VPTWIN: REAL-SIM-REAL VIDEO PREDICTION FOR ROBOTIC MANIPULATION PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

While action-conditioned video prediction provides an intuitive world model for robotics, purely data-driven predictors often suffer from compounding errors and physically implausible hallucinations in long-horizon rollouts, severely undermining downstream action planning. We propose VPTwin, a Real-Sim-Real video prediction framework that anchors real-world future prediction using real-synchronized simulation twins. For a target manipulation task, a VLM reconstructs an executable digital twin from a real demonstration episode. To accommodate the ill-posed estimation of unobserved physical properties, Isaac Sim simulates multiple forward dynamic rollouts across randomized physical configurations under candidate action trajectories. Using these rollouts as in-context references, VPTwin harmonizes both domains, using simulation dynamics to enforce physical plausibility while capturing unmodeled contact interactions from real video. Furthermore, we establish a predictive planning loop using VPTwin to visually verify VLM-proposed actions and guide reliable real-world execution. Evaluations show substantial reductions in physical hallucinations during video prediction and marked improvements in manipulation planning performance.

1 INTRODUCTION

Action-conditioned video prediction serves as an intuitive visual world model for robotic manipulation by allowing robots to imagine future visual states (Wu et al., 2024; Agarwal et al., 2026). However, purely data-driven predictors struggle with compounding errors during long-horizon autoregressive rollouts and often generate physically implausible hallucinations such as object interpenetration, floating artifacts, and spontaneous motion. Crucially, hallucinations mislead downstream policy evaluation and cause action planning to fail. Physics simulation offers a complementary remedy by enforcing explicit dynamical governing equations and conservation laws, guaranteeing physical consistency over extended horizons. Yet, relying solely on simulators inevitably exposes decision models to physical simulation-to-reality gaps in geometry, friction, and complex contacts (Chen et al., 2024; Liu et al., 2026a). Recognizing that simulation provides principled dynamic grounding while real-world video supplies unmodeled contact interactions and photorealistic fidelity, we propose a Real-Sim-Real video prediction paradigm that bridges both worlds.

To realize this paradigm, we present VPTwin, which uses simulation dynamics to physically anchor real-domain video prediction for robotic manipulation. Within this pipeline, a VLM reconstructs executable digital twins and estimates distributions over physical parameters directly from real-world scene observations. Notably, frontier VLMs, such as GPT-6 Astra (OpenAI, 2026), substantially reduce the manual effort required to construct task-specific digital twins. Since inferring exact, unobserved physical properties from visual observations is inherently ill-posed, we leverage Isaac Sim to simulate multiple forward dynamic rollouts under randomized physical parameter configurations, conditioned on the same future action trajectories (Figure 1A).

Building on these contexts, we introduce VPTwin-Core, a simulation-synchronized video prediction model that conditions on multiple possible future outcomes from the approximate digital twins, using them as physical references to mitigate physically implausible predictions and reduce forward hallucinations (Figure 1B). Furthermore, to prevent over-optimistic predictions during planning, VPTwin incorporates failure-aware co-training: successful episodes are supervised by real-world

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.

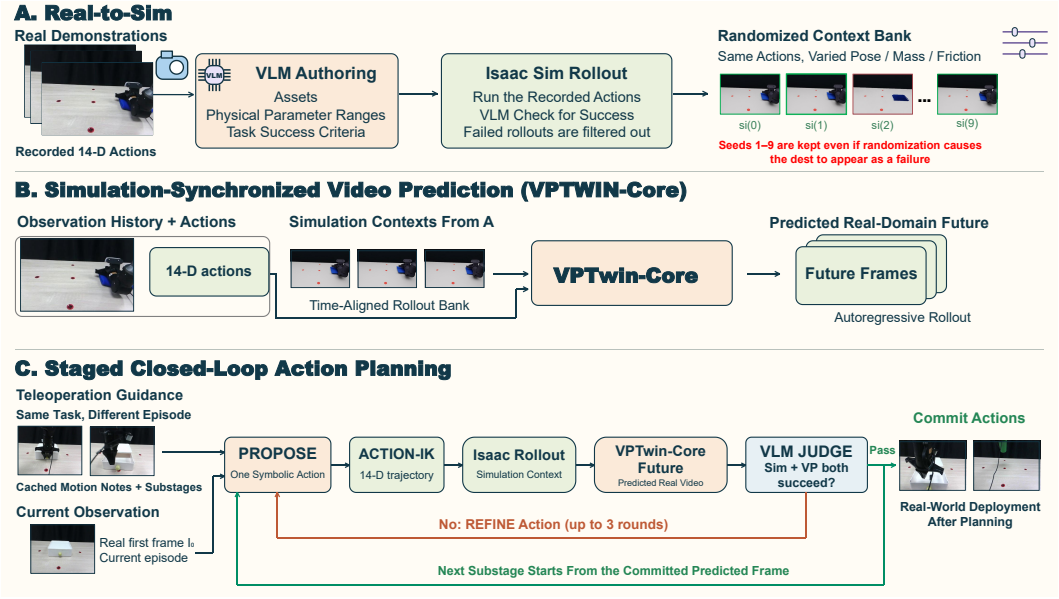


Figure 1: **Overview of VPTwin.** (A) A VLM reconstructs executable digital-twin assets to drive randomized dynamic rollouts in Isaac Sim. (B) Conditioned on time-aligned simulation rollouts, VPTwin-Core synthesizes physically grounded real-domain future video under candidate actions. (C) Staged action planning evaluates and refines candidate actions via joint simulation and visual verification before real-world deployment.

demonstrations, while synthetic failure rollouts generated in simulation expose VPTwin-Core to counterfactual failure modes, without the need for costly real-world failure collection.

We establish a closed-loop planning framework that employs VPTwin-Core as a predictive visual verifier (Figure 1C). Specifically, a VLM analyzes one-shot demonstrations drawn from the predictor's training set and decomposes them into semantically structured substages. At each substage, the VLM proposes candidate actions, which are first evaluated in Isaac Sim and then projected into prospective real-world visual outcomes using VPTwin-Core. An action is executed on the real robot only if its predicted visual outcome meets predefined success criteria, as assessed by the VLM. Otherwise, the predicted visual feedback guides iterative action refinement. By screening candidate actions through physics-anchored visual foresight before real-world execution, the framework reduces hardware trial-and-error and helps prevent costly execution failures.

Our primary contributions are summarized as follows:

- We introduce VPTwin, a Real-Sim-Real vision-based model predictive control framework for robotic manipulation. By grounding generative visual rollouts in automated digital twins, it enforces physical plausibility throughout forward modeling.
- We develop VPTwin-Core, an in-context world model conditioned on randomized, time-aligned physical rollouts, augmented by synthetic failure co-training to accurately foresee the visual futures under unsuccessful actions without collecting real failure data.
- We design a staged planning algorithm that leverages VPTwin-Core as a predictive visual verifier to evaluate VLM-proposed actions and guide adaptive refinement prior to real-world execution.
- Extensive evaluations across rigid, deformable, and articulated manipulation tasks demonstrate substantial reductions in physical hallucinations and marked gains in task success.

2 Related Work

Video prediction for robotic manipulation. Action-conditioned video prediction models the visual consequences of robot-object interactions, serving as an intuitive world model for manipulation planning and policy evaluation. Early frameworks such as UniPi convert generated video plans into robot actions (Du et al., 2023), while AVDC extracts control signals via dense visual correspon-

dences and geometric reasoning (Ko et al., 2024). Subsequent autoregressive and diffusion models learn controllable visual dynamics directly from demonstration data (Wu et al., 2024; Zhu et al., 2025). Recent efforts extend this paradigm to joint video-action modeling (Li et al., 2025), latent feature prediction for planning (Zhou et al., 2024), multiview generation (Guo et al., 2026; Gemini Robotics Team et al., 2025), human video transfer (Gao et al., 2026), and motion-aligned visual control (Cao et al., 2026; Li et al., 2026) However, in contact-rich manipulation, visual history and actions cannot resolve unobserved physical parameters, causing extended predictions to accumulate physically implausible hallucinations (Chen et al., 2026b; Wang et al., 2026). We therefore investigate whether alternative simulation rollouts across randomized physical parameters provide complementary context for predicting real-world continuations.

Real-to-Sim scene reconstruction. Constructing interactive simulation environments from visual inputs is essential for physical modeling. Prior work reconstructs rigid, soft, and articulated assets from visual inputs (Xiang et al., 2025; Yang et al., 2024; Zhang et al., 2026; Huang et al., 2026; Le et al., 2025; Chen et al., 2024; Pfaff et al., 2026), with recent systems further automating executable URDF generation using LLM-based coding agents (Zhou et al., 2026; Xiao et al., 2026; He et al., 2026). While existing methods prioritize geometric and kinematic fidelity, identifying unobserved physical parameters from visual observation is inherently ill-posed. Rather than relying on a single deterministic estimate, our pipeline uses a VLM to infer plausible parameter distributions, synthesizing executable digital twins specifically designed to generate diverse physical rollout contexts.

Simulation-guided video generation and planning. Modern physics engines provide controllable dynamic environments for embodied agents (Todorov et al., 2012; NVIDIA; Zhu et al., 2020). Several works leverage simulation to facilitate visual learning: ReBot synthesizes robot training videos through simulation replay (Fang et al., 2025), while RoboTransfer (Liu et al., 2026b), CRAFT (Chen et al., 2026a), AnchorDream (Ye et al., 2025), and RealWonder (Liu et al., 2026c) guide video diffusion with simulated geometry or motion priors. To mitigate dynamical discrepancy, PGRD learns residual corrections over simulated trajectories (Patel et al., 2026). VPTwin departs from this by conditioning real-video prediction on multiple randomized rollouts of the identical action sequence, using simulation as a multi-hypothesis physical prior. For downstream planning, frameworks such as Prompting-with-the-Future (Ning et al., 2025) and SIMPACT (Liu et al., 2026a) optimize trajectories or symbolic plans within digital twins. Yet, planning entirely within simulation remains susceptible to physical reality gaps. We address this with a staged Real-Sim-Real planner.

3 METHOD

This section provides the technical details of VPTwin. Sec. 3.1 introduces our Real-to-Sim procedure for authoring executable digital twins from visual inputs under domain-randomized physics. Sec. 3.2 presents VPTwin-Core, a video prediction model conditioned on robot actions, visual history, and synchronized physics rollouts. Sec. 3.3 establishes a planning algorithm that leverages the frozen predictive model to iteratively evaluate and refine action plans before physical execution.

3.1 REAL-TO-SIM TWIN AUTHORING AND RANDOMIZED PHYSICS ROLLOUTS

To provide physics-grounded guidance for real-domain video prediction, we develop an automated Real-to-Sim pipeline that transforms raw demonstration videos into executable digital twins in Isaac Sim. Because inferring exact, unobserved dynamic attributes (such as friction coefficients, mass distributions, and contact compliance) from passive visual observations is inherently ill-posed, treating a single simulated model as ground truth introduces severe model misspecification. To overcome this fundamental limitation, our pipeline couples automated twin authoring with domain-randomized dynamic rollouts, generating a multi-hypothesis physical context that covers plausible dynamic variations rather than relying on brittle point estimates (see appendix Figure 8).

Automated digital-twin authoring. Given a real demonstration episode, we employ a frontier VLM (*e.g.*, GPT-6 Astra) (OpenAI, 2026) to parse third-view keyframes sampled across the trajectory ($0\%, 20\%, 40\%, 60\%, 70\%, 80\%$, and 100%), identify object categories, detect geometric occlusions, infer kinematic articulation, and estimate initial placements on a calibrated workspace. The VLM then programmatically generates Python scripts that instantiate Universal Scene Description (USD) assets, specifying collision primitives, inertial properties, and joint limits. To guarantee physical stability within the physics engine, an automated coding agent (Cursor, 2026; SpaceXAI,

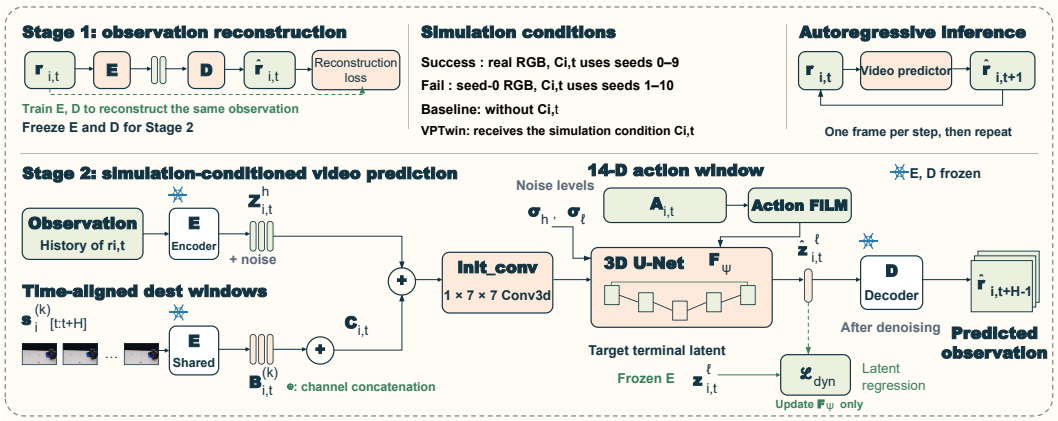


Figure 2: **Architecture of VPTwin-Core.** Two-stage training and autoregressive inference conditioned on candidate actions and time-aligned simulation rollouts.

