot motion over the prediction horizon.

We build our world model from a pretrained image-to-video (I2V) model and extend its visual and temporal priors to action-conditioned multimodal prediction. Given the fixed episode-initial RGB anchor $\mathbf{o}_0^{\text{cam}}$, current multimodal reference $\mathbf{o}_t$, and target joint-configuration sequence $\mathbf{a}_{t:t+H}$, the model predicts the next H frames:

$$p_\theta(\mathbf{o}_{t+1:t+H} \mid \mathbf{o}_0^{\text{cam}}, \mathbf{o}_t, \mathbf{a}_{t:t+H}) . \tag{2}$$

3.2 Unified Multimodal Canvas

Image-native modality rendering. Robotic video world models primarily model vision, while touch often requires a separate representation. Because optical tactile observations also exhibit local image structure, with contact encoded as localized texture and deformation, we test whether they can share a visual latent space with RGB instead of requiring a dedicated tactile VAE. To test this premise, we compare pretrained visual VAEs, without tactile fine-tuning, against the tactile-specific UniT VAE on tactile-image reconstruction (Table 1). The Wan VAE achieves higher PSNR (35.51 vs. 31.72) and lower LPIPS (0.0139 vs. 0.0210), suggesting that its visual representation transfers to optical touch. Figure 3 complements this comparison with VAE reconstructions of RGB, tactile images, RGB-encoded marker displacement, and task state.

Table 1: Tactile image reconstruction. †: zero-shot, without tactile fine-tuning.

Model	PSNR↑	SSIM↑	LPIPS↓
UniT	31.72	**0.9613**	0.0210
Wan VAE†	**35.51**	0.9539	**0.0139**
SD ft-MSE†	30.73	0.9020	0.0600
SDXL fp16-fix†	32.28	0.9226	0.0399

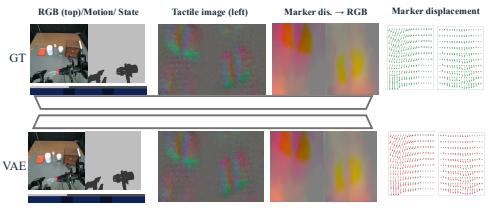


Figure 3: **Multimodal VAE reconstruction.** Inputs (top) and reconstructions (bottom).

Building on this evidence, we adopt the pretrained Wan image-to-video (I2V) model (Wan Team et al., 2025) to exploit its visual and temporal priors for joint multimodal prediction. PhysCanvas organizes camera RGB $\mathbf{I}_t^{\text{cam}}$, tactile images $\mathbf{I}_t^{\text{tac}}$, marker-displacement fields $\mathbf{M}_t^{\text{tac}}$, and task state $\mathbf{s}_t$ in a shared RGB canvas, allowing all modalities to share the visual VAE and video-generation backbone. Tactile images capture contact locations and local surface geometry, while marker displacement records each marker's three-dimensional offset from its reference configuration, providing cues to contact deformation and relative object motion. We map its x, y, and z components to R, G, and B channels through g_{disp}, using fixed normalization across training and inference. Task state indicates whether the task is complete. Predicting it serves as an auxiliary objective that teaches the world model high-level task semantics; the predicted state is not used by downstream policies. Color denotes the task and texture the outcome; Figure 10 shows all encodings. RGB and tactile images are inserted directly, whereas marker displacement and state use these image mappings. Camera views are ordered horizontally and modality tiles are stacked vertically:

$$\mathbf{x}_t = \text{Rasterize}(\mathbf{o}_t) = \text{Concat}_v\left[\mathbf{I}_t^{\text{cam}}, \mathbf{I}_t^{\text{tac}}, g_{\text{disp}}(\mathbf{M}_t^{\text{tac}}), g_{\text{state}}(\mathbf{s}_t)\right] . \tag{3}$$

This layout preserves within-modality spatial structure while exposing all modalities to shared spatiotemporal self-attention. Masking or omitting tactile regions supports RGB-only and visuo–tactile configurations without changing the backbone.

Action-conditioned latent dynamics. After arranging the prediction modalities on the canvas, we condition its evolution on robot motion. Following iMaC (Wu et al., 2026), we use camera intrinsics and extrinsics together with the robot's URDF to project the arm and gripper configuration into each camera view, producing action images that spatially align robot motion with visual observations.

We perform joint prediction in latent space. The canvas sequence $\mathbf{x}$ and the spatially arranged action-image sequence $\mathbf{m}$ are separately encoded by the video VAE E and tokenized by P, giving $\mathbf{z} = P(E(\mathbf{x}))$ and $\mathbf{z}^a = P(E(\mathbf{m}))$. The action layout is aligned with the canvas token grid so that conditioning can be added position-wise. At denoising time τ, we fuse action tokens with the noisy canvas tokens as

$$\widetilde{\mathbf{z}}_{\tau,i} = \mathbf{z}_{\tau,i} + M_i \mathbf{z}^a_i, \qquad M_i = \begin{cases} 0, & i \in \mathcal{S}_{\text{state}}, \\ 1, & \text{otherwise}, \end{cases} \tag{4}$$

where i indexes aligned spatiotemporal tokens and $\mathcal{S}_{\text{state}}$ contains the task-state token positions. Excluding direct action addition at these positions encourages state prediction to draw on the other modalities through shared self-attention, rather than a dedicated action-to-state input.

The fused tokens are processed by the Wan 2.2 diffusion transformer (Wan Team et al., 2025). Let $\mathbf{z}^{\text{anc}} = P(E(\mathbf{I}_0^{\text{cam}}))$ be the episode-initial RGB anchor, fixed throughout the rollout, and $\mathbf{r} = P(E(\mathbf{x}_t))$ the current multimodal reference tokens, updated from the preceding chunk's last frame. The transformer receives $\mathbf{r}$ as observed reference tokens and $\mathbf{z}^{\text{anc}}$ through cross-attention. We train the model with flow matching. Using $\tau = 0$ for clean data and $\tau = 1$ for noise, we interpolate future canvas latents as $\mathbf{z}_\tau = (1-\tau)\mathbf{z} + \tau\epsilon$, with $\epsilon \sim \mathcal{N}(0, I)$ and target velocity $\mathbf{u}_\tau = \epsilon - \mathbf{z}$. The objective is

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{\mathbf{z},\epsilon,\tau}\left[\|v_\theta(\widetilde{\mathbf{z}}_\tau, \tau; \mathbf{r}, \mathbf{z}^{\text{anc}}) - \mathbf{u}_\tau\|_2^2\right]. \tag{5}$$

The loss supervises future tokens across the generated modalities; observed reference tokens remain conditioning inputs.

3.3 Consistency-Aware Post-Training

Closed-loop simulation requires a reliable and consistent world model, beyond pixel-level similarity to observed trajectories. Pixel-space supervision is insufficient to enforce these requirements; we therefore introduce reward-based post-training to improve scene, state, and modality consistency.

3.3.1 Reward Models

We use a separate lightweight reward model for each consistency criterion. Given criterion-specific input $\mathbf{u}_q = h_q(\mathbf{x}_0, \mathbf{x}_{t+h})$, a ResNet-18 encoder ϕ_q and decoder D_q produce a logit and score:

$$R_q(\mathbf{u}_q) = D_q(\phi_q(\mathbf{u}_q)), \qquad \rho_q(\mathbf{u}_q) = \sigma(R_q(\mathbf{u}_q)), \tag{6}$$

where D_q may be an MLP or Transformer decoder and $\rho_q \in [0, 1]$. Each model is trained beforehand and remains frozen while supervising the generator.

Scene consistency. The Scene Reward Model takes an episode-initial RGB reference and a future RGB frame, assigning (1) to preserved scene identity and objects under valid motion and (0) to unintended changes. For training, we construct a class-balanced dataset using ground-truth observations and scene-consistent world-model generations as positive examples, and scene-inconsistent generations as negative examples. Each example is paired with its episode-initial RGB reference.

State consistency. The State Reward Model takes a future RGB observation and predicts whether the task's success condition holds (1) or remains unsatisfied (0). Its judgments provide labels for the canvas state region, whose color identifies the task and texture encodes completion. We train it exclusively on real trajectories, which naturally contain abundant negative examples: frames before task completion are labeled (0), while frames from completion onward are labeled (1).

Modality consistency. The Modality Reward Model takes paired RGB, tactile-image, and marker-displacement regions, assigning (1) when their contact evidence is compatible and (0) when it is mismatched. Negative training pairs are constructed by keeping the RGB observation fixed and replacing the paired tactile observations with a different tactile state, such as one with a different contact texture. This supervises cross-modal agreement rather than the presence of contact.

3.3.2 REWARD-GUIDED WORLD-MODEL POST-TRAINING

Reward supervision operates in image space. At a sampled diffusion time τ, we recover a clean canvas estimate from the predicted velocity and decode it with the video VAE:

$$\widehat{\mathbf{z}} = \mathbf{z}_\tau - \tau v_\theta(\widetilde{\mathbf{z}}_\tau, \tau; \mathbf{r}, \mathbf{z}^{\text{anc}}), \qquad \widehat{\mathbf{x}} = D\big(P^{-1}(\widehat{\mathbf{z}})\big)\,, \tag{7}$$

where P^{-1} restores the token sequence to the VAE latent grid and D is the decoder.

State supervision. The world model should reproduce the observed task outcome, whether successful or unsuccessful. The frozen State Reward Model evaluates the generated RGB observation and produces a success logit $\widehat{\ell}_{\text{state}}$. We obtain the binary target y_{state} by thresholding the same model's success score on the corresponding ground-truth RGB observation. We then minimize binary cross-entropy with logit temperature T_{state}:

$$\mathcal{L}_{\text{state}} = \mathbb{E}\Big[\text{BCEWithLogits}\Big(\widehat{\ell}_{\text{state}}/T_{\text{state}},\ y_{\text{state}}\Big)\Big]\,. \tag{8}$$

This preserves both successful and unsuccessful outcomes rather than forcing every prediction toward success.

Scene and modality supervision. Unlike task success, scene and cross-modal consistency are desirable for every prediction. Following the direct reward-maximization objective of Align-A-Video (Wang et al., 2025), we optimize the decoded canvas to increase the scores assigned by the frozen reward models:

$$\mathcal{L}_q = -\mathbb{E}[\rho_q(h_q(\mathbf{x}_0, \widehat{\mathbf{x}}))]\,, \qquad q \in \{\text{scene}, \text{modality}\}. \tag{9}$$

Here h_q selects the inputs required by each reward model, and ρ_q is its consistency score in $[0, 1]$. Gradients propagate through the frozen reward models and VAE decoder to update the world model. During reward-guided post-training, we retain the first-stage flow-matching MSE loss to preserve multimodal generation fidelity.

3.4 POLICY EVALUATION AND POLICY POST-TRAINING

Policy evaluation. PHYSCANVAS provides a multimodal environment for evaluating robot policies without executing every trial on hardware. Starting from a real initial observation, each policy is rolled out in the world model in a closed loop: the policy consumes the generated observation and issues the next action, and the world model returns its predicted visual, tactile, and task-state consequences. Repeating this process produces imagined policy trajectories under controlled initial conditions. We assess simulator reliability by comparing the success rates of these world-model rollouts with those of the corresponding real-world policy executions under the same success criteria.

Policy post-training. Beyond evaluation, PHYSCANVAS supports policy post-training without collecting additional robot trajectories. From real initial observations, a policy explores alternative behaviors through imagined closed-loop interactions, while the frozen State Reward Model scores their task outcomes. Group-relative policy optimization (Shao et al., 2024) converts differences among these outcomes into learning signals for the policy. Throughout this process, the world model and reward model remain fixed, so only the policy is updated from generated experience.

4 EXPERIMENTS

We evaluate action-conditioned multimodal prediction, the effect of consistency-aware post-training, and the use of imagined rollouts for policy evaluation and learning.