2026) refines actuator kinematics, converting the drawer as a fixed-base articulation with a prismatic joint and deploying deformable items as PhysX surface deformables. In parallel, to benchmark our automated authoring against an expert baseline, we curate a measured-catalog setting consisting of 548 real-world episodes across six task families, where object geometries are verified by human annotators. Workspace calibration and asset construction are detailed in Appendix A, B.

VLM-driven success filtering. To ensure that only physically valid simulations serve as predictive priors, the VLM establishes formal task specifications and visual success criteria for each task family (see appendix Table 9). The nominal simulation rollout (seed 0), driven by recorded 14-D actions, is evaluated via multi-timestamp visual inspection with denser temporal sampling during contact-rich phases. This automated verification retains 277 physically consistent episodes across four manipulation families for downstream world-model training.

Domain-randomized physical rollouts. For each retained demonstration, Isaac Sim rollouts the recorded 14-D action sequence across a bank of ten randomized dynamic rollouts. As shown in Figure 1, seed 0 represents the nominal configuration, while seeds 1–9 systematically perturb object initial poses by Δp and sample dynamical parameters ϑ (*e.g.*, friction and mass) from VLM-specified plausible distributions, while holding the recorded action trajectory, tabletop geometry, camera calibration, and lighting invariant. Rather than constraining the video predictor to an exact simulation twin, this ensemble of time-aligned rollouts exposes the model to diverse physical continuations under approximate digital twins.

3.2 Simulation-Synchronized Video Prediction

Counterfactual failure synthesis. Standard teleoperation datasets are often positively biased, containing almost exclusively successful executions. As a result, video predictors may develop overly optimistic predictions, hallucinating successful outcomes even for flawed candidate actions. To reliably verify downstream plans, the predictive model must generalize to counterfactual actions and faithfully visualize failure outcomes. Because collecting physical failures on hardware is costly and hazardous, we synthesize failure trajectories in simulation. For episode i, let $r_{i,t}$ and $a_{i,t} \in \mathbb{R}^{14}$ denote the visual observation and joint action, defining the full trajectory $A_i = (a_{i,0}, \ldots, a_{i,T_i-1})$. While successful demonstrations use real video supervision, failure episodes apply kinematic perturbations during critical phases (*e.g.*, misaligned approaches or premature gripper releases; Appendix C). The nominal simulation rollout $s_i^{(0)}$ under perturbed actions A_i then serves as the pseudo-observation target, providing failure supervision without collecting real failure data.

VPTwin-Core architecture. Given the co-training dataset, Isaac Sim simulates forward dynamic rollouts driven by action sequence A_i across varied physical parameters and initial placements to generate a multi-hypothesis physical context pool:

$$s_i^{(k)} = \mathcal{R}(A_i, p_i + \Delta p_{i,k}, \vartheta_{i,k}; \Gamma), \qquad k \in \mathcal{K}_i, \tag{1}$$

where p_i is the nominal object placement, $\Delta pi, k$ represents spatial placement perturbation, $\vartheta_{i,k}$ denotes sampled physical parameters, and Γ specifies static scene parameters, including tabletop

geometry, camera poses, lighting, and robot base calibration. The rollout configuration index set is $\mathcal{K}_i = \{0, \ldots, 9\}$ for successful episodes and $\mathcal{K}_i = \{0, \ldots, 10\}$ for counterfactual failure episodes. Given a prediction horizon $H = 10$, let $A_{i,t} = A_i[t : t + H]$ define the time-aligned action window. The visual latent representation of the k-th simulation rollout window is extracted via a frozen encoder as $B_{i,t}^{(k)} = E(s_i^{(k)}[t : t + H])$. As shown in Figure 2, following the two-stage formulation of IWS (Wang et al., 2026), Stage 1 trains a CNN encoder E and a consistency decoder D to reconstruct RGB frames via latent-conditioned denoising. Stage 2 freezes both E and D, and trains a 3D spatiotemporal dynamics model F_ψ. Observation, action, and simulation rollout windows are strictly time-aligned. Let $Z_{i,t}^h$ denote the mixed-noise latent clip of $r_i[t : t + H]$, with $H - 1$ lightly noised history slots and terminal noise level σ_h. The target $z_{i,t}^\ell$ is the latent of $r_{i,t+H-1}$ at noise level σ_ℓ. We concatenate ten rollout latent clips:

$$C_{i,t} = \oplus_{k \in \mathcal{I}_i} B_{i,t}^{(k)}, \qquad \mathcal{I}_i = \begin{cases} \{0, \ldots, 9\}, & \text{success episode,} \\ \{1, \ldots, 10\}, & \text{fail episode,} \end{cases} \tag{2}$$

where $\oplus$ denotes channel concatenation. The dynamics model predicts the terminal low-noise latent:

$$\widehat{z}_{i,t}^\ell = F_\psi\left(Z_{i,t}^h \oplus C_{i,t}; A_{i,t}, \sigma_h, \sigma_\ell\right). \tag{3}$$

In Stage 2, the same frozen encoder E encodes observations and simulation rollouts. The 3D U-Net F_ψ receives the concatenated volume $Z_{i,t}^h \oplus C_{i,t}$ at its first convolution and modulates intermediate spatiotemporal features with the action window $A_{i,t}$ via Feature-wise Linear Modulation (FiLM) (Perez et al., 2018). During autoregressive inference, each predicted latent step updates the sliding context window to condition subsequent steps before being decoded into RGB video frames. The dynamics network is optimized under a uniform-MSE objective: $\mathcal{L}_{\text{dyn}}(\psi) = \mathbb{E}[\|\widehat{z}_{i,t}^\ell - z_{i,t}^\ell\|_2^2]$. The exact noise schedule and effective terminal-only regression are detailed in Appendix D.

3.3 VPTwin-Based Model Predictive Control

To translate physics-grounded predictive foresight into reliable real-world manipulation, we formulate a staged Real-Sim-Real closed-loop planning framework that employs the frozen VPTwin-Core model as a predictive visual verifier. Directly optimizing symbolic plans across an entire long-horizon skill frequently suffers from compounding search errors. To mitigate this, we decompose complex manipulation skills into semantically structured substages informed by demonstration motion priors. Specifically, a VLM first analyzes teleoperation keyframes from the training set to construct a structured motion specification per task family, defining sequential 3D Cartesian waypoints, end-effector wrist orientations, gripper actuation timings, and critical contact milestones. A secondary pass establishes substage boundaries, permissible motion primitives, and local success criteria. The planner comprises five steps: scene reconstruction is performed once, followed by steps 2–5 within each substage. The complete trajectory is executed after all substages are planned (Algorithm 1 in Appendix E details the procedure).

1. **Reconstruction:** Identifies the target task family and spatial workspace configuration directly from the initial real observation frame I_0, once before substage planning.
2. **Propose:** Prompts a VLM to generate a candidate symbolic action conditioned on the current visual observation, reference demonstration keyframes, and the nominal simulated state.
3. **Action-IK:** Converts symbolic parameters (translations, wrist rotations, and gripper commands) into an executable 14-D joint trajectory via inverse kinematics (Lynch & Park, 2017) with 2 cm Cartesian interpolation. Isaac Sim subsequently simulates forward dynamic rollouts across seeds 0–9 to construct the physical conditioning context for visual prediction.
4. **Judge:** Evaluates the proposed action against the substage success criterion using both the nominal simulation rollout (seed 0) and the real-domain visual future synthesized by VPTwin-Core. Crucially, simulation-only checks leave the planner vulnerable to unmodeled discrepancies in contact and friction, leading to false acceptances during hardware deployment. In contrast, our dual-domain verification helps prevent physically infeasible actions from being deployed.
5. **Refine & Commit:** If either the simulation rollout or the predicted real future fails the check, an adaptive refinement module prompts the VLM to revise the action using the visualized failure rollout, demonstrations, and historical feedback for up to three rounds; detailed in Appendix E. The first passing candidate is committed, and its final predicted frame and simulator state initialize the next substage. If none passes, planning terminates without hardware execution.

Table 1: **Full-episode video prediction results** (mean ± std across tasks after averaging each family's scores over five training seeds) Corresponding results for each task are presented in Table 14.

Model	MSE ↓	LPIPS ↓	FID ↓	FVD ↓	PSNR ↑	SSIM ↑	UIQI ↑
Measured-catalog (548), 6 families							
IWS (2026)	0.016 ± 0.008	0.329 ± 0.071	276.0 ± 32.1	2095.0 ± 384.4	25.58 ± 3.00	0.770 ± 0.069	0.249 ± 0.049
VPTwin	$\mathbf{0.007 \pm 0.002}$	$\mathbf{0.284 \pm 0.056}$	$\mathbf{265.6 \pm 36.9}$	$\mathbf{2014.5 \pm 264.7}$	$\mathbf{28.57 \pm 1.92}$	$\mathbf{0.798 \pm 0.058}$	$\mathbf{0.283 \pm 0.040}$
VLM-authored (277), 4 families							
IWS (2026)	0.016 ± 0.009	0.310 ± 0.067	277.7 ± 45.3	2450.0 ± 701.8	25.64 ± 3.35	0.780 ± 0.073	0.262 ± 0.048
VPTwin	$\mathbf{0.006 \pm 0.002}$	$\mathbf{0.259 \pm 0.048}$	$\mathbf{259.1 \pm 42.3}$	$\mathbf{2131.0 \pm 340.6}$	$\mathbf{28.96 \pm 1.83}$	$\mathbf{0.809 \pm 0.060}$	$\mathbf{0.294 \pm 0.039}$

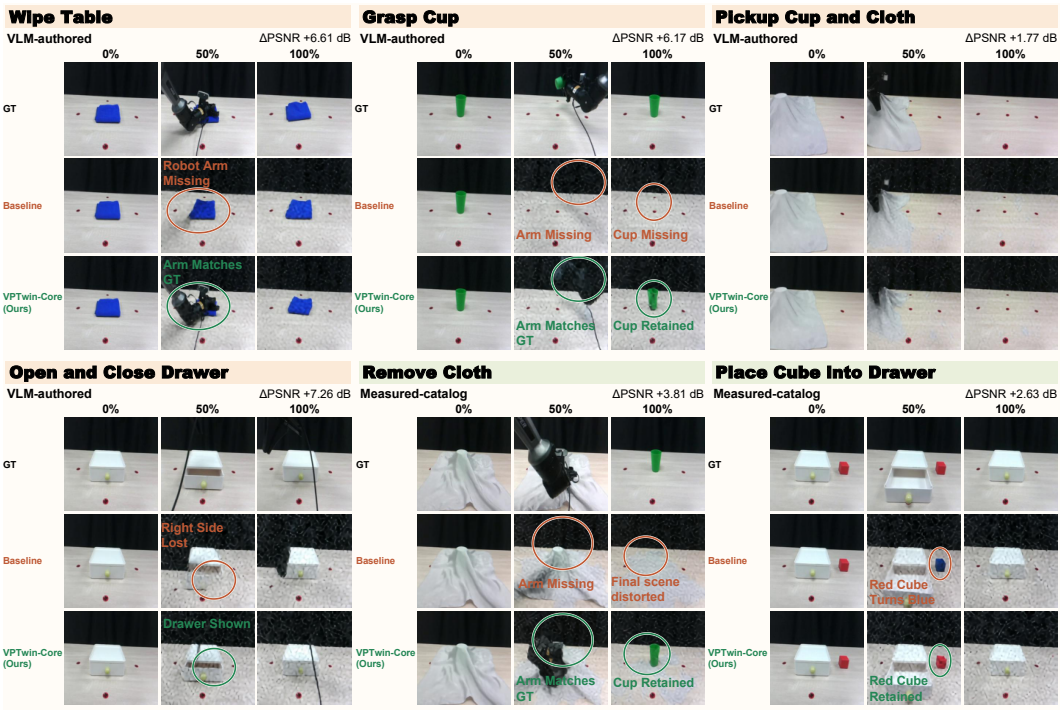


Figure 3: **Full-episode video prediction results at** 0%, 50%, **and** 100% **of an execution timeline.** Orange and green circles highlight baseline failure artifacts and our physically consistent predictions, respectively. ΔPSNR denotes the quantitative gain of VPTwin over IWS (Wang et al., 2026).

4 Experiment

4.1 Experimental Setup

We evaluate visual prediction on held-out episodes across two distinct digital-twin settings: (i) the *measured-catalog* setting (548 episodes across six tasks) and (ii) the automated *VLM-authored* setting (277 episodes across four tasks). The suite spans six tabletop tasks: *Wipe Table* (surface wiping with a cloth remaining on the table), *Grasp Cup* (lifting and setting a cup upright near its start, or at another slot in measured-catalog variants), *Pickup Cup and Cloth* (bimanual lifting of both objects out of view), *Remove Cloth* (clearing the cloth while leaving the cup upright), *Open and Close Drawer* (sequential prismatic drawer articulation), and *Place Cube into Drawer* (opening a drawer, inserting a cube, and closing it). We benchmark real-world planning against SIMPACT (Liu et al., 2026a) and open-loop $\pi_{0.5}$ (Physical Intelligence et al., 2025) on *Grasp Cup* (7 trials, rigid-body contact, five placement slots) and *Open and Close Drawer* (5 trials, articulated constraint, centered). To assess execution robustness under physical uncertainty, object placements are perturbed across episodes. Experiments use bimanual ALOHA with AgileX Piper arms and a static third-person camera. (Appendix Figure 7). Each trial initiates from frame I_0 and executes a committed 14-D joint trajectory. Teleoperation demonstrations are a separate training set. Details are in Appendix A.