4.1 EXPERIMENTAL SETUP

Platform and data. We use an ARX AC One dual-arm robot with three cameras and four fingertip Xense tactile sensors (Appendix A). Our study covers four tasks—test-tube placement, bottle-cap unscrewing, block relocation, and Rubik's Cube manipulation—spanning handover, constrained contact, and object rotation.

Table 2: **RGB comparison.** Visual metrics (left), consistency scores and tracking errors (right).

Model	Conventional RGB metrics					VBench-derived / control diagnostics				
	PSNR↑	SSIM↑	LPIPS↓	FID↓	FVD↓	Subject↑	Bg.↑	Smooth↑	EEF (px)↓	Object (px)↓
EVAC	16.439	0.528	0.258	26.512	165.705	79.223	90.456	97.731	1.964	2.150
Open-Sora	17.916	0.599	0.228	34.762	183.860	78.519	90.234	98.157	1.425	1.426
WoVR	17.278	0.5972	0.203	26.086	138.243	80.247	**91.082**	98.059	2.792	1.164
PHYSCANVAS	**21.172**	**0.735**	**0.121**	**21.459**	**116.274**	**80.349**	90.565	**98.301**	**0.685**	**1.150**

Models and training. We initialize PHYSCANVAS from Wan 2.2 and train the unified canvas on mixed-task clips. Training uses four recursive chunks of eight future frames each. Unless otherwise specified, models are trained for 300 epochs. Reward models use ResNet-18 encoders with criterion-specific decoders. We evaluate ACT, FM, and $\pi_{0.5}$ with visual and tactile inputs, and post-train FM and $\pi_{0.5}$.

Baselines and evaluation. We compare with EVAC (Jiang et al., 2025), Open-Sora (Zheng et al., 2024), and WoVR (Jiang et al., 2026). RGB prediction is evaluated with standard reconstruction and distributional metrics (PSNR, SSIM, LPIPS, FID, and FVD), VBench consistency scores (Subject, Bg., and Smooth) (Huang et al., 2024), and EEF/object tracking errors. For touch, we report tactile-image PSNR, SSIM, LPIPS, Contact MAE, Onset/Offset, and Event F1, together with tactile-flow Active MAE, Idle FP-MAE, and Amp. MAE. Appendix H gives the definitions and aggregation rules.

4.2 COMPARISON WITH BASELINES

Quantitative comparison. Table 2 summarizes RGB prediction results for PHYSCANVAS, EVAC, Open-Sora, and WoVR. Reconstruction metrics are complemented by temporal-consistency scores and tracking errors to assess visual fidelity and interaction geometry. The PHYSCANVAS results are favorable across most visual metrics in Table 2. Its LPIPS of 0.121 indicates close agreement with ground truth in learned perceptual features, complemented by an FVD of 116.274 and high subject-consistency and motion-smoothness scores. Action responsiveness is examined separately in Figure 4: forcing the gripper open changes the predicted continuation from holding to dropping the block. This qualitative response illustrates the role of action conditioning without conflating perceptual similarity with action-following accuracy.

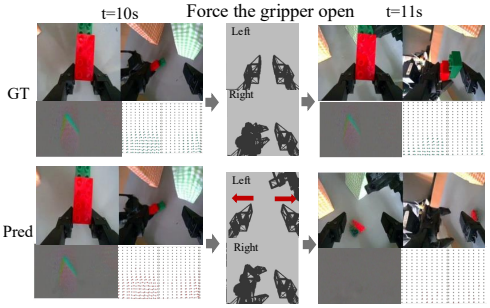


Figure 4: **Action intervention.** Forced-open gripper predicts a falling block.

4.3 ABLATION STUDIES

Table 3 compares two action-injection schemes and three consistency objectives. $\mathcal{M}_0$ injects a global motion embedding with the diffusion timestep through AdaLN (Peebles & Xie, 2023), whereas $\mathcal{M}_1$ uses view-aligned motion images fused position-wise with canvas tokens. Subsequent variants add scene, state, and multimodal consistency supervision. Compared with $\mathcal{M}_0$, spatially aligned $\mathcal{M}_1$ improves RGB PSNR/SSIM by 23.6%/18.3% and lowers LPIPS, FVD, and EEF error by 58.3%, 21.7%, and 32.6%. Tactile PSNR, LPIPS, Contact MAE, and marker-displacement MAEs improve, while a few other metrics worsen. Scene consistency ($\mathcal{M}_1 \rightarrow \mathcal{M}_2$) accompanies a 14.2% reduction in RGB LPIPS and a 10.4% reduction in FVD, with higher subject and background consistency, matching the goal of preserving scene appearance. With state supervision, $\mathcal{M}_3$ further reduces FID by 26.4% and FVD by 12.5%, while SSIM increases from 0.693 to 0.720. Multimodal consistency ($\mathcal{M}_3 \rightarrow \mathcal{M}_4$) lowers Contact MAE (0.02491 to 0.02278) and all three marker-displacement MAEs (Active: 4.666 to 4.083; Idle FP: 0.646 to 0.577; amplitude: 7.854 to 6.654),

while Event F1 dips from 0.4550 to 0.4455. These metrics assess tactile fidelity; Figure 5 (right) shows paired visual–tactile results.

Table 3: **Multimodal ablations.** $\mathcal{M}_0$: AdaLN action injection; $\mathcal{M}_1$: spatial motion-image injection; $\mathcal{M}_2$: $\mathcal{M}_1$ + scene consistency; $\mathcal{M}_3$: $\mathcal{M}_2$ + state supervision; $\mathcal{M}_4$: $\mathcal{M}_3$ + multimodal consistency.

RGB

Model	Reconstruction					VBench scores			Tracking error	
	PSNR↑	SSIM↑	LPIPS↓	FID↓	FVD↓	Subject↑	Bg.↑	Smooth↑	EEF (px)↓	Object (px)↓
$\mathcal{M}_0$	16.070	0.558	0.271	22.331	197.779	79.513	89.735	**98.588**	1.201	**1.017**
$\mathcal{M}_1$	19.869	0.660	0.113	33.145	154.901	80.272	90.347	98.333	0.810	1.651
$\mathcal{M}_2$	20.975	0.693	**0.097**	30.719	138.761	**80.544**	90.526	98.312	0.719	1.527
$\mathcal{M}_3$	20.699	0.720	0.132	22.619	121.484	80.220	90.549	98.334	0.783	1.323
$\mathcal{M}_4$	**21.172**	**0.735**	0.121	**21.459**	**116.274**	80.349	**90.565**	98.301	**0.685**	1.150

Tactile

Model	Tactile Image							Marker Disp. (MAEs $\times 10^3$)		
	PSNR↑	SSIM↑	LPIPS↓	Contact MAE↓	Onset (s)↓	Offset (s)↓	F1↑	Active MAE↓	Idle FP-MAE↓	Amp. MAE↓
$\mathcal{M}_0$	36.059	0.9330	0.1550	0.03012	1.638	1.241	0.4970	5.313	1.135	8.433
$\mathcal{M}_1$	36.292	0.9312	0.1265	0.02609	1.182	0.987	0.3498	4.834	0.900	7.634
$\mathcal{M}_2$	37.153	0.9349	0.1139	0.02458	**0.663**	0.900	**0.5335**	4.452	0.684	7.377
$\mathcal{M}_3$	36.921	0.9326	0.1221	0.02491	0.718	0.881	0.4550	4.666	0.646	7.854
$\mathcal{M}_4$	**37.598**	**0.9390**	**0.1032**	**0.02278**	0.664	**0.604**	0.4455	**4.083**	**0.577**	**6.654**

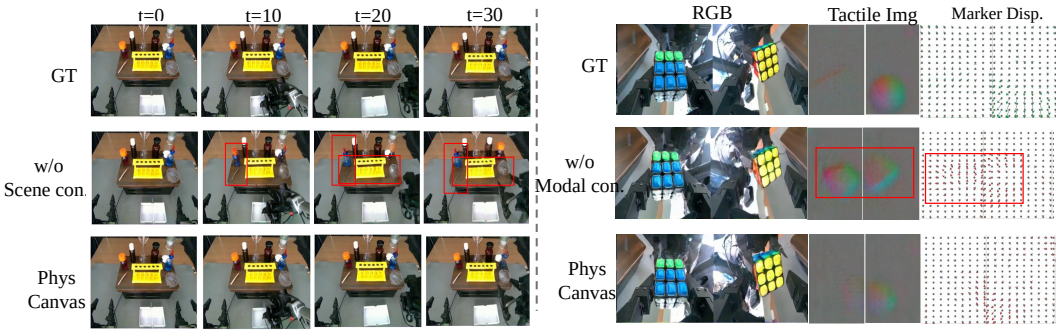


Figure 5: **Scene and modality consistency.** Left: temporal drift without Scene Reward Model supervision. Right: Rubik's Cube RGB, tactile image, and marker displacement before and after Modality Reward Model supervision; red boxes mark discrepancies.

Visual diagnostics. Figures 4 and 5 examine action response and consistency. In Figure 4, forced gripper opening makes the block fall and the tactile response disappear. Figure 5 (left) shows progressive scene drift without Scene Reward Model supervision, while the full model preserves the layout. On Rubik's Cube rotation (right), Modality Reward Model supervision suppresses diffuse tactile activity and retains the localized response. These examples are qualitative rather than contact-accuracy measurements.

4.4 PhysCanvas as a Policy Evaluator

To evaluate the effect of state supervision on policy evaluation, we roll out ACT, flow matching (FM), and $\pi_{0.5}$ in both M_2 and M_3 across four tasks. For each task–policy pair, evaluation uses three batches of 20 scenes collected at different times. Multiple human experts jointly judge whether each real-world scene satisfies the task-specific success criterion. Figure 6 compares the success rates of these world-model rollouts with the corresponding real-world success rates; stronger correlation indicates more reliable estimates across evaluation batches.

As shown in Figure 6, adding state supervision from M_2 to M_3 improves agreement between world-model rollout and real-world policy outcomes, raising the mean task-wise Pearson correlation from

Table 4: **Real-world policy success rates (%).** Values are mean ± population standard deviation over three runs; bold marks the highest nonzero mean per column (including ties).

Method	Test Tube		Bottle		Blocks		Rubik's Cube	
	FM	$\pi_{0.5}$	FM	$\pi_{0.5}$	FM	$\pi_{0.5}$	FM	$\pi_{0.5}$
SFT	0.0 ± 0.0	50.0 ± 7.0	20.0 ± 4.0	88.3 ± 4.7	25.0 ± 10.8	46.6 ± 2.3	0.0 ± 0.0	51.6 ± 6.2
RL in WM	0.0 ± 0.0	**60.0 ± 4.0**	33.3 ± 2.3	**90.0 ± 0.0**	33.3 ± 8.5	56.6 ± 4.7	0.0 ± 0.0	63.3 ± 4.7
RL in WM+RM	0.0 ± 0.0	**60.0 ± 8.1**	**35.0 ± 4.0**	86.6 ± 2.3	**41.6 ± 6.2**	**63.3 ± 4.7**	0.0 ± 0.0	**73.3 ± 4.7**

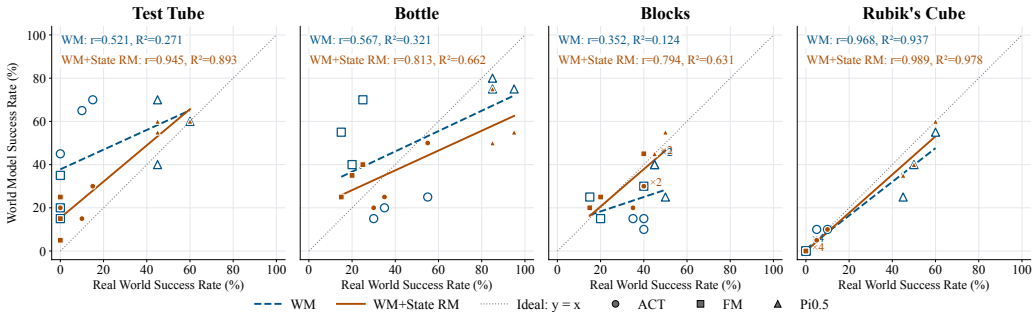


Figure 6: **Effect of state supervision on policy evaluation.** Each point compares world-model rollout and real-world success rates for one evaluation batch. Dashed blue and solid orange lines are task-wise fits for M_2 and M_3, respectively; the dotted diagonal denotes ideal agreement.