Following IWS (Wang et al., 2026), we evaluate MSE/PSNR (Wang & Bovik, 2009), SSIM (Wang et al., 2004), UIQI (Wang & Bovik, 2002), LPIPS (Zhang et al., 2018), FID (Heusel et al., 2017), and FVD (Unterthiner et al., 2018) using checkpoints trained for 10^6 steps.

Table 2: **Real-robot task success and critic evaluation metrics.** Precision, Recall, and F1 assess critic verification against true physical outcomes: Precision denotes physical success rates among accepted plans, while Recall reflects the proportion of successful executions accepted. Rejected plans are additionally executed for evaluation.

	Grasp Cup				Open and Close Drawer			
Method	Success	Precision	Recall	F1-score	Success	Precision	Recall	F1-score
$\pi_{0.5}$	0/7	na	na	na	0/5	na	na	na
SIMPACT	2/7	0.33	1.00	0.50	2/5	0.67	1.00	0.80
VPTwin	**3/7**	**1.00**	**1.00**	**1.00**	**5/5**	**1.00**	**1.00**	**1.00**

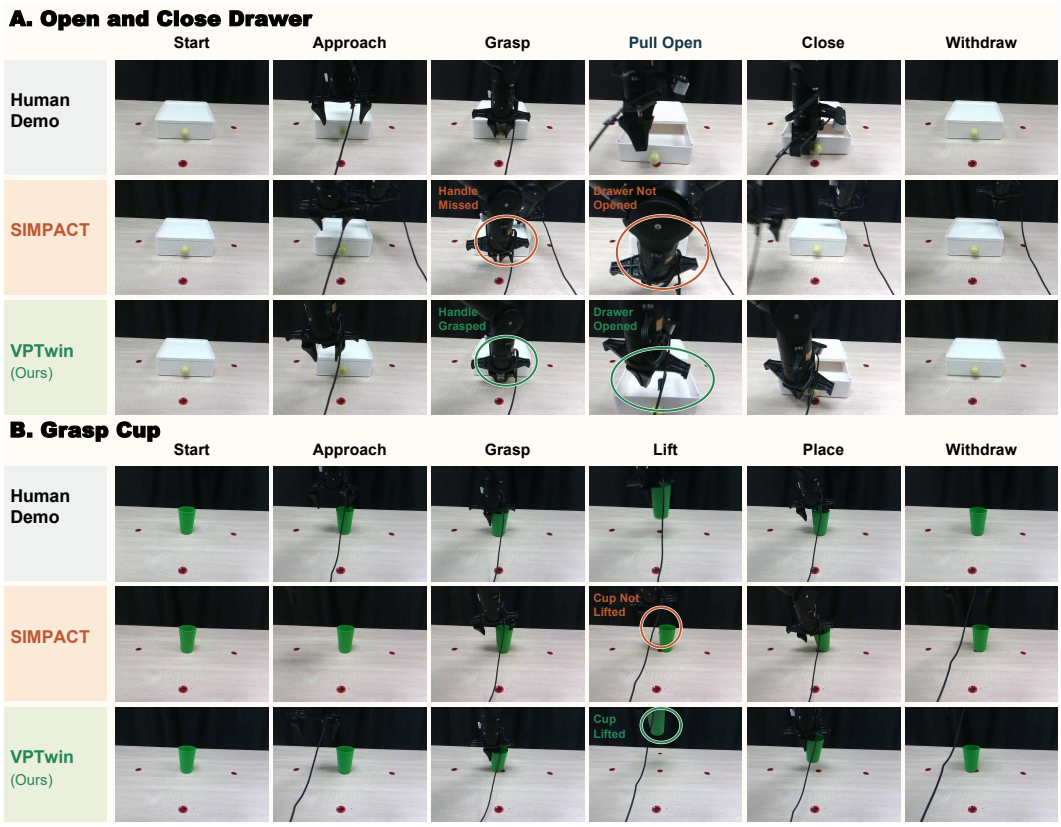


Figure 4: **Real-robot deployment trajectories.** Columns represent sequential task phases. Orange and green circles highlight failure and success modes, respectively.

4.2 Long-Term Video Prediction Results

As shown in Table 1, VPTwin consistently outperforms IWS across all family-averaged metrics. It improves PSNR by 2.99 dB in the measured-catalog setting and by 3.32 dB in the VLM-authored setting, while reducing LPIPS from 0.329 to 0.284 and from 0.310 to 0.259. Figure 3 shows that our model substantially mitigates temporal drift, accurately tracking manipulator articulation, object permanence (*e.g.*, cup presence), and rigid drawer geometry throughout extended rollouts.

4.3 Action Planning Results

As reported in Table 2, VPTwin demonstrates superior real-world manipulation performance over the baselines, improving execution success over SIMPACT from $2/7$ to $3/7$ on *Grasp Cup* and achieving a perfect $5/5$ (versus $2/5$) on *Open and Close Drawer*, while open-loop $\pi_{0.5}$ fails across all trials. Crucially, evaluating the alignment between critic decisions and physical execution outcomes reveals the key vulnerability of simulation-only planning: while both planners achieve a Recall of 1.00, SIMPACT's simulation critic incurs five false acceptances (four in grasping and one in drawer articulation), yielding low Precision scores of 0.33 and 0.67. In contrast, our dual-domain

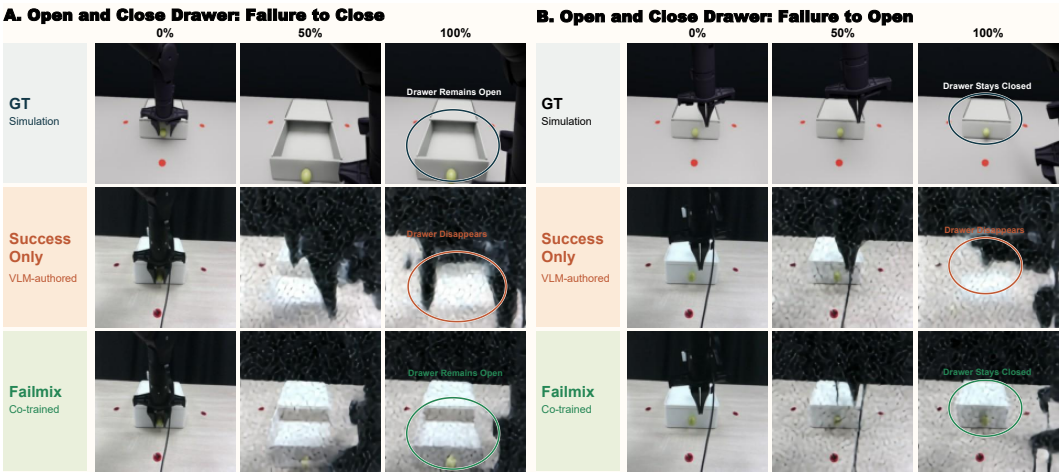


Figure 5: **Visual prediction rollouts on failure episodes.** Frame sequences across simulation ground truth, the success-only model, and the failure-aware co-trained model (*Failmix*) at 0%, 50%, and 100% completion for (A) unachieved drawer closing and (B) unachieved drawer opening.

Table 3: **Prediction results of failure episodes in PSNR (dB).** *Failmix* incorporates synthetic failure co-training, while *Success-Only* relies solely on the 277 filtered success demonstrations. Scores are evaluated on 30-frame prefixes and full rollouts.

		30 frames		Full-episode	
Start	Task	**Failmix**	**Success-only**	**Failmix**	**Success-only**
First frame	Grasp Cup	**23.38**	23.30	**23.21**	23.13
	Open and Close Drawer	**22.46**	22.39	**21.22**	20.25
Perturbed frame	Grasp Cup	**23.36**	23.23	**23.20**	23.11
	Open and Close Drawer	**21.32**	21.03	**20.76**	19.90

visual verification strictly rejects physically infeasible candidates, achieving perfect Precision and F_1 scores of 1.00 across both tasks by completely eliminating false acceptances. Figure 4 illustrates SIMPACT missing the drawer handle and failing to lift the cup, while VPTwin completes both tasks.

4.4 Ablation Studies

Effect of synthetic failure supervision. Incorporating simulated failure rollouts consistently improves visual prediction over extended execution horizons (Table 3). For *Open and Close Drawer*, full-rollout PSNR increases by 0.97 dB (first-frame initialization) and 0.86 dB (perturbed-frame initialization), compared to 0.07 dB and 0.29 dB across 30-frame rollouts. Full-rollout gains for *Grasp Cup* reach 0.08–0.09 dB, with consistent gains reflected in SSIM (Table 15). Co-training successfully prevents optimistic hallucinations, faithfully visualizing the incomplete drawer state and anticipating cup slippage under flawed actions where the success-only baseline fails.

Sensitivity to context volume. Figure 6(a,b) evaluates model sensitivity to the number of simulation rollouts provided at inference. For models trained with ten rollouts, one test-time rollout ($k = 1$) substantially improves PSNR over zero context ($k = 0$). However, this inference-time plateau does not imply that fewer rollouts suffice during training. As shown in Table 4, when evaluated with an identical budget of three test-time rollouts, the model trained with ten rollouts outperforms the three-rollout training configuration across all six measured-catalog tasks. Full results are presented in Appendix Tables 16-17.

Robustness to sim-real outcome mismatch. We further stress-test resilience against dynamic discrepancies by deliberately supplying failed simulation rollouts for physically successful demonstrations under identical actions. In Figure 6(c,d), At $k = 1$, failed simulation context still provides beneficial geometric priors on rigid-body and articulated tasks, yielding measured-catalog PSNR gains over the unconditioned baseline of 3.1 dB (*Grasp Cup*), 4.1 dB (*Open and Close Drawer*), and 3.0 dB (*Place Cube into Drawer*). Across VLM-authored tasks, drawer articulation consistently benefits, while deformable tasks (*Wipe Table*) degrade and *Pickup Cup and Cloth* improves

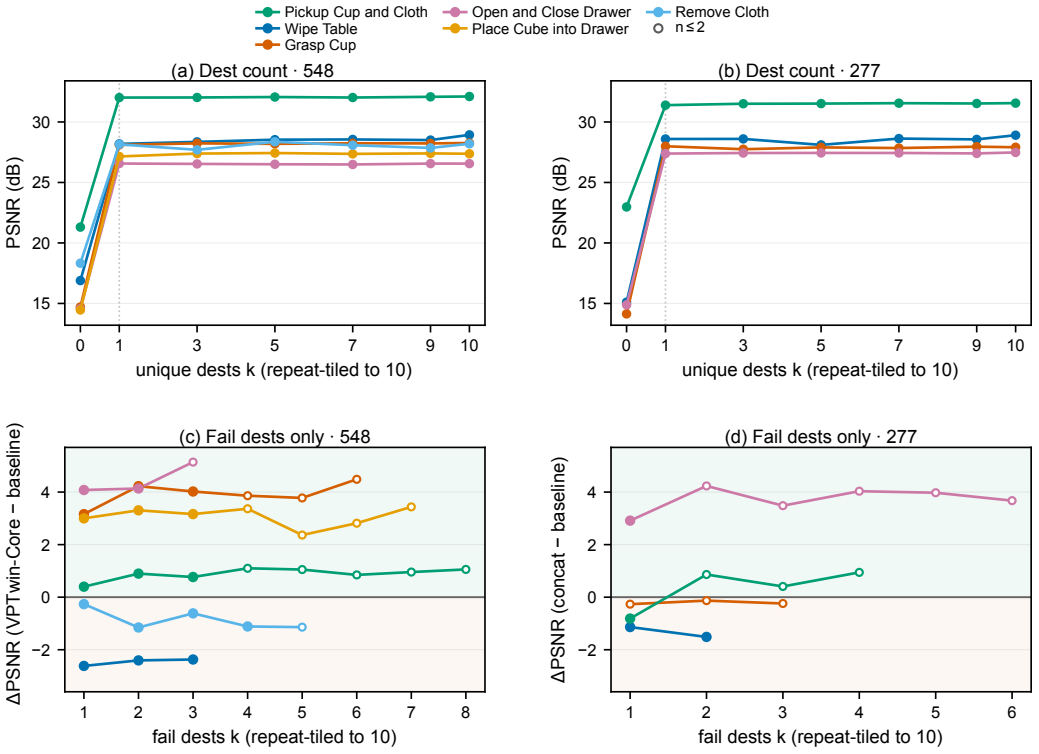


Figure 6: **Sensitivity to context quantity and outcome mismatch during inference.** Columns distinguish digital-twin settings: (a, c) measured-catalog and (b, d) VLM-authored. **(a, b)** Full-test PSNR as a function of the number of distinct simulation rollouts k cyclically repeated across the 10 conditioning slots ($k = 0$ corresponds to zero simulation context); values for $k \in \{1, 3, 5, 7, 9\}$ report averaged results across three evaluation seeds. **(c, d)** Relative prediction gain under counterfactual, failure-only simulation context. Open markers denote sparse sample regimes ($n \leq 2$).

Table 4: **Full-rollout PSNR (dB) with matched test context.** Both models use seeds 0–2, repeated to fill ten conditioning slots for the ten-rollout model. Rows denote training → test rollout counts.

Train → Test	WipeTable	GraspCup	PickCup/Cloth	RemoveCloth	Open/CloseDrawer	PlaceCube	Mean
3 → 3	27.04	28.03	31.89	27.12	25.72	27.14	27.82
10 → 3	**29.04**	**28.33**	**32.08**	**28.65**	**26.58**	**27.43**	**28.69**

as k scales. More results are in included Appendix Tables 11-13. These findings suggest that completely failed simulation can still provide useful spatial and physical cues rather than misleading the predictor. Crucially, our automated Real-to-Sim reconstruction and physical domain randomization pipeline suppresses such simulation false negatives in practice (where physical executions succeed while simulated rollouts fail), ensuring reliable execution guidance.

5 Conclusion and Limitations

We presented VPTwin, a Real-Sim-Real framework that grounds visual prediction in automated digital twins to suppress physical hallucinations. By conditioning on randomized simulation rollouts and co-training on synthetic counterfactual failures, our approach reliably anticipates candidate action outcomes. As a predictive visual verifier in staged planning, it eliminates false acceptances and improves manipulation success on physical hardware over simulation-only planners.

Several limitations outline avenues for future work. First, automated twin authoring assumes calibrated camera extrinsics and task-specific reconstruction, motivating extensions toward open-world digital twin synthesis. Second, synthetic failure co-training does not yet capture the full diversity of unmodeled real-world dynamics. Finally, while physical trials validate feasibility, scaling to larger multi-stage manipulation suites and closed-loop online adaptation remains promising future work.

AI USE STATEMENT

Generative AI tools and large language models were utilized exclusively for language polishing, grammatical refinement, and schematic figure drafting during manuscript preparation. AI was not employed to conceptualize research questions, design the core methodology, conduct physical or simulated experiments, or interpret scientific results. All algorithmic formulations, experimental analyses, and manuscript contents were conceived and verified by the authors, who assume full intellectual responsibility for this work.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, complete technical formulations, hyperparameter schedules, and experimental protocols are documented throughout the paper and supplementary material. Specifically, the core algorithmic formulations of VPTwin and staged action planning are established in Section 3. Digital-twin asset construction, workspace calibration, and domain randomization parameters are detailed in Appendices A-B. The counterfactual failure synthesis pipeline, dataset partitioning, training regimes, and autoregressive rollout protocols are specified in Appendices C-D. Staged planning heuristics and visual evaluation protocols (including FID and FVD benchmarks) are elaborated in Appendices E-F. All codebase, simulation environments, USD assets, and VLM prompt templates (Examples in Appendix G) will be made publicly available upon acceptance.

REFERENCES

Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

Zhi Cao, Howard Ji, Kevin Zhang, Kuangzhi Ge, Fei-Fei Li, Jiajun Wu, and Huang Huang. Tract: Bridging robot control and visual prediction with visual tracks. *arXiv preprint arXiv:2608.24101*, 2026.

Jason Chen, I-Chun Arthur Liu, Gaurav Sukhatme, and Daniel Seita. CRAFT: Video Diffusion for Bimanual Robot Data Generation. In *International Conference on Intelligent Robots and Systems (IROS)*, 2026a.

Taiye Chen, Xun Hu, Zihan Ding, and Chi Jin. Learning world models for interactive video generation. *Advances in Neural Information Processing Systems*, 38:154456–154483, 2026b.

Zoey Chen, Aaron Walsman, Marius Memmel, Kaichun Mo, Alex Fang, Karthikeya Vemuri, Alan Wu, Dieter Fox, and Abhishek Gupta. Urdformer: A pipeline for constructing articulated simulation environments from real-world images. *arXiv preprint arXiv:2405.11656*, 2024.

Cursor. Cursor Grok 4.6. Cursor Documentation, 2026. URL https://cursor.com/help/models-and-usage/grok-4-6. Accessed: 2026-09-14.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

Yu Fang, Yue Yang, Xinghao Zhu, Kaiyuan Zheng, Gedas Bertasius, Daniel Szafir, and Mingyu Ding. Rebot: Scaling robot learning with real-to-sim-to-real robotic video synthesis. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11351–11358. IEEE, 2025.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

Gemini Robotics Team, Krzysztof Choromanski, Coline Devin, Yilun Du, Debidatta Dwibedi, Ruiqi Gao, Abhishek Jindal, Thomas Kipf, Sean Kirmani, Isabel Leal, et al. Evaluating gemini robotics policies in a veo world simulator. *arXiv preprint arXiv:2512.10675*, 2025.

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.

Yumeng He, Yichen Song, Xiaotian Yang, Weijia Zhang, Zanwei Zhou, Junru Gong, Xiaokang Yang, and Yunbo Wang. Neoworld-pro: Programming interactive scenes from monocular images for embodied simulation. In *NeurIPS*, 2026.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Mu Huang, Hui Wang, Kerui Ren, Linning Xu, Yunsong Zhou, Mulin Yu, Bo Dai, and Jiangmiao Pang. Soma: A real-to-sim neural simulator for robotic soft-body manipulation. *arXiv preprint arXiv:2602.02402*, 2026.

Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to act from actionless videos through dense correspondences. In *International Conference on Learning Representations*, volume 2024, pp. 40938–40958, 2024.

Long Le, Jason Xie, William Liang, Hung-Ju Wang, Yue Yang, Yecheng Jason Ma, Kyle Vedder, Arjun Krishna, Dinesh Jayaraman, and Eric Eaton. Articulate-anything: Automatic modeling of articulated objects via a vision-language foundation model. In *International Conference on Learning Representations*, volume 2025, pp. 17578–17602, 2025.

Hongyu Li, Bowen Wen, Xinghao Zhu, Yixuan Wang, Yilun Du, Yunzhu Li, George Konidaris, Stan Birchfield, Soha Pouya, Chenran Li, et al. Hydra-0: Action flow for generalist world modeling and control. *arXiv preprint arXiv:2608.18077*, 2026.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model, 2025. URL https://arxiv.org/abs/2503.00200.

Haowen Liu, Shaoxiong Yao, Haonan Chen, Jiawei Gao, Jiayuan Mao, Jia-Bin Huang, and Yilun Du. Simpact: Simulation-enabled action planning using vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20790–20801, 2026a.

Liu Liu, Xiaofeng Wang, Guosheng Zhao, Keyu Li, Wenkang Qin, Jiagang Zhu, Jiaxiong Qiu, Zheng Zhu, Guan Huang, and Zhizhong Su. Robotransfer: Controllable geometry-consistent video diffusion for manipulation policy transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1410–1420, 2026b.

Wei Liu, Ziyu Chen, Zizhang Li, Yue Wang, Hong-Xing Yu, and Jiajun Wu. Realwonder: Real-time physical action-conditioned video generation. *arXiv preprint arXiv:2603.05449*, 2026c.

Kevin M Lynch and Frank C Park. Modern robotics. *Mechanics, Planning, and Control*, 2017.

Chuanruo Ning, Kuan Fang, and Wei-Chiu Ma. Prompting with the future: Open-world model predictive control with interactive digital twins. *arXiv preprint arXiv:2506.13761*, 2025.

NVIDIA. Isaac Sim. Computer software. URL https://github.com/isaac-sim/IsaacSim. Version 5.1.0-rc.19; accessed September 25, 2026.

OpenAI. GPT-6 Astra System Card. System card, OpenAI, 2026. URL https://deploymentsafety.openai.com/gpt-6-astra.

Shivansh Patel, Kaifeng Zhang, Sanjay Pokkali, Svetlana Lazebnik, and Yunzhu Li. Learning physics-guided residual dynamics for deformable object simulation. *arXiv preprint arXiv:2607.13451*, 2026.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Nicholas Pfaff, Thomas Cohn, Sergey Zakharov, Rick Cory, and Russ Tedrake. Scenesmith: Agentic generation of simulation-ready indoor scenes. In *Forty-third International Conference on Machine Learning*, 2026.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

SpaceXAI. Model Card: Grok 4.6. Model card, SpaceXAI, 2026. URL https://media.x.ai/v1/website/card-4p6-4cd2dc57.pdf.

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.

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

Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. ISBN 978-1-4414-1269-0.

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.

Zhou Wang and Alan C Bovik. A universal image quality index. *IEEE signal processing letters*, 9 (3):81–84, 2002.

Zhou Wang and Alan C Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine*, 26(1):98–117, 2009.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. ivideogpt: Interactive videogpts are scalable world models. *Advances in Neural Information Processing Systems*, 37:68082–68119, 2024.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21469–21480. IEEE, 2025.

Zijian Xiao, Zipeng Ye, Jinkun Hao, Xiong Yang, Yuchen Xie, and Ran Yi. Beyond placement and articulation: Usage-driven code scenes for embodied interaction. *arXiv preprint arXiv:2608.18840*, 2026.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, et al. Holodeck: Language guided generation of 3d embodied ai environments. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16277–16287. IEEE, 2024.

Junjie Ye, Rong Xue, Basile Van Hoorick, Pavel Tokmakov, Muhammad Zubair Irshad, Yue Wang, and Vitor Guizilini. Anchordream: Repurposing video diffusion for embodiment-aware robot data synthesis. *arXiv preprint arXiv:2512.11797*, 2025.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Shujie Zhang, Jingkun Yi, Weipeng Zhong, Zirui Zhou, Yangkun Zhu, Hanqing Wang, Xudong Xu, Weinan Zhang, and Chunhua Shen. Robosnap: One-shot real-to-sim scene generation for generalizable robot learning and evaluation. *arXiv preprint arXiv:2607.06699*, 2026.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

Matt Zhou, Ruining Li, Xiaoyang Lyu, Zhaomou Song, Zhening Huang, Chuanxia Zheng, Christian Rupprecht, Andrea Vedaldi, and Shangzhe Wu. Articraft: An agentic system for scalable articulated 3d asset generation. *arXiv preprint arXiv:2605.15187*, 2026.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: A fine-grained world model for robot manipulation. *https://arxiv. org/abs/2406.14540*, 2025.

Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Kevin Lin, Abhiram Maddukuri, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

APPENDIX

A EXPERIMENTAL DETAILS

A.1 PLATFORM AND WORKSPACE

The tabletop measures 120×60 cm along the world x and y axes and stands 74.4 cm above the floor. The table center defines $(x, y) = (0, 0)$, and $+y$ points from the robot toward the black backdrop. Illustrated in Figure 7. The left and right arm-base centers are at $(-31.6, -45.4)$ cm and $(+30.1, -45.4)$ cm, respectively. Both centers lie 15.4 cm beyond the near table edge and 75.5 cm above the floor. These coordinates refer to base centers, not footprint edges. The camera optical center is at $(+0.7, -45.1)$ cm, 100 cm above the floor and 25.6 cm above the tabletop. It faces downward at $25°$ from horizontal, with $1.93°$ roll and zero yaw. Its distance to the table-center marker is approximately 51.9 cm.

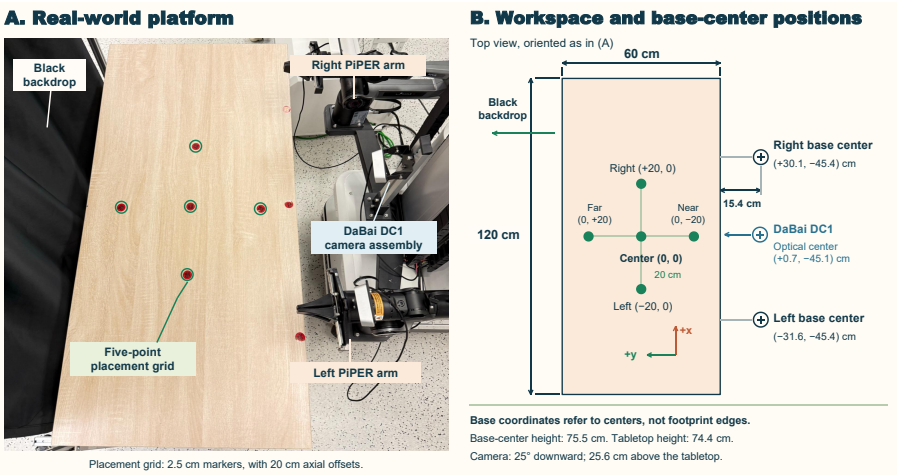


Figure 7: **Real-robot experimental setup.** (A) Bimanual platform, camera, and five-point placement grid. (B) Workspace dimensions and base-center coordinates. Each slot is marked by a 2.5 cm-diameter red sticker; objects cover the marker without requiring perfect center alignment.

A.2 TASK PLACEMENTS

Five red markers, each 2.5 cm in diameter, define discrete object placements: center $(0, 0)$, left $(-20, 0)$, right $(20, 0)$, far $(0, 20)$, and near $(0, -20)$, in centimeters. Objects occupy these designated points rather than continuously sampled positions. Task instructions and success criteria appear in Table 9; placements and object dimensions are listed in Table 5. The markers specify nominal placements rather than exact object centers. An object occupies a slot when its footprint covers the marker; the resulting placement error is not directly measured. Simulation placement perturbations are detailed in Appendix B.2; their ranges are not measured bounds on real-world placement error.

A.3 REAL-ROBOT EVALUATION PROTOCOL

The seven *Grasp Cup* trials use center, center, far, near, near, left, and right placements. The five *Open and Close Drawer* trials use separate initial photographs and executions of the same centered layout. Substage evaluation, refinement, and acceptance are detailed in Appendix E.

B SCENE-ASSET AND PHYSICS-ROLLOUT DETAILS

B.1 MEASURED-CATALOG TWIN

The measured catalog specifies the table mesh, camera calibration, robot geometry (The VLM-authored pipeline requires the first three as well), and placement grid described in Appendix A,

but not episode-specific object placements or measured physical properties. Object geometry is human-checked for the 548 episodes across six task families. A coding agent supplies mass, friction, other physical parameters, and their randomization ranges (Cursor, 2026; SpaceXAI, 2026); these properties are not physically measured.

Isaac rollouts each recorded 14-D action trajectory on these assets. Seed 0 is the human-accepted rollout. Seeds 1–9 preserve the recorded actions while randomizing placement, physical parameters, and selected geometric dimensions (Appendix B.2). The table, camera, lights, and robot remain fixed across seeds.