0.602 to 0.885 and the mean R^2 from 0.413 to 0.791. Gains are largest on Test Tube and Blocks, moderate on Bottle, and minimal on Rubik's Cube, where M_2 is already near ceiling. Since Table 3 shows mixed changes in low-level reconstruction metrics, the outcome gains suggest better task-semantic modeling rather than improved pixel-level reconstruction.

4.5 PhysCanvas as a Learning Simulator for Policy Improvement

We next ask whether interaction with a visuo-tactile world model can improve policies on the real robot. Starting from supervised fine-tuning (SFT), we optimize FM and $\pi_{0.5}$ with GRPO in PhysCanvas and evaluate the resulting policies on real tasks. In Table 4, RL in WM uses $\mathcal{M}_1$ as the simulator, whereas RL in WM+RM uses the consistency-post-trained $\mathcal{M}_4$; both use the frozen State Reward Model to score task outcomes. We omit ACT because it does not use the diffusion or flow-matching formulation assumed by our GRPO implementation.

RL in $\mathcal{M}_1$ raises mean real-world success from 11.2% to 16.6% for FM and from 59.1% to 67.5% for $\pi_{0.5}$, averaging equally across the four tasks. Using $\mathcal{M}_4$ as the simulator further raises these means to 19.1% and 70.8%. For FM, gains occur on Bottle and Blocks, while Test Tube and Rubik's Cube remain at zero. For $\pi_{0.5}$, $\mathcal{M}_4$ yields its largest additional gains over $\mathcal{M}_1$ on Blocks and Rubik's Cube (6.6 and 10.0 percentage points); Bottle decreases from 90.0% to 86.6%. Thus, imagined interactions in the consistency-post-trained world model improve average real-world success, although the benefit varies by task.

5 Conclusion

We introduced PhysCanvas, a state-consistent visuo-tactile world model that predicts the visual, tactile, and task-level consequences of robot actions. Its unified canvas, spatially aligned action conditioning, and consistency-aware rewards support policy evaluation and improvement through imagined interactions, highlighting the value of joint multimodal modeling for decision-relevant robot world models.

AI USE STATEMENT

Generative AI tools assisted literature discovery, language editing, and early-draft organization. The authors reviewed and verified all technical content, citations, claims, code, and generated text, and take responsibility for the final manuscript.

REFERENCES

Jai Bardhan, Patrik Drozdik, Josef Sivic, and Vladimir Petrik. Persistent robot world models: Stabilizing multi-step rollouts via reinforcement learning. In *European Conference on Computer Vision*, 2026. URL https://arxiv.org/abs/2603.25685.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Bi_Motus_A_Unified_Latent_Action_World_Model_CVPR_2026_paper.html.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, brian ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 2025. URL https://proceedings.mlr.press/v305/black25a.html.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://arxiv.org/abs/2304.08818.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C.Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando De Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623. PMLR, 2024. URL https://proceedings.mlr.press/v235/bruce24a.html.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *IEEE International Conference on Robotics and Automation*, 2017.

GigaWorld Team, Angyuan Ma, Boyuan Wang, Bohan Li, Chaojun Ni, et al. GigaWorld-1: A roadmap to build world models for robot policy evaluation. *arXiv preprint arXiv:2607.02642*, 2026.

Irmak Guzey, Yinlong Dai, Ben Evans, Soumith Chintala, and Lerrel Pinto. See to touch: Learning tactile dexterity through visual incentives. In *2024 IEEE International Conference on Robotics and Automation*, pp. 13825–13832, 2024. doi: 10.1109/ICRA57147.2024.10611407. URL https://ieeexplore.ieee.org/document/10611407/.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://papers.nips.cc/paper/2018/hash/2de5d16682c3c35007e4e92982f1a2ba-Abstract.html.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1912.01603.

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

Nick Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/cf73d57b6dcda32b293df7c2d5341f49-Abstract-Conference.html.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2204.03458.

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*, pp. 21807–21818, 2024.

Yuxin Jiang, Shengcong Chen, Siyuan Huang, Liliang Chen, Pengfei Zhou, Yue Liao, Xindong He, Chiming Liu, Hongsheng Li, Maoqing Yao, and Guanghui Ren. EnerVerse-AC: Envisioning embodied environments with action condition. *arXiv preprint arXiv:2505.09723*, 2025.

Zhennan Jiang, Shangqing Zhou, Yutong Jiang, Zefang Huang, Mingjie Wei, Yuhui Chen, Tianxing Zhou, Zhen Guo, Hao Lin, Quanlu Zhang, Yu Wang, Haoran Li, Chao Yu, and Dongbin Zhao. WoVR: World models as reliable simulators for post-training VLA policies with RL. *arXiv preprint arXiv:2602.13977*, 2026.

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 *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/748becc400a57c0e31cfe6a2e7951467-Abstract-Conference.html.

Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, Dinesh Jayaraman, and Roberto Calandra. DIGIT: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, 2020. URL https://arxiv.org/abs/2005.14679.

Michelle A. Lee, Yuke Zhu, Peter Zachares, Matthew Tan, Krishnan Srinivasan, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Learning multimodal representations for contact-rich tasks. *IEEE Transactions on Robotics*, 36(3), 2020. URL https://arxiv.org/abs/1907.13098.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Shengbang Liu, Yueru Jia, Yuyang Yan, Jiaming Liu, Xinran Zhang, Qiuxuan Feng, Yandong Guo, Shiji Zhou, Boxin Shi, and Shanghang Zhang. TACO: TActile world model as a Self-COrrector for scalable robot policy post-training. *arXiv preprint arXiv:2607.02840*, 2026. URL https://arxiv.org/abs/2607.02840.

Wenxuan Ma, Chaofan Zhang, Chao Xue, Yinghao Cai, Guocai Yao, Shaowei Cui, and Shuo Wang. FeelWorld: Visuo-tactile world model for hierarchical contact prediction and planning. *arXiv preprint arXiv:2607.24267*, 2026.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2209.00588.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. URL https://arxiv.org/abs/2212.09748.

Yan Qin, Yue Chen, Wenwei Lin, Shujia Liu, Chuqiao Lyu, Kailun Su, Weiyang Jin, Chenze Yu, Ping Luo, Wenbo Ding, Tianxing Chen, and Renjing Xu. DexTouch-WM: Learning action-conditioned tactile world models from human touch for dexterous robot manipulation. *arXiv preprint arXiv:2609.20649*, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Wan Team, 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.

Shengzhi Wang, Yingkang Zhong, Jiangchuan Mu, Kai Wu, Mingliang Xiong, Wen Fang, Mingqing Liu, Hao Deng, Bin He, Gang Li, and Qingwen Liu. Align-A-Video: Deterministic reward tuning of image diffusion models for consistent video editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2074–2083, 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. *arXiv preprint arXiv:2603.08546*, 2026. URL https://arxiv.org/abs/2603.08546.

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

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 of *Proceedings of Machine Learning Research*, pp. 2226–2240. PMLR, 2023. URL https://proceedings.mlr.press/v205/wu23c.html.

Zhenyu Wu, Xiuwei Xu, Yukun Zhou, Yifan Li, Qiuping Deng, Xiaofeng Wang, Zheng Zhu, Bingyao Yu, Ziwei Wang, Jiwen Lu, and Haibin Yan. iMaC: Translating actions into motion and contact images for embodied world models. *arXiv preprint arXiv:2606.09813*, 2026.

Jiazhi Yang, Kunyang Lin, Wencong Zhang, Jinwei Li, Tianwei Lin, Longyan Wu, Ya-Qin Zhang, Hao Zhao, Ping Luo, Zhizhong Su, Hongyang Li, Xiangyu Yue, and Li Chen. Self-improving robot policy with compositional world model. In *Proceedings of Robotics: Science and Systems*, 2026a. doi: 10.15607/RSS.2026.XXII.012. URL https://www.roboticsproceedings.org/rss22/p012.html.

Sherry Yang, Yilun Du, Seyed Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/c4d66eae503694424123b93ac0fbaf17-Abstract-Conference.html.

Tianzhuo Yang, Zihan Shen, Zirui Mi, Zhaoyi Zhang, Jiayi Zhou, Jiaming Ji, Juntao Dai, Jiawei Chen, Boyuan Chen, and Yaodong Yang. MiraBench: Evaluating action-conditioned reliability in robotic world models. *arXiv preprint arXiv:2605.29360*, 2026b. URL https://arxiv.org/abs/2605.29360.

Chao Yu, Yuanqing Wang, Zhen Guo, Hao Lin, Si Xu, Hongzhi Zang, Quanlu Zhang, Yongji Wu, Chunyang Zhu, Junhao Hu, Zixiao Huang, Mingjie Wei, Yuqing Xie, Ke Yang, Bo Dai, Zhexuan Xu, Jiakun Du, Xiangyuan Wang, Xu Fu, Letong Shi, Zhihao Liu, Kang Chen, Weilin Liu, Gang Liu, Boxun Li, Jianlei Yang, Zhi Yang, Guohao Dai, and Yu Wang. RLinf: Flexible and efficient Large-Scale reinforcement learning via Macro-to-Micro flow transformation. In *20th USENIX Symposium on Operating Systems Design and Implementation (OSDI 26)*, pp. 829–846. USENIX Association, 2026. URL https://www.usenix.org/conference/osdi26/presentation/yu-chao.

Haoran Yuan, Zekai Wang, Boning Shao, Haoran Lu, Trevor Darrell, Ismini Lourentzou, and Wei Zhan. DexTacWAM: A visuo-tactile world-action model for dexterous manipulation. *arXiv preprint arXiv:2609.24976*, 2026. URL https://arxiv.org/abs/2609.24976.

Wenzhen Yuan, Siyuan Dong, and Edward H. Adelson. GelSight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017. doi: 10.3390/s17122762.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems*, 2023.

Yuhang Zheng, Songen Gu, Yupeng Zheng, Weize Li, Yujie Zang, Shuai Tian, Xiang Li, Ce Hao, Chen Gao, Si Liu, Haoran Li, Yilun Chen, Shuicheng Yan, and Wenchao Ding. OmniVTA: Visuo-tactile world modeling for contact-rich robotic manipulation. *arXiv preprint arXiv:2603.19201*, 2026.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-Sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

APPENDIX

A Hardware Configuration **14**

B Bimanual Tasks and Data **15**

B.1 Task design and object selection 15

B.2 Data scale and tactile representation 16

C World Model Training and Inference **17**

C.1 World-model data collection 17

C.2 Training configuration 18

C.3 Inference and recursive rollout 19

D Policy Architectures and Tactile Integration **19**

D.1 Common observation and control interface 19

D.2 Flow-matching policy 19

D.3 ACT 20

D.4 Tactile-augmented $\pi_{0.5}$ 20

E Reward Model Training **20**

E.1 Roles and Supervision 20

E.2 Network Architecture 21

E.3 Training Configuration 21

F Policy–World Model Interaction Rollout **21**

F.1 Closed-loop interaction 21

F.2 Rollout configuration and execution 22

G Policy Reinforcement Learning in the World Model **22**

G.1 RL in World Model 22

G.2 Groups, rewards, and objective 23

H Metric Computation **24**

I Visualization Analysis **26**

I.1 Qualitative Comparison with Baselines 26

I.2 Layer-Wise Multimodal Attention Analysis 26

A HARDWARE CONFIGURATION

Our experiments use an ARX AC One dual-arm platform with a tabletop workspace. Each arm has six joints and a gripper. Three cameras provide a fixed base view and two wrist views, while four Xense optical tactile sensors mounted on the gripper fingers provide contact observations. Visual, tactile, and robot-state observations are synchronized at 10 Hz.