B.2 PHYSICS RANDOMIZATION

Placement perturbations act along one tabletop direction, chosen from $\{+x, -x, +y, -y\}$; the other horizontal coordinate and z remain unchanged. Magnitudes are integer millimeters. Measured-catalog seeds 1–3 sample from $\{1, 2\}$ mm, seeds 4–6 from $\{3, 4\}$ mm, and seeds 7–9 use 5 mm. VLM-authored seeds sample uniformly from $\{1, 2, 3, 4, 5\}$ mm. Placement perturbation range(1-5mm) suggested by VLM (OpenAI, 2026).

Measured-catalog parameters. Seeds 1–3, 4–6, and 7–9 use mild, mid, and harsh parameter ranges, respectively (Table 6). Interval-valued parameters are sampled uniformly; braced entries denote two-point distributions.

Towel pack height is perturbed by $\delta_h \in \{-3, -2, -1, 1, 2, 3\}$ mm and clipped to $[1.7, 2.3]$ cm. Cup wall thickness is perturbed by $\delta_w \in \{-1, 0, 1\}$ mm and clipped to $[0.3, 1.5]$ mm. These geometric perturbations use the same rule across the three severity levels. Cloth thickness changes only when the seed modulo 9 belongs to $\{0, 3, 6\}$, using signed offsets of 1–3 mm and clipping to $[0.5, 3.8]$ mm. Ranges suggested by coding agent (SpaceXAI, 2026; Cursor, 2026)

VLM-authored parameters. Each parameter is sampled uniformly from its authored interval $[c - \delta_-, c + \delta_+]$, where c is the suggested value and δ_- and δ_+ are the authored offsets.

Table 5: Placement and object size for the six task families. The five markers are the table center and four axial offsets of 20 cm. Catalog object dimensions were human-checked; listed masses are nominal coding-agent settings, not physical measurements. Only *Grasp Cup* and *Open and Close Drawer* are executed in the real-robot comparison. The VLM-authored set is Wipe Table, green-cup Grasp Cup, Pickup Cup and Cloth, and Open and Close Drawer.

Task	Placement	Catalog reference	VLM suggested
Wipe Table	Folded blue towel on one of the five markers.	Pack 15 × 15 × 2 cm.	Pack 16 × 16 × 1.8 cm.
Grasp Cup	Upright cup on one of the five markers. Catalog episodes use a green or blue cup; VLM episodes use the green cup.	Height 10 cm; bottom / top radius 2.5 / 3.5 cm; 40 g.	Height 10 cm; bottom / top radius 2.5 / 3.48 cm; 45 g.
Pickup Cup and Cloth	Cloth covers the cup. Both sit on the left, center, or right marker.	Sheet 50 × 53 cm, 1.0 mm thick, 4 g.	Sheet 52.8 × 50.4 cm, 0.8 mm thick, 4 g.
Remove Cloth	Cloth covers the cup. Both sit on the left, center, or right marker.	Same sheet as Pickup Cup and Cloth.	Same VLM sheet and VLM cup. This family is catalog-only.
Open and Close Drawer	Fixed on the center marker.	20 × 20 × 7 cm; travel 15 cm.	20 × 20 × 7.3 cm; cavity depth 6.0 cm; travel 15.5 cm.
Place Cube into Drawer	Fixed at the center; blue or red cube on the left or right marker.	Cube 5 cm, 50 g.	Cube 5 cm, 75 g. This family is catalog-only.

Table 6: Measured-catalog parameter randomization. Severity levels correspond to seeds 1–3, 4–6, and 7–9. Friction coefficients are dimensionless. Ranges suggested by coding agent.

Parameter	Mild	Mid	Harsh
Towel mass (g)	30–50	22–65	$\{20, 80\}$
Towel friction	0.8–1.2	0.55–1.4	$\{0.40, 1.60\}$
Cup mass (g)	25–48	15–55	$\{10, 60\}$
Cup friction	0.9–1.5	0.6–1.8	$\{0.40, 2.00\}$
Cloth mass (g)	3–6	2.5–8	$\{2, 10\}$
Cloth friction	1.4–2.0	1.0–2.1	$\{0.60, 2.20\}$
Drawer static effort (N)	1.2–3.0	0.7–5.0	$\{0.40, 8.0\}$
Cube mass (g)	40–70	30–90	$\{25, 120\}$
Cube friction	1.2–1.8	0.8–2.0	$\{0.60, 2.20\}$

Table 7: VLM-suggested parameter ranges. Each interval is $[\text{suggested} - \text{minus}, \text{suggested} + \text{plus}]$ from the VLM range file. The green and blue cups share one range set; the red and blue cubes share another.

Parameter	Suggested	Suggested range
Towel width x (cm)	16	14–18.5
Towel depth y (cm)	16	14–19
Towel pack height (cm)	1.8	1.2–2.6
Towel density (kg/m^3)	90	10–210
Towel mass (g)	41.47	1.47–101.47
Towel friction	1.05	0.65–1.25
Cloth width x (cm)	52.8	46.8–62.8
Cloth depth y (cm)	50.4	44.4–58.4
Cloth thickness (mm)	0.80	0.45–1.05
Cloth areal density (kg/m^2)	0.01509	0.01509–0.12
Cloth friction	1.85	0.35–2.05
Cup height (cm)	10	8.8–11.5
Cup bottom radius (cm)	2.5	2.2–2.9
Cup top radius (cm)	3.48	3.18–3.88
Cup wall thickness (mm)	0.50	0.30–1.30
Cup base thickness (mm)	1.0	0.50–2.50
Cup mass (g)	45	30–70
Cup friction	1.2	0.40–1.40
Drawer width x (cm)	20	18–22.5
Drawer depth y (cm)	20	17.5–23
Drawer height (cm)	7.3	6.1–8.8
Drawer cavity depth (cm)	6.0	5.0–7.2
Drawer lid well (mm)	8	6–11
Drawer outer wall (mm)	3	2–4
Drawer tray wall (mm)	4	3–5
Drawer side gap (mm)	1.0	0.7–1.5
Drawer bottom gap (mm)	1.0	0.7–1.5
Drawer front thickness (mm)	8	6–10
Drawer handle radius (cm)	1.4	1.1–1.7
Drawer travel (cm)	15.5	13–18
Drawer static effort (N)	2.0	1.0–3.5
Drawer dynamic effort (N)	1.0	0.55–2.0
Drawer mass (g)	400	200–700
Drawer friction	0.85	0.45–1.05
Cube edge (cm)	5.0	4.4–5.8
Cube mass (g)	75	30–130
Cube friction	1.6	0.45–1.80

B.3 VLM TWIN AUTHORING AND FILTERING

GPT-6 Astra (OpenAI, 2026) is called twice: first to parse the scene and then to author the assets. Both calls receive the 0% third-view layout image and later stills at 20, 40, 60, 70, 80, and 100%.

Later stills are used to infer object identity, occlusion, articulation, and motion; the placement slot is determined from the 0% image. The prompt provides the measured table extents and five-point 20 cm slot grid. The VLM selects a discrete slot, which the host converts to metric coordinates. The authoring call writes Python (Van Rossum & Drake, 2009) scripts specifying geometry, collision shapes, mass, friction, articulation, and sampling ranges. The scripts are executed locally to produce USD assets. The VLM-authored drawer and deformable assets require simulator-specific adaptation. A coding agent converts the drawer into a fixed-base articulation and deploys the cloth and towel as PhysX surface deformables (Cursor, 2026; SpaceXAI, 2026; NVIDIA). The authoring procedure examines 24 initial-layout variants. The VLM-authored video-prediction set uses 14 of these variants: five wipe-table, five green-cup grasp, three pickup, and one drawer variant. The overview pipeline is in Figure 8.

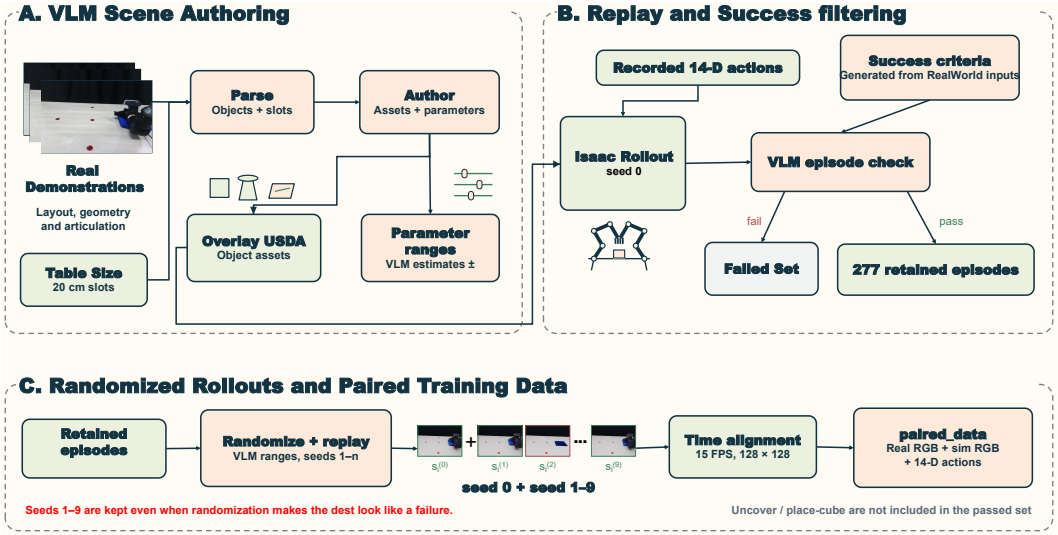


Figure 8: **VLM scene authoring and simulation-data construction.** (A) A VLM authors digital-twin assets and parameter ranges from real demonstrations. (B) VLM-derived success criteria retain 277 seed-0 rollouts. (C) Retained episodes are randomized, time-aligned, and paired with real RGB and actions for video-predictor training.

Success filtering. For each task family, the VLM receives real third-view stills at 0, 20, 40, 60, 80, and 100% of the episode and writes a textual task instruction and success criterion (Table 9). Each seed-0 dest is then evaluated against that criterion using stills at 0, 50, and 100%. Grasp episodes are rechecked on a denser mid-timeline, at 60, 70, 75, and 80% as well as 0, 50, and 100%. Only accepted seed-0 dests enter the VLM video-prediction set, yielding 277 episodes across wipe table, grasp cup, pickup cloth and cup, and open and close drawer.

C FAIL-DEST CONSTRUCTION

The real dataset contains successful demonstrations but no real failure videos. We construct 118 failure episodes from the filtered VLM set: 32 grasp, 43 drawer never-open, and 43 drawer fail-close cases.

Perturbation onset and constraints. Grasp and never-open cases preserve the demonstration until five frames before the first gripper-open event. Fail-close cases preserve the initial pull-and-release sequence and begin perturbation five frames after release. Perturbations affect the active arm's six joints and gripper; some drawer modes hold the other arm fixed from the sign-flip frame onward. The resulting 14-D joint trajectory is clipped to URDF limits. Fingertip targets are constrained to $x \in [-0.22, 0.22]$ m, $z \in [0.78, 1.28]$ m, and the original trajectory's y range expanded by 0.08 m at each end.

Action perturbations. Remaining Cartesian displacements receive axis signs $s_x, s_y, s_z \in \{-1, +1\}$, with $\Pr(s_x = -1) = \Pr(s_z = -1) = 0.7$ and $\Pr(s_y = +1) = 0.7$. If $s_x = s_z = +1$, we set $s_x = -1$. Displacements are scaled by a factor sampled from $U(0.75, 1.25)$ and supplemented with a wandering component of amplitude $U(2, 6)$ cm and period $U(12, 28)$ frames. Gripper commands use inversion, held-open, held-closed, or periodic toggling with probabilities 0.25, 0.20, 0.20, and 0.35, respectively. Toggling intervals range from 6 to 16 frames.

Outcome-based verification. Failure labels depend on object trajectories rather than perturbation magnitude. Successful grasp requires a lift of at least 5 cm, return to the tabletop, an upright orientation, and stability over the final eight frames. Drawer success requires opening by at least 8 cm and subsequently closing to within 3 cm. Never-open cases fail the opening condition; fail-close cases satisfy opening but fail closing.

Simulation supervision. Isaac Sim rollout s each perturbed action trajectory on the corresponding VLM-authored assets. Seed 0 supplies the observation and prediction target; seeds 1–10 use the same actions with randomized simulation parameters and supply ten conditioning rollouts. Seed 0 is excluded from the conditioning set.

D Video Prediction Implementation Details

D.1 Training settings

Both stages use AdamW with learning rate 8×10^{-5}, weight decay 10^{-4}, and betas $(0.9, 0.999)$. The learning rate warms up linearly over 10^4 steps from 10^{-4} of its base value. Gradients are clipped at norm 1, and training uses FP32. Each stage runs for $1{,}000{,}005$ steps; reported predictions use `final.ckpt`. Stage 1 trains E and D with batch size 1 and horizon 1. Stage 2 freezes both and trains F_ψ with batch size 1 and horizon $H = 10$. The action-only baseline uses the same optimizer, batch size, training budget, frozen autoencoder, data split, and action windows, but omits $C_{i,t}$. Models are trained separately for each task family and arm. The comparisons use five independent training seeds.

D.2 Stage-1 encoder and decoder

E consists of Conv2d layers; D combines a consistency U-Net with a ControlNet conditioned on $z = E(r)$. Stage 1 jointly trains E and D to reconstruct observation frames by denoising RGB conditioned on the encoded latent. Both are frozen in Stage 2. For failure-aware co-training, Stage 1 reconstructs real success frames and seed-0 simulated failure frames; seeds 1–10 serve only as conditioning rollouts in Stage 2.

D.3 Aligned Stage-2 windows

Stage 2 uses $H = 10$ time-aligned frames sampled at 15 Hz and resized to 128×128. We define,

$$s_i^{(k)}[t : t + H] = \left(s_{i,t}^{(k)}, \ldots, s_{i,t+H-1}^{(k)} \right), \tag{4}$$

so the window contains exactly H frames and excludes $t + H$. The same temporal indices are applied to the observation video, 14-D actions, and every dest rollout. Because E operates frame-wise, $B_{i,t}^{(k)}$ is an H-frame latent clip. Success observations are real RGB frames, conditioned on simulation seeds 0–9. Failure observations are seed-0 renders under the perturbed action trajectory; conditioning slot $k \in \{0, \ldots, 9\}$ uses seed $k + 1$. Observation, action, and conditioning windows use matching temporal indices.

D.4 Terminal-only supervision

Stage 2 processes an H-frame latent clip in which the first $H - 1$ positions provide the observation history and the final position is the prediction target. For each history position $j < H - 1$, the input and target noise levels are matched, $\tau_j = \rho_j$, so the consistency model applies an identity mapping and the corresponding regression residual is zero. Only the terminal position is denoised from noise

Table 8: Episode-level train/validation/test splits. The VLM-authored set inherits the measured-catalog assignments.

	Measured-catalog			VLM-authored		
Task	Train	Val	Test	Train	Val	Test
Wipe Table	86	10	10	62	8	8
Grasp Cup	103	13	13	24	5	3
Pickup Cup and Cloth	85	10	10	68	5	8
Remove Cloth	35	5	5	–	–	–
Open and Close Drawer	84	10	10	70	8	8
Place Cube into Drawer	47	6	6	–	–	–
Total	440	54	54	224	26	27

step $\tau_{H-1} = 999$ to $\rho_{H-1} = 0$. Consequently, although the implementation averages the regression loss over all temporal positions, the effective objective is

$$\mathcal{L}_{\text{dyn}} = \frac{1}{H} \left\| \widehat{z}^{\ell}_{i,t} - z^{\ell}_{i,t} \right\|_2^2, \tag{5}$$

which supervises only the terminal-frame latent. The factor $1/H$ is constant and does not change the optimization objective.

D.5 Autoregressive inference

Inference starts from one encoded observation without padding, using a shorter prefix until the window reaches $H = 10$. Each step appends a standard Gaussian latent clipped to $[-6, 6]$ and retains at most H slots. Existing slots use stabilization noise index 15; F_ψ denoises the new slot in one consistency step, $999 \to 0$ (`dyn_infer_steps`=1). The updated window, including rewritten history latents, is carried into the next step. The same Stage-1 decoder D renders each latent in three consistency steps, $999 \to 666 \to 333 \to 0$ (`dec_infer_steps`=3). The configured 50-step DDIM schedule is unused.

D.6 Episode splits

Splits are assigned to complete episodes before extracting training windows. For the measured-catalog set, train, validation, and test counts are fixed within each task variant (placement, cup color, or cube color), and episodes are shuffled within each variant using seed 42. This yields $440/54/54$ train/validation/test episodes.

The VLM-authored set inherits these assignments by retaining the same episode keys that pass VLM filtering, without reshuffling. It contains $224/26/27$ episodes and excludes *Remove Cloth* and *Place Cube into Drawer*. Table 8 lists the family-level counts.

D.7 Success–failure split and sampling

The main video-prediction comparison uses success-only training. Failure-aware co-training is evaluated separately and provides the frozen predictor used for action planning. The co-training set comprises 118 real success episodes and their 118 simulated failure counterparts, whereas the success-only comparator uses the 277-episode success set. Each failure episode inherits its successful parent's split. The 118 success–failure pairs yield 188/26/22 train/validation/test episodes: 48/10/6 for *Grasp Cup* and 140/16/16 for *Open and Close Drawer*. The test set contains three grasp pairs and eight drawer pairs. Training uses balanced sampling of real successes and constructed simulation failures. Success episodes use real RGB targets with seeds 0–9 as simulation conditions, whereas fail episodes use seed 0 as the observation and target and seeds 1–10 as conditions. The paired set contains 32 grasp episodes and 86 drawer episodes: 43 never-open and 43 fail-close cases. Separately generated failures for 78 wipe and 81 pickup episodes are excluded from this co-training set.

Failure-rollout evaluation starts. *First frame* initializes prediction from frame 0 of the failed seed-0 simulation. *Perturbed frame* initializes prediction from the frame at which action perturbation

begins. For either start frame t_0, the ground-truth seed-0 video, perturbed 14-D actions, and ten conditioning rollouts (seeds 1–10) are all indexed from the same t_0. The 30-frame score uses frames $t_0, \ldots, t_0 + 29$; the full-rollout score uses frames from t_0 to the end of that failure episode.

E Details on Real-Sim-Real Model Predictive Control

This appendix specifies the substage planner of Sec. 3.3. The motion note is cached text per family, reused for every test I_0, and is not a trained motion model. Items marked *(baseline)* are also given when the whole-skill baseline is evaluated. The cached motion note, the teleop strips, the six substages, and the video predictor are not.

Cached motion note. A VLM reads third-view teleoperation stills of the same task from the predictor's training set, at $0\%, 5\%, \ldots, 100\%$ (21 frames per episode): five grasp episodes/variations, and three drawer episodes. The test I_0 is a different recording. The first VLM pass writes how the hand moved, jaw timing, contact, withdrawal, and ordered Cartesian waypoint rules, plus one planner reminder. The reminder is a single sentence: what happens physically if those waypoints are skipped and one straight MOVE is sent into the object. For example, for the grasp, the wrist is still parked, the fingers meet the rim or the wall, and the cup is pushed or tipped. JUDGE scores only the per-substage success/fail test written by the second pass. The prompt fixes the camera frame ($+x$ image right, $+y$ toward the backdrop, $+z$ up, table top $z = 0.744\,\mathrm{m}$) *(baseline)*. In that frame the model writes ordered waypoint rules: clearance, then jaw gap, then wrist relative to the object. They contain no joint angles *(baseline)* and no substage list. The second VLM pass cuts the skill into six substages, each with an allowed primitive subset, a duration hint, ordered waypoints (`role`, geometric `where`, jaw, primitive; no metres), and a success/fail test for that substage alone.

Per-substage loop. **RECONSTRUCT** identifies the placement slot from I_0 on the five-point grid with $20\,\mathrm{cm}$ axial offsets; the baseline uses its locate call for the same purpose. The drawer occupies the center slot. Both planners receive slot coordinates and the initial fingertip position in meters in the world coordinate frame. The first substage starts from I_0, with the fingertip position computed by forward kinematics from the recorded initial joint configuration $\boldsymbol{\theta}_0$. Later substages use the last committed predicted frame and object and end-effector positions from the committed seed-0 simulator state. **PROPOSE** sees the current image, the matching slot's 21-frame teleop strip, and one seed-0 Isaac frame, and writes one short symbolic plan that realises this substage's waypoints in order, using only that substage's allowed action types. The metres in that plan come from the $20\,\mathrm{cm}$ slot on I_0 and, after the first substage, from the Isaac state of the committed prefix. **ACTION-IK** maps the action to 14-D with Piper DH inverse kinematics and $2\,\mathrm{cm}$ linear vias *(baseline executor)*. Every written opening, translation, and wrist field is executed. Isaac Sim rollouts the committed prefix followed by the candidate chunk under seeds 0–9. The predictor uses the current image, candidate action chunk, and ten time-aligned current-substage simulation clips to generate a real-domain future. **JUDGE** pass/fails this substage from the current image, predicted stills, and the seed-0 Isaac window, without teleop frames. A physics miss in Isaac (no contact, no lift, or no opening) fails the substage even if the prediction looks successful. Both need to be satisfied. Predicted stills shown to JUDGE are at $0, 20, 40, 60, 80, 100\%$ for both grasp and drawer. The seed-0 Isaac window is at $0, 50, 100\%$. The baseline instead scores Isaac seed-0 stills/simulation results at $0, 20, 40, 60, 80, 100\%$ against the VLM-authored whole-skill success text *(baseline)*. **REFINE**, at most three rounds, rewrites one action from the failed predicted and Isaac stills, the matching-slot teleop cropped to this substage, and the failure history; the revised action is converted by ACTION-IK, simulated, predicted, and judged again.

Motion notes m and substage specifications $\{q_j\}_{j=1}^{J}$ are cached per task family from separate training episodes, with $J = 6$. Each q_j defines allowed primitives and a local success criterion. RolloutPrefix rollouts the committed prefix A followed by candidate u from the initial scene under each seed, returning current-substage clips $\mathcal{S} = \{S^{(k)}\}_{k=0}^{9}$ and the terminal seed-0 state s'. Here, $\|$ denotes concatenation. The first passing candidate is committed and refinement stops. Its action chunk is appended to A, and its final predicted frame and simulator state initialize the next substage. If no candidate passes after the initial proposal and up to three refinement rounds, planning terminates without hardware execution. Otherwise, the complete trajectory is executed after all six substages pass both checks (Algorithm 1). Algorithm 1 describes normal deployment. For critic

Algorithm 1 Staged action planning through VPTwin-Core

Require: VLM G, simulator SIM, frozen predictor P, initial image I_0, initial joint configuration $\boldsymbol{\theta}_0$, task ℓ_{task}, same-task training demonstrations $\mathcal{D}$
Ensure: Accepted 14-D trajectory A and verdict b; empty trajectory on rejection

$(m, \{q_j\}_{j=1}^{J}) \leftarrow$ CACHEDMOTIONANDSTAGES$(G, \mathcal{D}, \ell_{\text{task}})$
$s_0 \leftarrow$ RECONSTRUCT$(G, I_0, \ell_{\text{task}}, \boldsymbol{\theta}_0)$
$s \leftarrow s_0$; $I \leftarrow I_0$; $A \leftarrow [\,]$
for $j = 1, \ldots, J$ **do**
  $\mathcal{H} \leftarrow [\,]$; $b_j \leftarrow$ false
  $d_j \leftarrow$ MATCHDEMO$(\mathcal{D}, q_j, s)$
  $\alpha \leftarrow$ PROPOSE$(G, I, d_j, \mathrm{Render}(s), m, q_j)$
  for $r = 0, \ldots, 3$ **do** ▷ Initial proposal plus up to three refinements
    $u \leftarrow$ ACTIONIK(α)
    $(\mathcal{S}, s') \leftarrow$ ROLLOUTPREFIX(SIM, $s_0, A, u, \{0, \ldots, 9\}$)
    $\widehat{V} \leftarrow P(I, u, \mathcal{S})$
    $(b_{\text{sim}}, b_{\text{vp}}, f) \leftarrow$ JUDGE$(G, q_j, I, S^{(0)}, \widehat{V})$
    if $b_{\text{sim}} \wedge b_{\text{vp}}$ **then**
      $b_j \leftarrow \text{true}$; **break**
    end if
    $\mathcal{H} \leftarrow \mathcal{H} \,\|\, [(\alpha, f)]$
    if $r < 3$ **then**
      $c \leftarrow (u, \widehat{V}, S^{(0)}, s')$
      $\alpha \leftarrow$ REFINE$(G, I, d_j, m, q_j, c, \mathcal{H})$
    end if
  end for
  if $\neg b_j$ **then**
    return $([\,], \text{false})$ ▷ No hardware execution
  end if
  $A \leftarrow A \,\|\, u$
  $I \leftarrow \mathrm{LastFrame}(\widehat{V})$; $s \leftarrow s'$
end for
EXECUTEONROBOT(A)
return (A, true)

evaluation, planning continues through all six substages even if one fails: passing substages use their first passing candidate, and failed substages use the candidate the VLM ranks closest to success. The resulting complete trajectory is executed on hardware while retaining its rejection verdict.

Relative to SIMPACT. The baseline samples ten whole-skill plans from I_0, allows up to three whole-skill revisions, and judges seed-0 simulation stills without teleoperation guidance or video prediction. Both planners receive the initial image, metric slot coordinates, initial fingertip position, primitive schema, and whole-skill success criteria.

Critic agreement. SIMPACT accepts a plan when a candidate passes its simulation check. VPTwin accepts a plan only when all six substages pass both simulation and video-prediction checks. Accepted plans are deployed normally; rejected plans are separately executed as described above to measure critic agreement. We report hardware success and critic precision, recall, and F1 using outcomes:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{F1} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \tag{6}$$

TP (True Positive), FP (False Positive), TN (True Negative), FN (False Negative). TP and FP denote accepted plans that succeed and fail on hardware, respectively; TN and FN denote rejected plans that fail and succeed on hardware, respectively. All rejected plans fail on hardware in this evaluation. The policy $\pi_{0.5}$ has no acceptance decision. We therefore compute precision, recall, and F1 using observed outcomes for both accepted and rejected plans. The policy $\pi_{0.5}$ provides no acceptance decision, so its acceptance and critic metrics are not applicable. For VPTwin on *Grasp Cup*, three of seven planning trials are accepted and executed, and all three succeed, giving an acceptance rate of $3/7$ and precision of 1.00.

F FID AND FVD EVALUATION PROTOCOL

We evaluate full-length test episodes at 128×128 resolution. Within each task family, predictions from all test episodes are pooled separately for each model and compared against the same ground-truth set. Baseline and VPTwin use identical test episodes, frame indices, and clip boundaries.