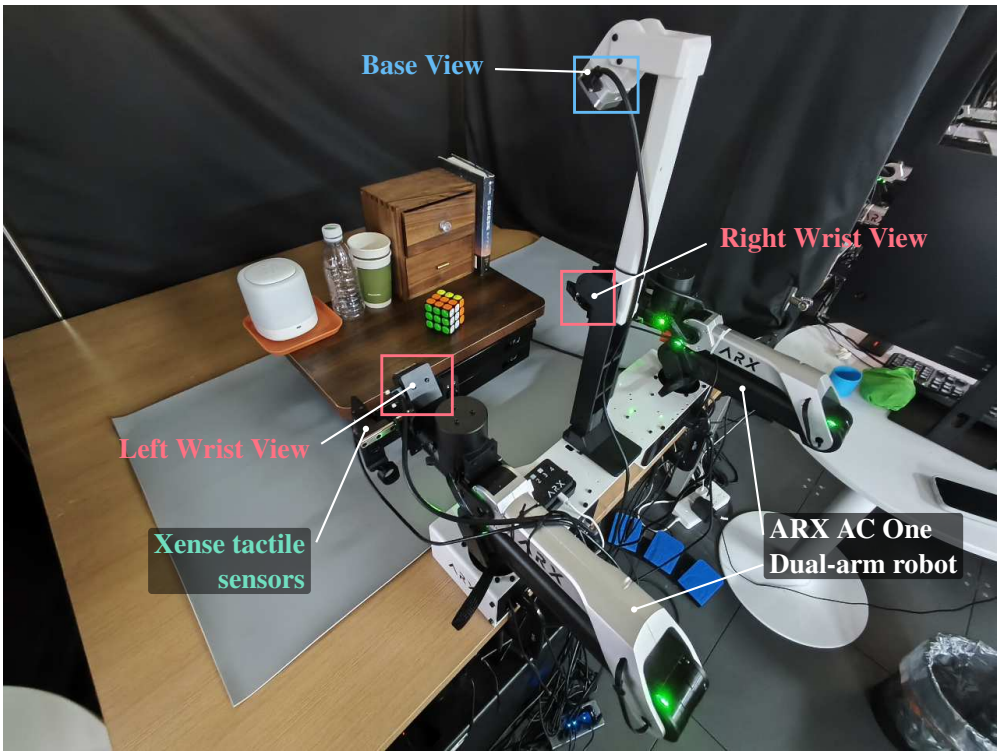


Figure 7: **Hardware layout.** The dual-arm platform, three camera views, and fingertip tactile sensors.

Optical tactile sensing. Xense sensors capture local contact appearance and surface deformation at the fingertips. We use tactile images and marker-displacement fields as complementary observations of the interaction. The photographs below show the sensor's contact-facing side, rear connector and mounting holes, and installation on a gripper finger.



(a) Contact-facing side (b) Rear and connector (c) Gripper installation

Figure 8: **Xense optical tactile sensors.** Sensor views and gripper mounting.

B BIMANUAL TASKS AND DATA

B.1 TASK DESIGN AND OBJECT SELECTION

All four tasks require coordinated use of both arms. We select objects that expose complementary contact phenomena: reorientation of a slender object, separation of interlocking parts, threaded rotation, and changing surface textures. Figure 9 pairs object photographs with robot rollouts from our data and synchronized tactile observations.

Test Tube. The left gripper picks up a test tube and passes it to the right gripper. The right gripper then uses contact with the table to reorient the tube in-hand before inserting it into the rack. The slender, transparent tube makes grasp stability and orientation difficult to infer from appearance alone, motivating tactile feedback during handover and reorientation.

Blocks. The left gripper lifts two connected blocks. Both grippers separate them and place each block into the bag of the matching color. Interlocking studs and edges create localized contact patterns, while separation produces a distinct change in contact that must agree with the visible motion.

Bottle. One gripper stabilizes the bottle while the other unscrews its cap; the cap and bottle are then placed separately on the table. The ridged cap and compliant bottle provide different contact geometries. This task couples rotational motion with sustained grasping and a final contact-release event.

Rubik's Cube. The robot restores a Rubik's Cube after a single face-turn perturbation. Its colored, textured faces change the local contact pattern during rotation, making it a challenging test of visual–tactile alignment. The restricted scramble isolates coordinated manipulation and sensory consistency rather than general Rubik's Cube solution planning.



Figure 9: **Four bimanual tasks.** Object photographs, rollout stages, and paired tactile-image/marker-displacement examples. L/R denotes the gripper and 0/1 its pad; green arrows show recorded displacement at $5\times$ display scale.

B.2 DATA SCALE AND TACTILE REPRESENTATION

Data. We use 3,805 real-robot policy rollouts for world-model training and 80 held-out test rollouts per task (Table 5).

Table 5: **Policy-rollout training and test data used by the world model.**

Task	Training episodes	Test episodes
Test Tube	1,000	80
Blocks	1,000	80
Bottle	926	80
Rubik's Cube	879	80
Total	3,805	320

Visual observation. Each timestamp contains three synchronized RGB views: one fixed base view and two wrist views. Observations are recorded at 10 Hz. Each view is cropped and resized to 224×224 pixels, then the three views are concatenated horizontally in a fixed order to form a 224×672 visual strip.

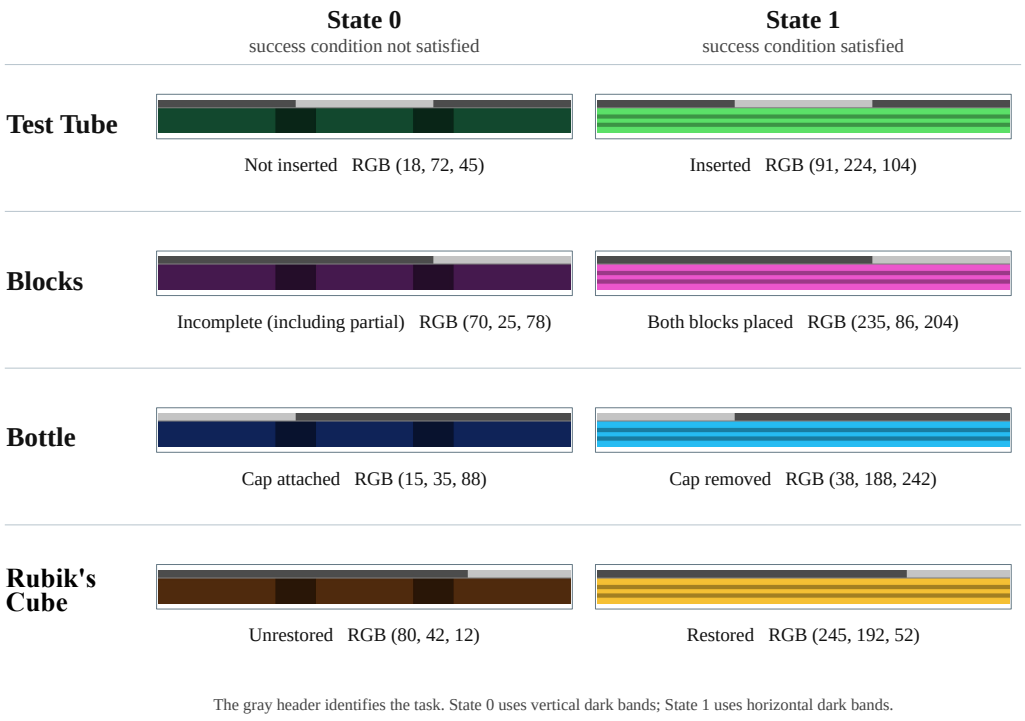


Figure 10: **Task-state canvas encoding.** Each row shows state 0 (the success condition is not satisfied) and state 1 (the success condition is satisfied) for one task. Task-specific color identifies the task, while vertical and horizontal low-frequency bands distinguish states 0 and 1, respectively. For Blocks, partial completion remains state 0; state 1 requires both blocks to be placed.

Tactile image. We use the Xense internal camera's recorded difference image relative to a sensor reference, not consecutive-frame differences or external scene RGB. Preprocessing decodes and time-aligns these $400 \times 700 \times 3$ unsigned 8-bit images. Display rotates each pad 90° clockwise; the world model does not re-estimate the SDK's reference or intensity mapping.

Marker displacement. Each pad stores a 35×20 grid of 700 tracked markers. For marker i, the first frame of an episode defines the reference state $\mathbf{q}_{0,i}$. At time t, we subtract this reference

from the recorded three-component marker state $\mathbf{q}_{t,i}$ to obtain $\Delta\mathbf{q}_{t,i} = (\Delta x, \Delta y, \Delta z)$. A stationary marker is therefore centered at zero, whereas contact-induced translation or deformation produces a signed three-dimensional offset. To place this nonvisual field on the shared RGB canvas, we clip each component to $[-s, s]$, normalize it to $[0, 1]$, and assign x, y, and z to the R, G, and B channels, respectively:

$$\Delta\mathbf{q}_{t,i} = \mathbf{q}_{t,i} - \mathbf{q}_{0,i}, \qquad \mathbf{c}_{t,i} = \tfrac{1}{2}\left[\mathrm{clip}(\Delta\mathbf{q}_{t,i}/s, -1, 1) + 1\right]. \tag{10}$$

The audited configurations use $s = 0.06$ in cache-native units; this is an encoding scale rather than an observed physical bound. The RGB image is only an image-native representation for the model: quantitative statistics are computed directly from the unencoded $\Delta\mathbf{q}$. For vector visualization, the initial planar marker locations provide the arrow origins and $(\Delta x, \Delta y)$ gives the sensor-plane arrow direction; Δz remains part of the three-dimensional field and its RGB encoding. Only the displayed grid is subsampled to 18×10, while statistics retain all 700 markers.

Action representation. The action condition is a numerical sequence of arm-joint positions and gripper states. At each timestamp, we apply it to the robot URDF, recover the articulated meshes through forward kinematics, and project them into the fixed and wrist cameras using calibrated intrinsics and extrinsics. Each render uses the same crop, resize, and view order as RGB, producing a spatially aligned action-image sequence. The shared video VAE encodes these images before position-wise fusion with the noisy world-model latents. The overlay at right visualizes registration and is not a prediction target.



Figure 11: **Action projection.** Projected URDF wireframes over the fixed and wrist views.

Tactile ranges. We summarize tactile-image intensities and marker-displacement magnitudes from sampled rollouts across all four tasks. The P95 displacement norms are 0.0043, 0.0161, 0.0234, and 0.0266 for Test Tube, Blocks, Bottle, and Rubik's Cube, respectively, in cache-native units (Figure 12).

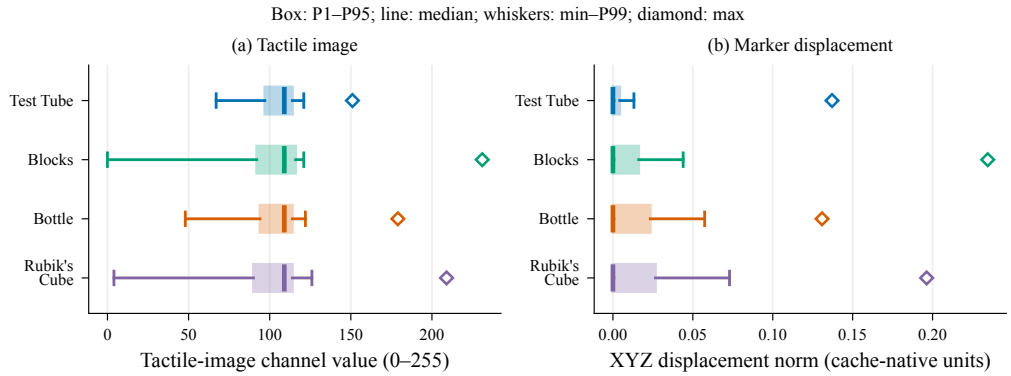


Figure 12: **Tactile data distributions.** Unified percentile-box summaries from sampled rollouts.

C World Model Training and Inference

C.1 World-model data collection

To expose the world model to a broad action distribution, we execute stochastic rollouts of task policies on the real robot. The resulting data contain successful executions as well as naturally occurring failures, including missed grasps, premature release, inaccurate handover, misalignment, and incomplete insertion or placement. This diversity is important because a policy evaluator must model not only expert demonstrations but also the off-nominal states visited by imperfect policies.

Figure 13 summarizes the data loop. Robot demonstrations are first converted to the LeRobot format and used to train task policies. Stochastic policy execution then provides rollouts for world-model learning. After collection, these rollouts undergo offline multimodal preprocessing: (1) numerical

robot actions are rendered as view-aligned action images; (2) camera images are cropped and resized to the fixed world-model layout; and (3) each raw tactile record is separated into its tactile image and tracked marker positions, from which marker displacement is computed. We time-align these processed signals and assemble them into the unified canvas before world-model training. The trained world model subsequently interacts with the policy in closed loop, providing generated visual–tactile observations for either policy evaluation or RL-based policy improvement.

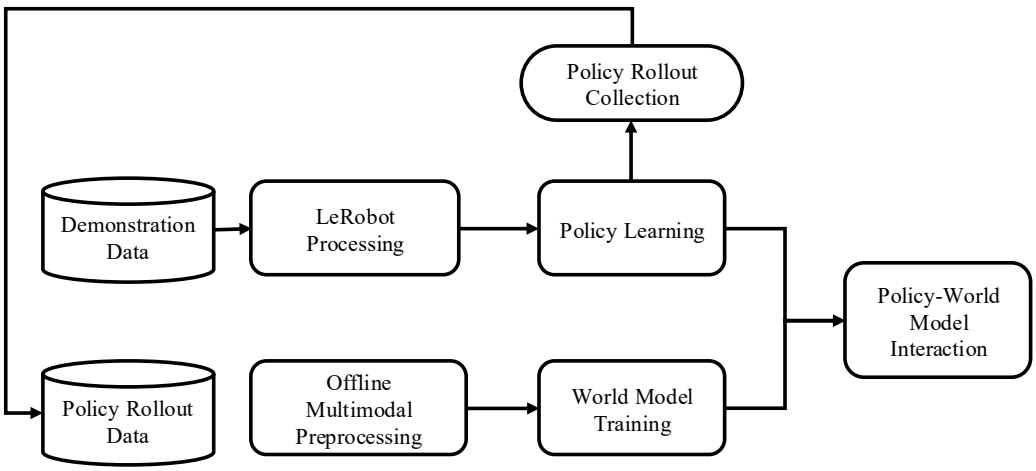


Figure 13: **World-model data and interaction loop.** Policy rollouts are processed offline into aligned action-image and visual–tactile training data. The trained world model then interacts with a policy for RL or evaluation.

C.2 Training configuration

Initialization and action conditioning. We initialize PhysCanvas from `Wan2.2-TI2V-5B`, a 5-billion-parameter latent video flow-matching transformer pretrained for text/image-to-video generation. We retain its visual VAE and spatiotemporal transformer, freeze the VAE, and fine-tune the transformer for multimodal future prediction. As described in Section 3.2, robot configurations are rendered into view-aligned action images using camera calibration and the robot URDF. Their latent tokens are fused position-wise with the noisy canvas tokens, converting the pretrained generator into an action-conditioned world model without a dedicated action encoder beyond the shared video VAE.

Optimization and compute. Training uses 64 PPU-ZW810E accelerators (four nodes with 16 devices each) for 300 epochs and takes approximately three days. We use the 8-bit CAME optimizer with a constant learning rate of $2^{-14.5} = 4.316 \times 10^{-5}$ and weight decay 10^{-2}. The per-device batch size is 2 with one gradient-accumulation step, giving an effective batch size of 128 clips. Training uses BF16, DeepSpeed ZeRO-2, and activation checkpointing of Wan transformer blocks. No explicit gradient clipping is applied.

Noise and rollout training. For each flow-matching target, we sample $u \sim \mathcal{U}(0,1)$ and use the shifted noise level

$$\sigma = \frac{5u}{1+4u}, \qquad \mathbf{z}_\sigma = (1-\sigma)\mathbf{z} + \sigma\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, I), \tag{11}$$

where σ is quantized to increments of 10^{-3}. Each training example retains the episode-initial three-view RGB anchor across four successive rollout chunks. A chunk predicts eight future frames; its terminal observation replaces only the current multimodal reference $\mathbf{r}$ for the next chunk. The loss is averaged across the four chunks, so the model is optimized under its own recursively generated context rather than only under ground-truth one-step conditioning.

C.3 Inference and recursive rollout

At inference, the same episode-initial RGB anchor is fixed across all chunks. Each chunk starts from Gaussian noise at $\sigma = 1$ and is denoised with 5 flow-matching steps to generate eight future multimodal frames; its last frame updates $\mathbf{r}$ for the next call. In interactive evaluation, the policy consumes the generated RGB and tactile observations and supplies the next action sequence; the action is rendered into the aligned conditioning canvas before the following chunk is generated. We repeat this process until the full episode horizon is reached, so rollout length is not limited to the four chunks used during training.

D Policy Architectures and Tactile Integration

We evaluate three policy families: Action Chunking with Transformers (ACT) (Zhao et al., 2023), a task-specific flow-matching policy (FM) (Lipman et al., 2023), and a tactile-augmented $\pi_{0.5}$ policy (Black et al., 2025). All three receive three synchronized RGB views, the current bimanual robot state, and four tactile pads, and output continuous dual-arm joint commands at 10 Hz. Because tactile-to-policy fusion is not the primary focus of this work, we deliberately adopt a simple modality-concatenation strategy rather than designing a specialized fusion architecture. The three policies nevertheless differ in their pretraining, action parameterization, and data sources. FM and ACT are trained separately for each task from successful real-robot executions, whereas $\pi_{0.5}$ shares one multitask checkpoint across language instructions. We therefore use them as deployed policy families rather than as a controlled architecture comparison.

D.1 Common observation and control interface

The RGB observation contains one base-camera view and two wrist-camera views, each resized to 224×224. FM and ACT encode each view with a frozen DINOv2-small backbone and use its 384-dimensional CLS feature. The tactile input uses marker displacement rather than the tactile image: for each of four Xense pads, 700 tracked markers form a 35×20 grid with an XYZ displacement vector relative to the episode reference. FM and ACT concatenate the four pads along the channel dimension into a $35 \times 20 \times 12$ tensor; $\pi_{0.5}$ retains four separate $35 \times 20 \times 3$ grids. Each policy observes only the current synchronized frame and the current 14-dimensional dual-arm joint state. The robot interface consumes the first 14 output dimensions as absolute joint targets.

Figure 14 summarizes how the same sensory interface is specialized by the three policies. FM compresses vision, touch, and state into one global condition for a one-dimensional U-Net velocity field. ACT exposes the modality features as Transformer memory tokens and decodes a complete action chunk with learned queries. $\pi_{0.5}$ appends four tactile tokens to its vision–language prefix and predicts actions with a Gemma action expert.

D.2 Flow-matching policy

The FM policy encodes the three RGB views with frozen DINOv2-small features and uses lightweight encoders for the 14D robot state and the concatenated tactile-displacement grid. The three modalities are fused into a 256D condition c_t. A conditional 1D U-Net then learns a flow field from Gaussian noise to a normalized relative 32×14 joint-action chunk using the standard flow-matching objective. At inference, we integrate the learned flow field with 32 Euler steps. The predicted chunk is unnormalized using the checkpoint action normalizer and converted, relative to the current robot state, into absolute 14D joint targets. All 32 actions are executed at 10 Hz before the policy replans.

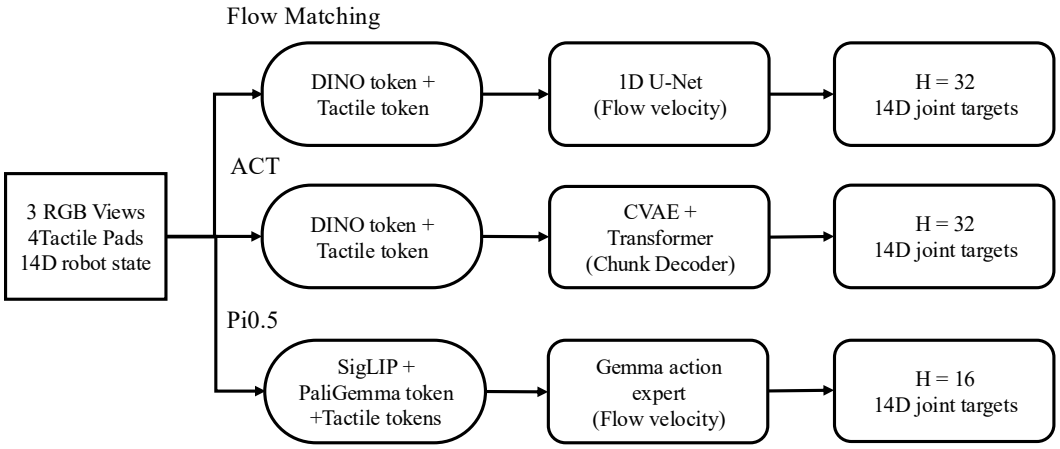


Figure 14: **Policy architecture overview.** All policies use synchronized multi-view RGB, marker displacement, and robot state, but fuse these modalities differently. FM and ACT are task-specific; $\pi_{0.5}$ is prompt-conditioned and shared across tasks.

D.3 ACT

Our ACT variant uses frozen DINOv2-small features for the three RGB views and compresses the four tactile grids into a single token. These modality tokens, together with the current robot state and a CVAE latent, condition a Transformer decoder with learned action queries. The model is trained to reconstruct normalized relative joint-action chunks with KL regularization on the CVAE latent. At deployment, the latent is fixed to zero and the complete action chunk is decoded in one forward pass. The predicted relative actions are unnormalized using the checkpoint action normalizer and converted, relative to the current robot state, into absolute 14D joint targets before execution. For all four tasks, ACT predicts and executes a 32-step action chunk at 10 Hz before replanning, without temporal ensembling.

D.4 Tactile-augmented $\pi_{0.5}$

The $\pi_{0.5}$ policy combines SigLIP visual tokens, a PaliGemma language–state context branch, and a Gemma action expert for flow-based generation. A shared CNN independently encodes each tactile pad into a token, and the four tactile tokens are appended to the language–vision prefix through a learned gate. The model predicts a 16-step action chunk in a 32D internal space, of which the first 14 dimensions control the robot. Deterministic evaluation and real-robot deployment use a ten-step Flow-ODE sampler from Gaussian noise to action. During GRPO rollout, one of the ten denoising steps is selected uniformly and replaced by a stochastic Flow-SDE transition, while the remaining steps use Flow-ODE integration.

E Reward Model Training

E.1 Roles and Supervision

We train reward models to provide supervision for world-model post-training and rewards for policy reinforcement learning inside the trained world model. Three models address complementary aspects of the generated interaction. The State Reward Model scores whether an RGB observation satisfies the task outcome; its scores guide world-model post-training and provide the binary reward for policy GRPO. The Scene Reward Model scores whether a generated frame preserves the initial scene while allowing intended robot and object motion. The Modality Reward Model scores whether visual and tactile observations describe a compatible physical state. The latter two provide scene and cross-modal supervision during world-model post-training.