FID. FID (Heusel et al., 2017) uses 2048-dimensional Inception-v3 features extracted with TorchMetrics and its torch-fidelity backend. All frames enter the extractor as uint8 images and are internally resized to 299×299.

FVD. FVD (Unterthiner et al., 2018) uses pre-softmax features from the Kinetics-400-pretrained I3D detector distributed with StyleGAN-V in TorchScript format. We extract 16-frame clips with stride 8 within each episode, adding a final clip ending at the last frame if the sliding windows do not cover the episode's end. Pixel values are scaled to $[-1, 1]$, and the detector internally resizes frames to 224×224. Clips never cross episode boundaries.

Aggregation and sample counts. For each task family, we first average scores over five independent training seeds. Table 1 reports the mean and standard deviation across these family-level averages: six families in the measured-catalog setting and four in the VLM-authored setting. Table 10 lists the sample counts shared by both models and the ground-truth reference. Counts vary across families, and overlapping clips are not independent; comparisons use matched samples within each family.

Context-ablation seeds and aggregation. For the dest-count ablation, $k \in \{1, 3, 5, 7, 9\}$ uses evaluation seeds $0, 1, 2$, while $k = 0$ uses evaluation seed 0. The $k = 10$ result reuses the existing full-test rollout, without additional evaluation-seed averaging. The failure-only ablation produces one evaluation per k. Metrics are computed separately for each task family, checkpoint, and evaluation seed: FID pools the family's frames, and FVD pools its 16-frame clips.

G PROMPT EXAMPLES

The following is the prompt sent to the VLM for scene parsing.

```
Look at these photographs. Reason step by step (chain of thought) about
    what is on the table besides the given table, floor, and black cloth.

For each distinct object: what it looks like; whether anything covers it
    or sits on it; whether an interior is visible; whether one part can
    move relative to another. Watch which part stays on the table and
    which part moves. Do not model the table, floor, black cloth, robot,
    camera, or the red marks on the tabletop.

Do not assume a catalog. Only write what the photographs support. Do not
    list the red marks on the tabletop as objects; they are already on
    the given table. Use the centre mark as the visual origin only.

After each Image 01..24 (0% of that episode) there are later photographs
    from that same real episode (every 20% through 100%, with additional
    70%). They are not layout images. Use them for object identity,
    covering, interiors, and how parts move. Do not use them to decide
    where objects sit at 0%. If something is hidden in a 0% photograph
    and later photographs of that episode reveal it, it still belongs in
    that 0% layout (under or inside whatever covered it). A later
    photograph of an empty table is a later moment; it does not mean
    those objects were absent at 0%. If several 0% photographs show the
    same kind of covering (the same sheet over a similar raised shape),
    give the thing underneath one object id even when one episode never
    uncovers it. Do not invent a second anonymous support for an episode
    whose later frames go empty. 3 extra photographs come from another
    episode of the same task; also not layout images.
```

```
Write the reasoning in prose first. Then fence one JSON file:

'''json scene_parse.json
{
  "objects": [
    {
      "id": "obj_1",
      "appearance": "what it looks like in the photographs",
      "parts": [{"id": "part_1", "appearance": "...", "motion": "
    stays_put|translates_in_plane|lifts_off|rotates|flexes|unknown"}],
      "softbody": "none|thin_shell|folded_pack",
      "relations": [{"other_id": "obj_2", "relation": "on|under|covering|
    covered_by|inside|contains|attached_to|moves_relative_to|rests_on"}],
      "notes": "occlusion, interior if visible, how parts move"
    }
  ],
  "per_image": [
    {"image": 1, "visible": ["obj_1"], "present": ["obj_1", "obj_2"]}
  ]
}
''''

`image` is 1..24 matching Image 01..24 (the 0% layout photographs). Extra
     photographs do not get a `per_image` row.
`visible` is what you can see in that 0% photograph.
`present` is what is actually on the table at that 0% layout, including
    objects only revealed later in the same episode (covered, inside,
    behind). `present` must include every `visible` id. Do not drop a
    covered object from `present` because it is missing from `visible`.
If an object flexes, `softbody` is exactly one of two kinds. `thin_shell`
     = one connected sheet (a drape, cover, or unfolded cloth); its
    thickness is the fabric thickness. `folded_pack` = a compact folded
    towel or cloth pack sitting on the table. Do not unfold it into plies
     and do not describe stacked sheets. A folded pack is approximated
    later as one soft rectangular block (volume) or one midplane surface
    whose thickness is the pack height (how tall the pack sits), not a
    single-ply cloth thickness. Rigid objects use `none`. Do not invent a
     third kind.
```

The following prompt was sent to the VLM to derive task instructions and success criteria from real demonstration stills. Additional information will be released when the repository goes public.

```
These photographs are real third-view stills. They are grouped by episode
    .

Image 01..24 is 0% of that episode. Immediately after each one you will
    see later stills from that same real episode at every 20% through
    100%. Use each group as one timeline.

For each Image 01..24 group, reason step by step about the interaction
    from 0% to 100%: what starts on the table, what the arm does, what
    moves or stays, and how the table looks at the end. The arm may
    appear; describe what it does. Do not reconstruct meshes.

Some groups show the same kind of interaction, only starting at a
    different place on the table. Merge those into one kind. Do not
    invent names from a catalog. Refer to groups as Image 01, Image 02,
    and so on.

Then fence one JSON file. `success` is one paragraph: when this kind of
    episode would count as successful, written from what the photographs
    show.

'''json success.json
```

````
{
  "episodes": [
    {"image": 1, "interaction": "what happens from 0% to 100% in this
    episode"}
  ],
  "kinds": [
    {
      "id": "kind_1",
      "images": [1, 2],
      "interaction": "what these episodes share",
      "success": "one paragraph: this kind is successful when ..."
    }
  ]
}
```

Every image 1..24 must appear in exactly one kind's `images` list.
````
````

Table 9: Definition of tasks. For each manipulation task, we list the corresponding instruction and success criteria. Instruction and success conditions are GPT-6 Astra text authored from multi-time real-video stills. The *Grasp Cup* row describes the return-near-start variant retained in the VLM-authored set; the measured-catalog set also includes cup-relocation variants.

Task	**Instruction**	**Success Condition**
Remove Cloth	Remove a white cloth from over a cup while leaving the cup standing on the table. The arm grips and gathers the fabric beside the covered form, then pulls or lifts the cloth clear rather than carrying away the covered object.	This kind is successful when the arm removes the covering cloth completely, exposing the green cup and leaving it standing on its base near the previously covered location. The cloth must no longer cover or hang over the cup at the end, and the cup must remain on the table rather than leave with the cloth. The final view should show the uncovered cup, the surrounding bare tabletop, and the unchanged red markers after the arm withdraws.
Place Cube into Drawer	Open a white drawer using its yellow front knob, pick up the cube from the table, place it inside the open compartment, and close the drawer.	This kind is successful when the drawer is pulled open far enough to receive the cube, the arm transfers the cube from the tabletop into the compartment, and the drawer is then pushed fully closed. The cube should no longer be at its original outside location or sitting on the unit's top. The final view should show the closed drawer unit standing in place, with the cube enclosed following the visible placement sequence and the surrounding table markers undisturbed.
Open and Close Drawer	Pull the initially closed drawer outward to expose its empty compartment, then push it closed again without transferring another object.	This kind is successful when the arm produces a visible opening of the drawer, revealing the empty compartment, and subsequently returns the drawer front to its closed position against the housing. No object needs to be added or removed. At the end, the unit should remain standing in approximately its original place, the drawer should no longer be extended, and the surrounding tabletop should be unchanged.
Pickup Cup and Cloth	Grasp the raised portion of a white cloth-covered bundle and lift the entire bundle away, leaving no uncovered object behind. Its concealed contents are not separately revealed in these photographs.	This kind is successful when the arm takes hold of the cloth-covered raised portion and removes the whole bundle from the tabletop, including the trailing fabric. The former covered area should be clear at the end, with no cup or other loose object left standing there and no cloth remaining on the table. The visible endpoint is a bare tabletop apart from its stationary red markers; identifying the concealed contents is not necessary to judge that outcome.
Wipe Table	Press a blue cloth against the table with the gripper held nearly horizontal and slide it through short local strokes, then lift the arm away while leaving the cloth on the table.	This kind is successful when the arm maintains contact that presses the blue cloth against the tabletop and moves the cloth through a visible short wiping stroke or back-and-forth motion. It should then withdraw, leaving the cloth resting on the table in roughly the same local area rather than carrying it away. Small shifts, changes in outline, and rearranged folded edges are consistent with the photographs, while the fixed markers remain undisturbed. The supported success criterion is the contact-and-sliding action, not a claim that unseen dirt has been removed.
Grasp Cup	Grasp a standing cup, lift it off the table, and set it down near its original position.	This kind is successful when the gripper takes hold of the cup, lifts it clear of the tabletop, and subsequently returns it to a stable standing position near where it began. The cup may tilt while held or temporarily leave the image, but it should finish upright on its base after the gripper releases and withdraws. The final table arrangement should remain essentially the starting arrangement, apart from a small placement difference, with the red markers unchanged.

Table 10: Per-family sample counts for FID and FVD. n denotes test episodes, N_{frame} the number of frames used for FID, and N_{clip} the number of clips used for FVD. Counts are identical for Baseline, VPTwin, and ground truth.

Task family	n	N_{frame}	N_{clip}
Measured-catalog setting			
Wipe Table	10	1369	166
Grasp Cup	13	3346	412
Pickup Cup and Cloth	10	1207	146
Remove Cloth	5	713	87
Open and Close Drawer	10	2463	302
Place Cube into Drawer	6	2471	305
VLM-authored setting			
Wipe Table	8	1116	135
Grasp Cup	3	706	87
Pickup Cup and Cloth	8	943	114
Open and Close Drawer	8	2004	246

Table 11: Dest-count ablation, family-averaged (mean $\pm$ std over task families). The input retains ten conditioning slots. For $k > 0$ the k sampled dests are cycled into every slot (no black frames); k=0 is all zeros. $k \in \{1, 5\}$ averages three evaluation seeds, while $k = 10$ uses all ten rollouts.

Metric	**k=0**	**k=1**	**k=5**	**k=10**
Measured-catalog (548), 6 families				
MSE ↓	0.101 ± 0.048	0.007 ± 0.002	0.007 ± 0.002	0.007 ± 0.002
LPIPS ↓	0.579 ± 0.030	0.289 ± 0.055	0.286 ± 0.055	0.284 ± 0.056
FID ↓	363.3 ± 18.6	267.8 ± 36.5	266.6 ± 37.5	265.6 ± 36.9
PSNR ↑	16.71 ± 2.74	28.36 ± 1.91	28.51 ± 1.89	28.57 ± 1.92
SSIM ↑	0.547 ± 0.110	0.796 ± 0.057	0.797 ± 0.056	0.798 ± 0.058
UIQI ↑	0.053 ± 0.026	0.277 ± 0.040	0.281 ± 0.039	0.283 ± 0.040
FVD ↓	2884.7 ± 383.8	2028.3 ± 273.2	2015.7 ± 262.2	2014.5 ± 264.7
VLM-authored (277), 4 families				
MSE ↓	0.109 ± 0.059	0.006 ± 0.002	0.007 ± 0.002	0.006 ± 0.002
LPIPS ↓	0.585 ± 0.051	0.263 ± 0.046	0.264 ± 0.045	0.259 ± 0.048
FID ↓	360.6 ± 23.3	261.0 ± 40.8	262.1 ± 38.7	259.1 ± 42.3
PSNR ↑	16.78 ± 4.15	28.84 ± 1.77	28.74 ± 1.87	28.96 ± 1.83
SSIM ↑	0.546 ± 0.152	0.807 ± 0.060	0.806 ± 0.060	0.809 ± 0.060
UIQI ↑	0.059 ± 0.042	0.291 ± 0.035	0.291 ± 0.035	0.294 ± 0.039
FVD ↓	3016.2 ± 724.2	2091.2 ± 373.6	2155.0 ± 373.7	2131.0 ± 340.6

Table 12: Fail-dest ablation on the measured catalog (548). Concat receives the k lowest-index failed dests, cycled into all 10 slots. Baseline is the action-only model on the same episodes. k=3 is marked * when $n \leq 2$. Bold is the better of baseline and concat. Δ is concat minus baseline. n is the k=1 episode count.