All three models output a binary logit and are optimized with `BCEWithLogitsLoss`; a sigmoid is applied only when a score in $[0, 1]$ is needed. Table 6 summarizes their supervision. The

task-state labels come entirely from real robot rollouts: human annotators identify the transition into a successful state, frames before that transition are negative, and unsuccessful rollouts remain negative throughout. Scene-consistency supervision combines a real initial-frame reference with world-model-generated rollouts; annotators mark the first generated frame that no longer agrees with the reference scene. Visual–tactile supervision starts from synchronized real multimodal bundles: aligned observations are positive, while within-hand tactile swaps and tactile bundles drawn from another rollout create controlled negative examples. We also add world-model-generated observations judged to have mismatched visual and tactile content as negative examples. These sources expose the model to both constructed modality conflicts and generation failures.

Table 6: **Reward-model supervision.** "WM" denotes world-model-generated observations and "GT" denotes real observations.

Reward model	Input and question	Data provenance
State Reward Model	Three RGB views; is the task complete?	Human-labeled real rollouts with frame-level completion transitions.
Scene Reward Model	Initial GT and current WM views; is the scene preserved?	Human-labeled GT–WM pairs, including preserved and drifted frames.
Modality Reward Model	RGB, tactile image, and marker displacement; do modalities agree?	Aligned real bundles, controlled tactile mismatches, and labeled WM generation failures.

E.2 Network Architecture

Following the image-based reward-model design in RLinf Yu et al. (2026), we use ImageNet-initialized ResNet-18 encoders and a scalar scoring head. The State Reward Model shares one encoder across three RGB views and scores their concatenated features with a $1536 \to 256 \to 1$ MLP. The Scene Reward Model shares an encoder between the initial GT and current WM views; a two-layer Transformer decoder compares their 7×7 spatial tokens, using the reference as memory. The Modality Reward Model uses separate encoders for RGB, tactile image, and marker displacement, shared across views or pads within each modality. Its four-layer decoder lets visual tokens query tactile tokens. Both decoders use eight attention heads and a $512 \to 256 \to 1$ scoring head.

E.3 Training Configuration

We train all three reward models for 100 epochs with AdamW on PPU or H20 accelerators. The State Reward Model and Scene Reward Model use balanced positive and negative batches, whereas the Modality Reward Model retains the mixture of aligned observations, controlled tactile mismatches, and world-model-generated failures described above. We use a learning rate of 10^{-4} for task state and scene consistency and 2×10^{-4} for visual–tactile consistency. Data are split by rollout to keep observations from the same trajectory together. Once trained, the reward models are frozen during world-model post-training and policy GRPO.

F Policy–World Model Interaction Rollout

F.1 Closed-loop interaction

A policy rollout in PhysCanvas treats the world model as an action-driven virtual environment. A trajectory begins from one synchronized real observation, after which the policy predicts an action chunk from the current observation. We use a common task specification ξ: it identifies the task for task-specific FM and ACT, and includes the language instruction required by the shared $\pi_{0.5}$ policy. For all policy families, the corresponding task prompt is independently supplied to the text-conditioned world model. The world model executes each chunk in short segments and generates the corresponding future observations. Its latest prediction is converted back into the policy observation interface, closing the loop until the target horizon is reached. Except for the initial state, the policy therefore acts on generated rather than recorded future observations.

This interaction extends visual world-model rollout to the sensory interface required by contact-rich policies. In addition to multi-view RGB, the generated canvas carries tactile images, marker

displacement, and task-state information. The policy consumes its required subset of these synchronized signals together with the current robot state. Consequently, policy behavior can respond not only to visible geometry but also to predicted contact changes, while the same simulator provides a common interaction loop for policies with different action horizons.

F.2 Rollout configuration and execution

Algorithm 1 gives the closed-loop schedule. At 10 Hz, FM and ACT each predict an action chunk spanning $H = 32$ future time steps, while $\pi_{0.5}$ predicts $H = 16$. Each step supplies 14 dual-arm joint targets, so the executed chunk has shape $H \times 14$. The latter policy generates actions in a 32D internal space, but only the first 14 dimensions of each step are sent to the robot. The world model consumes the resulting 14D commands in four or two segments of eight actions, respectively.

Here $\mathrm{PolicyStep}(\pi, o, \xi)$ denotes $\pi_{\mathrm{task}}(o)$ for task-specific FM and ACT and $\pi(o, \ell)$ for prompt-conditioned $\pi_{0.5}$, where ℓ is the language component of ξ. The task prompt represented by ξ is also supplied to the world model's text-conditioning branch for every policy family.

Algorithm 1 Closed-loop rollout

$Y \leftarrow [\,],\ q \leftarrow q_0,\ r \leftarrow o_0$
repeat
  $o \leftarrow \mathrm{Adapt}(r, q)$
  $A \leftarrow \mathrm{First}_{T-|Y|}(\mathrm{PolicyStep}(\pi, o, \xi))$
  for each $S \in \mathrm{Split}_8(A)$ **do**
    $C \leftarrow \mathrm{Render}(q, S)$
    $\widehat{O} \leftarrow \mathrm{WM}(o_0^{\mathrm{cam}}, r, C, \xi)$
    $Y \leftarrow Y \,\|\, \widehat{O}$
    $q \leftarrow \mathrm{Last}(S)$
    $r \leftarrow \mathrm{Last}(\widehat{O})$
  end for
until $|Y| = T$
return Y

For each segment, we render the robot state and commands into view-aligned action images, then predict eight multimodal frames. The last frame and command update the reference and robot state. The policy replans only after the entire chunk has been executed.

The three RGB views in the real initial observation form the fixed episode anchor, while the full initial observation initializes the current multimodal reference r. Thereafter, only r advances with each world-model prediction. The observation adapter extracts the generated RGB and tactile signals required by the policy and attaches the updated robot state. For $\pi_{0.5}$, it additionally attaches the language instruction from ξ; task-specific FM and ACT receive no language input. Independently, the task prompt remains available to the world model's text-conditioning branch throughout the rollout. Repeating this cycle produces a full multimodal trajectory without conditioning on recorded future observations.

Here o_0 and q_0 are the synchronized real initial observation and robot state, o_0^{cam} is the fixed three-view RGB anchor, r is the current multimodal reference, ξ is the task specification, and T is the requested horizon. In the algorithm, Render creates the action images, WM generates the next segment, and Split_8 partitions a chunk into segments of at most eight commands. The final chunk is truncated at T.

G Policy Reinforcement Learning in the World Model

G.1 RL in World Model

We optimize a policy by treating the frozen world model as the interaction environment and the frozen State Reward Model as the outcome evaluator. Starting from one real reset observation, the policy samples multiple action trajectories. The world model converts each trajectory into a closed-loop multimodal rollout, and the State Reward Model evaluates the generated three-view RGB observations. Trajectory rewards are normalized only against alternatives generated from the same initial condition. The resulting group-relative advantages supervise a clipped policy update; gradients update the policy while the world model and State Reward Model remain fixed.

Figure 15 summarizes the GRPO sampling process. FM uses 32 integration steps and $\pi_{0.5}$ uses 10. Each policy call selects one step uniformly for a Gaussian Flow-SDE transition; the remaining steps use deterministic Flow-ODE integration. We store the selected index, action latents before and after that transition, current RGB, robot state, and tactile inputs, the tokenized prompt for $\pi_{0.5}$, and the

old-policy log-probability. During optimization, FM recomputes the selected Flow-SDE transition, whereas $\pi_{0.5}$ recomputes the selected Flow-SDE denoising transition. The resulting current-policy log-probability forms the GRPO likelihood ratio. The generated rollout supplies the reward and the stored transition supplies actor gradients without differentiating through the world model.

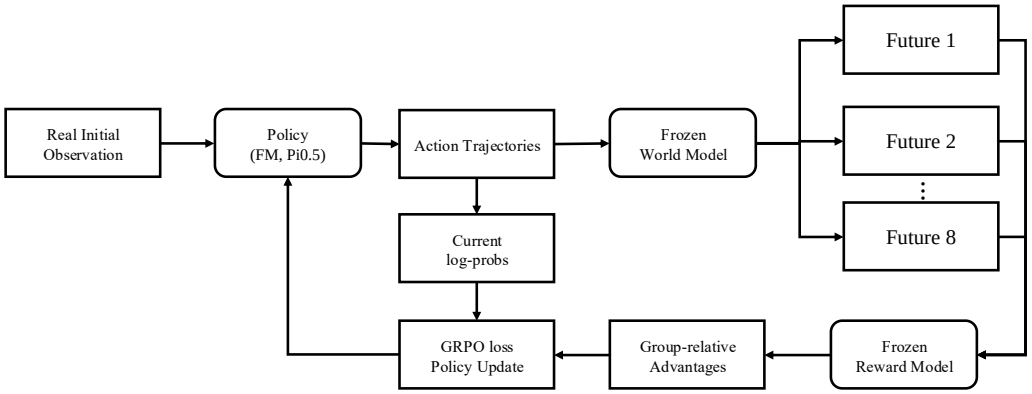


Figure 15: **Policy GRPO in PHYSCANVAS.** Eight sampled trajectories share one real reset condition; three representative futures are shown. The frozen world and reward models provide group-relative advantages, while replayed actions provide current policy log-probabilities for the clipped update.

G.2 GROUPS, REWARDS, AND OBJECTIVE

Each initial condition defines a GRPO group of eight candidate trajectories. A rollout epoch contains 128 trajectories, or 16 groups, and four rollout epochs are accumulated for one update. Thus, one update collects 512 trajectories from 64 initial-condition groups. Candidates within a group share the reset observation and task specification but differ in the stochastic policy transition used to generate their actions. Rollouts run in the multimodal closed loop described in Appendix F, with early termination when the generated task outcome satisfies the task criterion.

For candidate trajectory i, the State Reward Model evaluates the final frame of the generated chunk and outputs a success probability p_i^{end}. We use the success of this last frame as the binary trajectory reward,

$$R_i = \mathbf{1}[p_i^{\text{end}} \geq 0.7] \in \{0, 1\}. \tag{12}$$

The State Reward Model's continuous confidence therefore determines the terminal success label, while GRPO is optimized with the resulting discrete outcome.

For group G, the trajectory advantage is

$$A_i = \frac{R_i - \mu_G}{\sigma_G + 10^{-6}}, \qquad \mu_G = \frac{1}{|G|} \sum_{j \in G} R_j. \tag{13}$$

The same A_i is broadcast to all valid action log-probabilities in trajectory i, while positions after termination are masked. Groups in which all eight trajectories succeed or all eight fail contain no relative preference signal and are removed. Equivalently, for group size eight, training retains groups containing one to seven successful candidates.

Let $\log \pi_{\text{old}}$ be the log-probability recorded during rollout and $\log \pi_\theta$ the value recomputed by the current actor for the same sampled transition. We form

$$\rho_{i,k} = \exp(\log \pi_\theta(a_{i,k} \mid o_{i,k}) - \log \pi_{\text{old}}(a_{i,k} \mid o_{i,k})) \,. \tag{14}$$

We first apply asymmetric PPO ratio clipping,

$$\ell_{i,k}^{\text{clip}} = \max(-A_i \rho_{i,k}, -A_i \,\text{clip}(\rho_{i,k}, 0.8, 1.28)) \,. \tag{15}$$

We additionally use dual clipping with coefficient $c = 3.0$:

$$\ell_{i,k}^{\text{dual}} = \begin{cases} \min\left(\ell_{i,k}^{\text{clip}}, -cA_i\right), & A_i < 0, \\ \ell_{i,k}^{\text{clip}}, & A_i \geq 0, \end{cases} \qquad c = 3.0. \tag{16}$$

The final actor loss is

$$\mathcal{L}_{\text{actor}} = \underset{i,k \in \mathcal{V}_i}{\text{mean}} \ell_{i,k}^{\text{dual}}, \tag{17}$$

where $\mathcal{V}_i$ denotes action log-probability positions that remain valid before trajectory termination. We first apply asymmetric PPO ratio clipping over $[0.8, 1.28]$. For trajectories with negative advantage, dual clipping caps the per-position loss at $-cA_i = c|A_i|$, preventing an excessively large update when the current policy assigns a much higher probability to an action sampled from a below-average trajectory. For non-negative advantages, the dual-clip term has no effect. FM and $\pi_{0.5}$ differ in how the stochastic Flow-SDE transition and its log-probability are constructed, but both policies are optimized with this same dual-clipped GRPO actor objective. The old policy is the snapshot that generated the current rollout batch, not the original supervised checkpoint. We do not train a critic and do not add KL, entropy, or value losses in the reported configuration.

Table 7: **Common GRPO configuration for FM and $\pi_{0.5}$.**

Parameter	Value	Parameter	Value
Group size	8 trajectories	Actor global batch	128
Rollout epochs/update	4	Trajectories/update	512
Micro batch	8	Actor update epochs	1
Learning rate	5×10^{-6}	Adam (β_1, β_2)	$(0.9, 0.95)$
Adam ϵ	10^{-8}	Weight decay	0.01
Gradient clipping	1.0	Ratio clipping	$[0.8, 1.28]$
Dual clip	3.0	KL / entropy coefficient	$0.0/0.0$
Critic	disabled	Checkpoint interval	4 updates
Compute	2 nodes $\times$ 8 GPUs	Reward threshold	0.7

H Metric Computation

This section defines how each metric in Tables 2 and 3 is computed. Ground-truth and predicted quantities are denoted without and with a hat, respectively. Higher PSNR, SSIM, consistency scores, and F1 indicate better performance; lower LPIPS, distributional distances, tracking errors, timing errors, and MAEs are better.

Image reconstruction and distributional distances. For an aligned RGB frame pair $I, \hat{I} \in [0, 1]^{H \times W \times 3}$, the mean-squared error and peak signal-to-noise ratio are

$$e_{\text{MSE}} = \frac{\|I - \hat{I}\|_F^2}{3HW}, \qquad \text{PSNR} = -10 \log_{10} e_{\text{MSE}}. \tag{18}$$

PSNR is computed for each frame and then averaged. Structural similarity is computed over corresponding local windows as

$$\text{SSIM}(u, v) = \frac{(2\mu_u\mu_v + C_1)(2\sigma_{uv} + C_2)}{(\mu_u^2 + \mu_v^2 + C_1)(\sigma_u^2 + \sigma_v^2 + C_2)}, \tag{19}$$

where μ, σ^2, and σ_{uv} denote local means, variances, and covariance, and $C_1 = (0.01L)^2$, $C_2 = (0.03L)^2$ for intensity range L. LPIPS is the spatially averaged, learned weighted squared distance between normalized intermediate AlexNet features of I and $\hat{I}$, summed across feature layers.

FID and FVD compare Gaussian feature distributions of ground-truth and generated data using the Fréchet distance

$$D = \|\mu_g - \mu_p\|_2^2 + \text{Tr}\left(\Sigma_g + \Sigma_p - 2(\Sigma_g^{1/2}\Sigma_p\Sigma_g^{1/2})^{1/2}\right). \tag{20}$$

FID applies this equation to image-level Inception features, whereas FVD applies it to I3D features extracted from temporally ordered video clips.

Temporal consistency. Subject and background consistency use normalized frame features z_t from DINO and CLIP, respectively. Each frame is compared with both the first and preceding frames:

$$C = \frac{100}{2(T-1)} \sum_{t=1}^{T-1} \left[\max(0, z_t^\top z_0) + \max(0, z_t^\top z_{t-1}) \right]. \tag{21}$$

Motion smoothness reconstructs an intermediate frame from its two neighbors with a pretrained interpolation model and computes

$$C_{\text{smooth}} = 100 \left(1 - \text{MAE}(I_{\text{mid}}, I_{\text{interp}})\right), \tag{22}$$

averaged over all reconstructed intermediate frames.

End-effector and object tracking. Let p_t and $\hat{p}_t$ be corresponding two-dimensional positions in the ground-truth and generated videos. End-effector position is the mean position of the gripper-jaw points, while object position is obtained from the object track. With visibility indicator v_t, the average tracking error is

$$E_{\text{track}} = \frac{\sum_{t=1}^{T-1} v_t \|\hat{p}_t - p_t\|_2}{\sum_{t=1}^{T-1} v_t}, \tag{23}$$

and is reported in pixels.

Tactile activity and contact error. For one tactile sensor, let I_t and $\hat{I}_t$ denote the ground-truth and predicted difference images and let $R = I_0$ be the reference image. Pixel activity is

$$d_t(x) = \tfrac{1}{3}\|I_t(x) - R(x)\|_1, \qquad A_t(x) = \mathbf{1}[d_t(x) > \tau_d], \qquad \hat{A}_t(x) = \mathbf{1}[\tfrac{1}{3}\|\hat{I}_t(x) - R(x)\|_1 > \tau_d], \tag{24}$$

where τ_d is the fixed image-change threshold. Contact MAE measures RGB prediction error over ground-truth active pixels:

$$\text{Contact MAE} = \frac{\sum_{t,x} A_t(x) \|I_t(x) - \hat{I}_t(x)\|_1}{3 \sum_{t,x} A_t(x)}. \tag{25}$$

Marker-displacement errors. For each valid marker–time pair i, let $f_i, \hat{f}_i \in \mathbb{R}^3$ be the ground-truth and predicted displacement vectors. Define the active and idle sets as

$$\mathcal{A}_f = \{i : \|f_i\|_2 > \tau_f\}, \qquad \mathcal{Z}_f = \{i : \|f_i\|_2 \le \tau_f\}, \tag{26}$$

where τ_f is the fixed displacement threshold. The three marker metrics are

$$\text{Active MAE} = \frac{1}{3|\mathcal{A}_f|} \sum_{i \in \mathcal{A}_f} \|\hat{f}_i - f_i\|_1, \tag{27}$$

$$\text{Idle FP-MAE} = \frac{1}{3|\mathcal{Z}_f|} \sum_{i \in \mathcal{Z}_f} \|\hat{f}_i\|_1, \tag{28}$$

$$\text{Amp. MAE} = \frac{1}{|\mathcal{A}_f|} \sum_{i \in \mathcal{A}_f} \left| \|\hat{f}_i\|_2 - \|f_i\|_2 \right|. \tag{29}$$

Active MAE compares displacement components, Idle FP-MAE measures spurious motion on idle markers, and Amp. MAE compares displacement magnitudes. The reported values are multiplied by 10^3.

Event F1 and boundary timing. For each tactile sensor, a frame is active when its active-pixel fraction exceeds a fixed event threshold. The same rule is applied to ground truth and prediction. Inactive gaps of at most one frame are filled, active runs shorter than two frames are removed, and each remaining run forms an interval $[a, b)$. Ground-truth and predicted intervals are matched one-to-one when their temporal intersection-over-union is at least 0.1.

If K pairs are matched among N_g ground-truth and N_p predicted events, then

$$\text{Precision} = \frac{K}{N_p}, \qquad \text{Recall} = \frac{K}{N_g}, \qquad \text{F1} = \frac{2K}{N_g + N_p}. \tag{30}$$

For each matched pair, onset and offset errors are

$$E_{\text{onset}} = \frac{1}{K}\sum_{j=1}^{K}\frac{|a_j^g - a_j^p|}{r}, \qquad E_{\text{offset}} = \frac{1}{K}\sum_{j=1}^{K}\frac{|b_j^g - b_j^p|}{r}, \tag{31}$$

where r is the frame rate, so both errors are reported in seconds.

Aggregation. Framewise image and temporal scores are averaged over time. Pixel and marker errors pool their eligible elements within each tactile sensor before averaging across sensors and trajectories. Event F1 and timing errors are computed per sensor trajectory and then macro-averaged. FID and FVD are computed directly from the corresponding collections of image and video features.

I Visualization Analysis

This section presents qualitative comparisons with video-generation baselines and examines how multimodal attention changes across network depth. The displayed cases are diagnostic examples rather than estimates of failure frequency.

I.1 Qualitative Comparison with Baselines

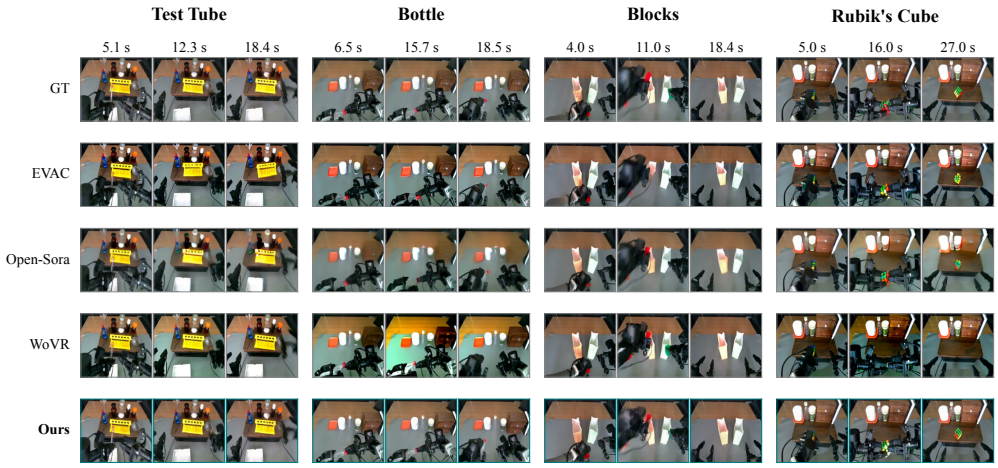


Figure 16: **Qualitative comparison across four tasks.** GT and model rollouts are shown at aligned timestamps.

Figure 16 shows that the displayed PhysCanvas rollouts retain small objects and scene layout more consistently. Open-Sora smooths small container boundaries, WoVR introduces a strong appearance shift, and several baseline predictions alter the bag openings. These selected examples are descriptive rather than estimates of failure frequency.

I.2 Layer-Wise Multimodal Attention Analysis

We visualize modality-level attention for Bottle case 01, averaged over all heads and five denoising steps at layers 0, 14, and 29. Rows denote query modalities and columns denote key modalities; uniform attention would allocate one third of the mass to each equally sized modality band.