		k=1			**k=3**		
Family	**Metric**	**B**	**C**	**Δ**	**B**	**C**	**Δ**
Pickup Cup and Cloth n=6	MSE ↓	0.0040	**0.0036**	−0.0004	0.0042	**0.0036**	−0.0006
	LPIPS ↓	**0.260**	0.277	+0.017	**0.281**	0.293	+0.012
	FID ↓	**242.5**	242.8	+0.3	260.2	**254.4**	−5.8
	PSNR ↑	30.28	**30.68**	+0.40	30.05	**30.82**	+0.77
	SSIM ↑	**0.865**	0.862	−0.003	0.864	0.864	0.000
	UIQI ↑	**0.313**	0.292	−0.021	**0.290**	0.281	−0.009
	FVD ↓	1727.7	**1582.3**	−145.4	1956.3	**1875.5**	−80.8
Wipe Table n=9	MSE ↓	**0.0143**	0.0227	+0.0084	**0.0160**	0.0264	+0.0104
	LPIPS ↓	**0.257**	0.299	+0.042	**0.252**	0.289	+0.037
	FID ↓	**256.4**	301.8	+45.4	**253.9**	294.2	+40.3
	PSNR ↑	**25.93**	23.32	−2.62	**25.05**	22.67	−2.37
	SSIM ↑	**0.807**	0.800	−0.007	**0.802**	0.794	−0.008
	UIQI ↑	**0.293**	0.281	−0.012	**0.291**	0.284	−0.007
	FVD ↓	**2316.0**	3066.8	+750.8	**2401.7**	2885.5	+483.8
Grasp Cup n=5	MSE ↓	0.0140	**0.0062**	−0.0078	0.0168	**0.0066**	−0.0102
	LPIPS ↓	0.325	**0.293**	−0.032	0.329	**0.288**	−0.041
	FID ↓	297.4	**262.7**	−34.7	307.1	**278.2**	−28.9
	PSNR ↑	25.05	**28.21**	+3.16	23.91	**27.93**	+4.02
	SSIM ↑	0.765	**0.794**	+0.029	0.757	**0.787**	+0.030
	UIQI ↑	0.227	**0.267**	+0.040	0.224	**0.264**	+0.040
	FVD ↓	2810.6	**2555.0**	−255.6	2763.8	**2615.0**	−148.8
Open and Close Drawer n=7	MSE ↓	0.0250	**0.0092**	−0.0158	0.0264*	**0.0079***	−0.0185*
	LPIPS ↓	0.391	**0.344**	−0.047	0.386*	**0.335***	−0.051*
	FID ↓	**300.7**	306.9	+6.2	331.5*	**326.2***	−5.3*
	PSNR ↑	22.38	**26.46**	+4.08	21.91*	**27.05***	+5.14*
	SSIM ↑	0.687	**0.721**	+0.034	0.683*	**0.727***	+0.044*
	UIQI ↑	0.204	**0.240**	+0.036	0.200*	**0.249***	+0.049*
	FVD ↓	2111.2	**2009.8**	−101.4	2330.5*	**2205.3***	−125.2*
Place Cube into Drawer n=4	MSE ↓	0.0181	**0.0091**	−0.0090	0.0179	**0.0088**	−0.0091
	LPIPS ↓	0.422	**0.360**	−0.062	0.422	**0.357**	−0.065
	FID ↓	**299.6**	304.6	+5.0	**303.5**	305.9	+2.4
	PSNR ↑	23.48	**26.48**	+3.00	23.49	**26.65**	+3.16
	SSIM ↑	0.710	**0.724**	+0.014	0.707	**0.727**	+0.020
	UIQI ↑	0.224	**0.236**	+0.012	0.218	**0.239**	+0.021
	FVD ↓	2308.1	**2281.2**	−26.9	**2271.4**	2316.7	+45.3
Remove Cloth n=5	MSE ↓	0.0135	**0.0103**	−0.0032	**0.0070**	0.0081	+0.0011
	LPIPS ↓	**0.344**	0.359	+0.015	**0.314**	0.331	+0.017
	FID ↓	**309.5**	316.0	+6.5	**319.8**	326.1	+6.3
	PSNR ↑	**26.43**	26.17	−0.26	**27.69**	27.08	−0.61
	SSIM ↑	**0.790**	0.787	−0.003	**0.811**	0.800	−0.011
	UIQI ↑	**0.228**	0.204	−0.024	**0.243**	0.218	−0.025
	FVD ↓	**1663.2**	1780.6	+117.4	**1526.2**	1567.5	+41.3

Table 13: Fail-dest ablation on the VLM-authored set (277). Same protocol as Table 12. Grasp Cup is a single episode at k=1. Wipe Table has no episode with three failures. Every k=3 cell that exists is one episode (*).

		k=1			k=3		
Family	**Metric**	**B**	**C**	**Δ**	**B**	**C**	**Δ**
Pickup Cup and Cloth n=5	MSE ↓	**0.0044**	0.0052	+0.0008	0.0059*	**0.0054***	−0.0005*
	LPIPS ↓	**0.252**	0.339	+0.087	**0.264***	0.307*	+0.043*
	FID ↓	**228.5**	253.1	+24.6	278.5*	**271.5***	−7.0*
	PSNR ↑	**29.82**	29.00	−0.81	28.29*	**28.69***	+0.41*
	SSIM ↑	**0.857**	0.846	−0.011	0.830*	**0.831***	+0.001*
	UIQI ↑	**0.305**	0.248	−0.057	**0.286***	0.263*	−0.023*
	FVD ↓	1876.4	**1478.6**	−397.8	2383.1*	**1838.4***	−544.7*
Wipe Table n=6	MSE ↓	**0.0211**	0.0279	+0.0068	—	—	—
	LPIPS ↓	**0.279**	0.296	+0.017	—	—	—
	FID ↓	**268.5**	303.4	+34.9	—	—	—
	PSNR ↑	**24.17**	23.04	−1.14	—	—	—
	SSIM ↑	**0.796**	0.792	−0.004	—	—	—
	UIQI ↑	**0.283**	0.278	−0.005	—	—	—
	FVD ↓	**2634.5**	3100.7	+466.2	—	—	—
Grasp Cup n=1*	MSE ↓	**0.0048**	0.0051	+0.0003	**0.0047***	0.0050*	+0.0003*
	LPIPS ↓	0.298	**0.283**	−0.015	0.298*	**0.282***	−0.016*
	FID ↓	332.8	**310.2**	−22.6	333.1*	**312.3***	−20.8*
	PSNR ↑	**29.25**	28.99	−0.27	**29.27***	29.03*	−0.24*
	SSIM ↑	0.801	0.801	0.000	0.801*	0.801*	0.000*
	UIQI ↑	0.248	**0.251**	+0.003	0.247*	**0.251***	+0.004*
	FVD ↓	3462.5	**2677.6**	−784.9	3292.9*	**2595.5***	−697.4*
Open and Close Drawer n=3	MSE ↓	0.0153	**0.0078**	−0.0075	0.0179*	**0.0080***	−0.0099*
	LPIPS ↓	0.372	**0.333**	−0.039	0.378*	**0.348***	−0.030*
	FID ↓	330.4	**320.2**	−10.2	362.6*	**357.4***	−5.2*
	PSNR ↑	24.20	**27.11**	+2.91	23.50*	**26.99***	+3.48*
	SSIM ↑	0.704	**0.728**	+0.024	0.688*	**0.722***	+0.034*
	UIQI ↑	0.219	**0.250**	+0.031	0.201*	**0.237***	+0.036*
	FVD ↓	**2053.0**	2067.2	+14.2	**2388.1***	2582.2*	+194.1*

Table 14: Per-family full-episode scores. Each cell is Baseline /- Ours. Baseline is action-only VPTwin; Ours is simulation-conditioned VPTwin (10 dests). Pixel metrics are mean ± std over test episodes. Higher PSNR, SSIM, and UIQI are better (↑); lower MSE, LPIPS, FID, and FVD are better (↓). The better value is bolded. These family scores are the entries averaged in Table 1.

Task family	n	MSE ↓	LPIPS ↓	FID ↓	FVD ↓	PSNR ↑	SSIM ↑	UIQI ↑
Measured-catalog (548), six task families								
Wipe Table	10	0.0137 ± 0.0153 / **0.0063 ± 0.0050**	0.252 ± 0.037 / **0.213 ± 0.014**	250.3 / **230.6**	2332.2 / **2194.8**	26.25 ± 3.46 / **28.93 ± 2.80**	0.812 ± 0.051 / **0.847 ± 0.026**	0.296 ± 0.033 / **0.336 ± 0.016**
Grasp Cup	13	0.0239 ± 0.0204 / **0.0067 ± 0.0032**	0.349 ± 0.051 / **0.277 ± 0.021**	278.5 / **254.2**	2644.5 / **2396.9**	23.54 ± 3.66 / **28.26 ± 2.33**	0.747 ± 0.048 / **0.794 ± 0.019**	0.216 ± 0.029 / **0.270 ± 0.013**
Pickup Cup and Cloth	10	0.0036 ± 0.0012 / **0.0026 ± 0.0010**	0.237 ± 0.049 / **0.223 ± 0.032**	225.0 / **216.5**	**1689.7** / 1700.6	30.73 ± 1.40 / **32.11 ± 1.56**	0.872 ± 0.019 / **0.876 ± 0.021**	0.324 ± 0.038 / **0.329 ± 0.034**
Remove Cloth	5	0.0132 ± 0.0162 / **0.0080 ± 0.0066**	0.337 ± 0.076 / **0.310 ± 0.038**	306.0 / **295.4**	**1633.7** / 1748.7	26.68 ± 4.06 / **28.20 ± 3.65**	0.793 ± 0.061 / **0.802 ± 0.049**	0.234 ± 0.046 / **0.240 ± 0.039**
Open and Close Drawer	10	0.0243 ± 0.0092 / **0.0090 ± 0.0016**	0.394 ± 0.028 / **0.337 ± 0.010**	**299.0** / 300.9	2147.3 / **1988.5**	22.48 ± 1.78 / **26.56 ± 0.75**	0.687 ± 0.021 / **0.728 ± 0.010**	0.206 ± 0.019 / **0.255 ± 0.009**
Place Cube into Drawer	6	0.0168 ± 0.0026 / **0.0074 ± 0.0010**	0.408 ± 0.025 / **0.343 ± 0.009**	297.2 / **295.7**	2122.6 / **2057.7**	23.82 ± 0.68 / **27.37 ± 0.59**	0.707 ± 0.008 / **0.741 ± 0.009**	0.220 ± 0.011 / **0.265 ± 0.012**
VLM-authored (277), four task families								
Wipe Table	8	0.0189 ± 0.0154 / **0.0069 ± 0.0064**	0.279 ± 0.050 / **0.215 ± 0.018**	265.9 / **234.0**	2622.7 / **2173.8**	24.47 ± 3.49 / **28.91 ± 3.33**	0.802 ± 0.054 / **0.844 ± 0.034**	0.286 ± 0.035 / **0.335 ± 0.024**
Grasp Cup	3	0.0222 ± 0.0164 / **0.0072 ± 0.0035**	0.347 ± 0.052 / **0.268 ± 0.027**	311.9 / **282.6**	3359.1 / **2575.7**	24.22 ± 5.48 / **27.91 ± 2.62**	0.759 ± 0.047 / **0.785 ± 0.021**	0.233 ± 0.028 / **0.262 ± 0.016**
Pickup Cup and Cloth	8	0.0037 ± 0.0015 / **0.0030 ± 0.0013**	0.233 ± 0.046 / **0.231 ± 0.035**	218.5 / **214.1**	1774.6 / **1765.8**	30.60 ± 1.62 / **31.56 ± 1.79**	0.866 ± 0.022 / **0.870 ± 0.023**	0.317 ± 0.042 / **0.320 ± 0.034**
Open and Close Drawer	8	0.0206 ± 0.0098 / **0.0072 ± 0.0009**	0.382 ± 0.028 / **0.322 ± 0.014**	314.4 / **305.6**	2043.6 / **2008.5**	23.26 ± 1.88 / **27.48 ± 0.56**	0.692 ± 0.023 / **0.736 ± 0.009**	0.212 ± 0.023 / **0.261 ± 0.010**

Table 15: Failure-rollout SSIM on the same 11 test pairs as Table 3. Failmix uses success–failure co-training; Success-only uses the 277 successful VLM-filtered episodes. Scores use a 30-frame prefix or the full rollout from the indicated start frame. Higher is better.

		30 frames		Full rollout	
Start	Task	Failmix	Success-only	Failmix	Success-only
First frame	Grasp Cup	**0.714**	0.704	**0.707**	0.700
	Open and Close Drawer	**0.674**	0.637	**0.624**	0.579
Perturbed frame	Grasp Cup	**0.713**	0.703	**0.707**	0.700
	Open and Close Drawer	**0.620**	0.590	**0.605**	0.558

Table 16: Full-rollout MSE, SSIM, LPIPS, and UIQI for Table 4. Same episodes and the same two settings.

	MSE ↓		SSIM ↑		LPIPS ↓		UIQI ↑	
Task	3→3	10→3	3→3	10→3	3→3	10→3	3→3	10→3
Wipe Table	0.0124	**0.0060**	0.823	**0.847**	0.197	**0.173**	0.312	**0.335**
Grasp Cup	0.0073	**0.0067**	0.792	**0.794**	0.244	**0.241**	0.266	**0.269**
Pickup Cup and Cloth	0.0028	**0.0026**	0.876	0.876	**0.173**	0.174	0.329	0.329
Remove Cloth	0.0117	**0.0068**	0.800	**0.806**	0.261	**0.252**	**0.248**	0.240
Open and Close Drawer	0.0112	**0.0090**	0.723	**0.730**	0.314	**0.304**	0.247	**0.255**
Place Cube into Drawer	0.0078	**0.0073**	0.741	**0.743**	0.315	**0.292**	0.262	**0.268**
Mean over tasks	0.0089	**0.0064**	0.793	**0.799**	0.251	**0.239**	0.277	**0.283**

Table 17: Pooled FID and FVD for Table 4. FID uses every predicted frame (Inception-2048). FVD uses 16-frame clips with stride 8. Both are one score per task, not a mean of per-episode scores. Lower is better.

	FID ↓		FVD ↓	
Task	3→3	10→3	3→3	10→3
Wipe Table	231.90	**230.12**	2182.2	**2136.6**
Grasp Cup	**255.10**	255.43	2431.9	**2391.0**
Pickup Cup and Cloth	220.18	**217.52**	1709.4	**1693.5**
Remove Cloth	296.15	**292.55**	**1653.2**	1781.9
Open and Close Drawer	309.82	**307.23**	2087.1	**2011.7**
Place Cube into Drawer	294.68	**294.55**	2091.2	**2070.8**
Mean over tasks	267.97	**266.23**	2025.8	**2014.2**