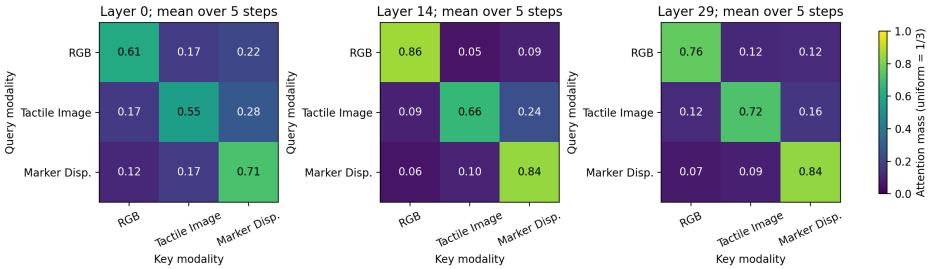
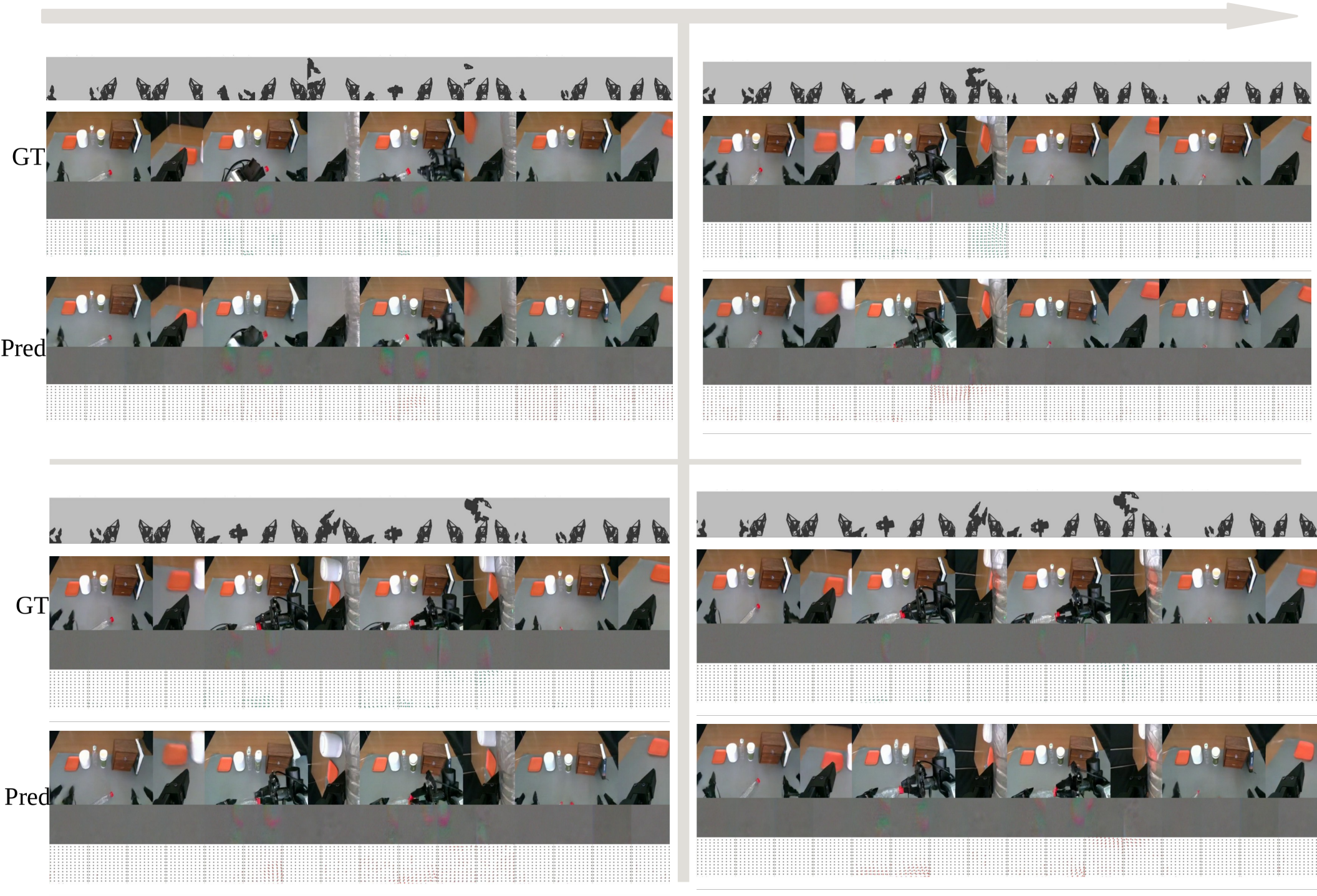
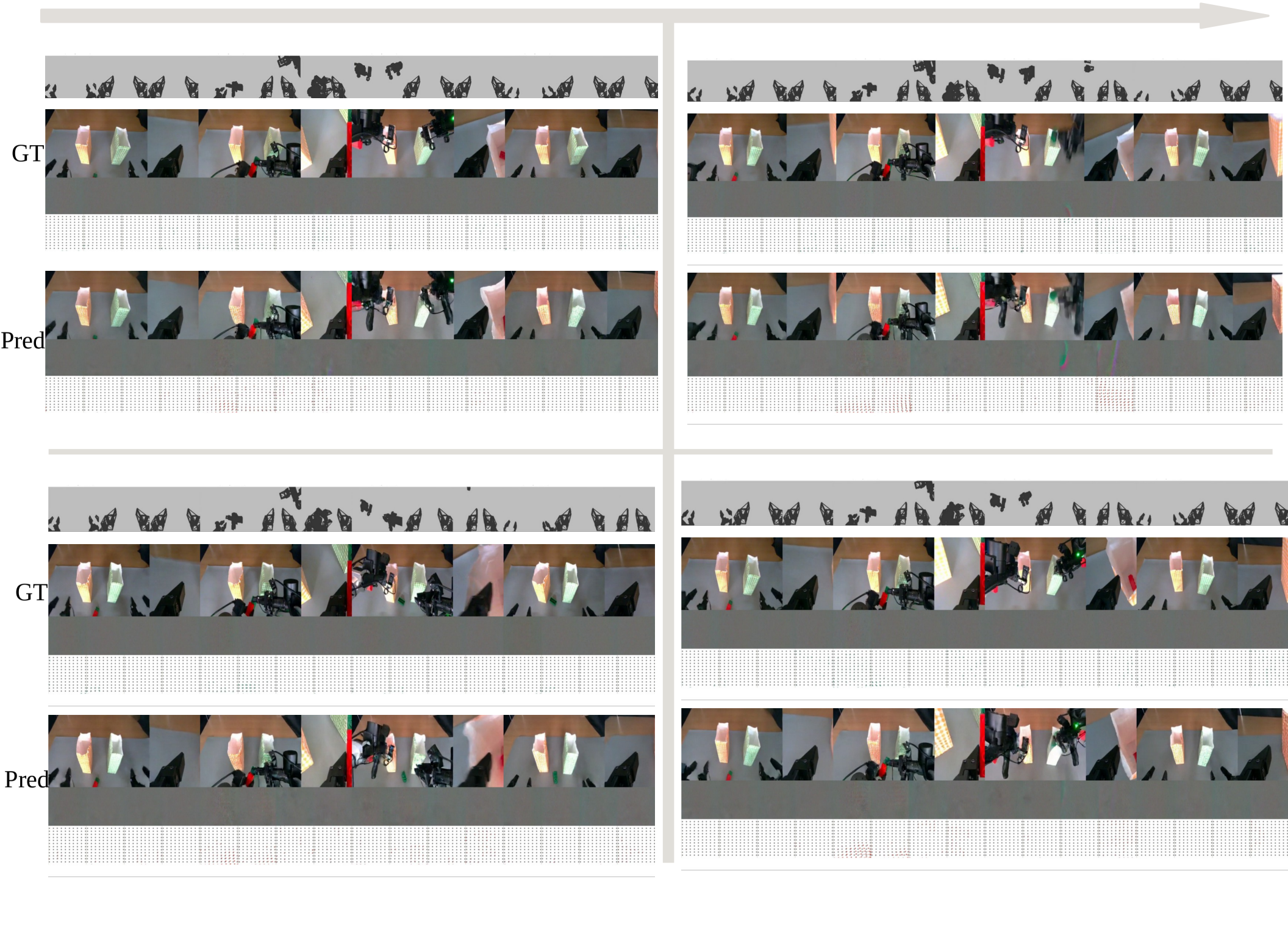


Figure 17: **Layer-wise multimodal attention for Bottle case 01.** Off-diagonal entries show cross-modal routing, while diagonal entries show within-modality attention.

Figure 17 shows the strongest cross-modal exchange in the shallow layer: at layer 0, RGB, tactile-image, and marker-displacement queries allocate 0.39, 0.45, and 0.29 of their attention outside their own modality. The diagonal mass becomes dominant at layers 14 and 29, where the three within-modality entries are $(0.86, 0.66, 0.84)$ and $(0.76, 0.72, 0.84)$, respectively. We find that modality fusion occurs primarily in the initial attention blocks; after cross-modal attention in the shallow representation space, deeper blocks concentrate increasingly on their respective modalities.

The following six pages present qualitative rollout comparisons. The first four compare ground-truth (GT) rollouts with PHYSCANVAS predictions (Pred) conditioned on the same action sequence. The final two compare real-world executions of policies before and after post-training: Base denotes the original supervised fine-tuned policy, whereas GRPO denotes the corresponding policy after Group Relative Policy Optimization through imagined interactions in PHYSCANVAS. In all comparisons, frames progress from left to right.

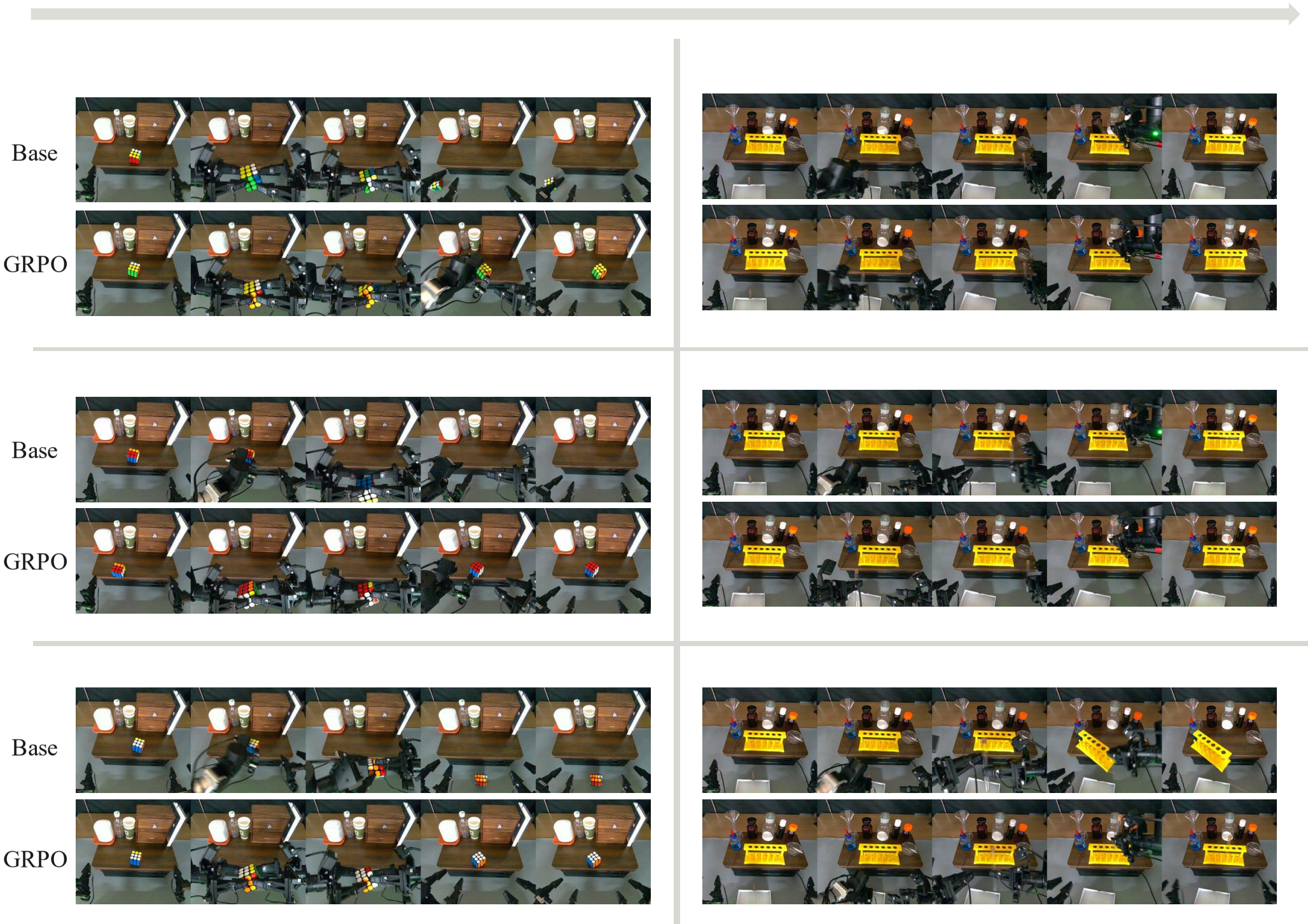

GT
Pred
GT
Pred


GT
Pred
GT
Pred


GT
Pred
GT
Pred


GT
Pred
GT
Pred


Base
GRPO
Base
GRPO
Base
GRPO


Base
GRPO
Base
GRPO
Base
GRPO