DO ROBOTIC WORLD MODELS REALLY FOLLOW ACTIONS? DIAGNOSING AND ALIGNING ACTION-CONDITIONED GENERATION FOR POLICY LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Action-conditioned world models increasingly serve as learned simulators for robotic policy improvement, yet their reliability rests on an unverified assumption: generated futures faithfully reflect the consequences of arbitrary valid actions. Systematic evaluation against action-specific ground truth remains limited beyond expert demonstrations. To address this gap, we introduce WorldEcho, which evaluates action following over a broader action distribution through visual integrity and $\mathrm{SE}(3)$ end-effector trajectory alignment. Our diagnosis shows that expert-trained models perform better on demonstrated actions but struggle with off-expert trajectories, producing motion inconsistent with the commanded actions or visually invalid rollouts. To mitigate these failures, we propose WorldSync, which broadens action coverage, grounds intermediate video representations in action-induced robot dynamics through an Action-Forcing Expert, and aligns predicted and ground-truth changes under action interventions. Experiments show that WorldSync reduces integrity-gated error on RoboTwin and, when used for iterative policy improvement, yields higher success rates in simulation and on real robots. Project page: `https://worldecho-worldsync.github.io/`.

1 INTRODUCTION

Large-scale pretraining has improved the success rates and generalization of Vision-Language-Action (VLA) models (Zitkovich et al., 2023; O'Neill et al., 2024; Black et al., 2025) and World Action Models (WAM) (Wu et al., 2024; Li et al., 2026a; Ye et al., 2026). Task-specific online post-training can further improve downstream performance (Physical Intelligence et al., 2025; Pan et al., 2026; Guo et al., 2026a), but requires costly real-world interaction (Xiao et al., 2026). To reduce this burden, action-conditioned world models (AC-WMs) (Zhu et al., 2025; Guo et al., 2026b) serve as learned simulators for policy evaluation (Quevedo et al., 2026; Li et al., 2025b) and provide synthetic experience for post-training (Guo et al., 2026a; Yu et al., 2026; Xiao et al., 2026; Li et al., 2025a; Jiang et al., 2026b). Their use as simulators (Li et al., 2026b) raises a critical question: **do generated futures faithfully reflect the consequences of the queried actions?**

Recent studies report improved policy performance from world-model-generated experience (Guo et al., 2026a; Jiang et al., 2026b). Such gains demonstrate downstream utility, but do not directly measure whether generated futures faithfully reflect the queried actions. Synthetic experience may benefit policy learning through visual augmentation even when its action responses are inaccurate. Direct evaluation must therefore compare model predictions with the consequences of executing the same actions from the same initial state. Learned policies may deviate from expert behavior during evaluation and improvement, so these comparisons must also cover off-expert actions (Ross et al., 2011; Yu et al., 2026; Guo et al., 2026a). Yet existing benchmarks mainly assess perceptual and semantic quality, reference similarity, or downstream executability (Hu et al., 2025; Shang et al., 2026a; Jiang et al., 2026a). Although recent studies probe out-of-distribution or failure-inducing actions (Quevedo et al., 2026; Yang et al., 2026a), systematic comparisons against action-specific ground truth across broad numerical action queries remain limited.

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.

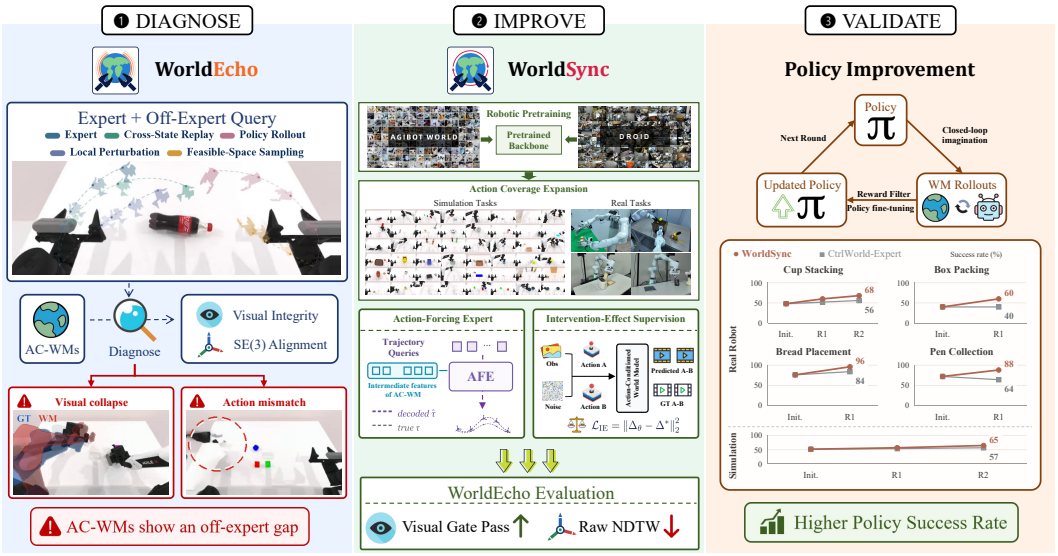

Figure 1: Diagnose–improve–validate pipeline. WorldEcho evaluates action following, WorldSync aligns action-conditioned generation, and simulation and real-robot experiments assess downstream policy improvement.

To address this gap, WorldEcho evaluates action following on demonstrated queries and four complementary off-expert categories (Figure 1). Each query is replayed in RoboTwin to obtain its ground-truth future, including unsuccessful outcomes. We assess generated rollouts through visual integrity and $\mathrm{SE}(3)$ alignment between generated and reference end-effector trajectories. Relative to demonstrated queries, off-expert queries increase integrity-gated error across six expert-trained models (Figure 5), with both visually plausible action mismatches and visual degradation (Figure 2).

These findings motivate WorldSync, which combines distributional coverage, representational grounding, and intervention-effect alignment. Simulated expert and off-expert trajectories broaden action support, while a small amount of target-domain real-world data preserves visual fidelity. Because coverage alone does not ground intermediate video representations in action consequences, an *Action-Forcing Expert* (AFE) predicts future end-effector state changes. AFE supervises individual rollouts but does not explicitly constrain changes across actions. *Intervention-Effect* (IE) supervision therefore aligns predicted and ground-truth changes across paired trajectories with shared observations and different actions.

On RoboTwin (Chen et al., 2026b), WorldSync with and without additional robotic pretraining achieves the two lowest integrity-gated error point estimates among evaluated configurations (Table 1). When used for policy improvement, it achieves higher success rates than expert-trained CtrlWorld in simulation and across four real-robot tasks on three platforms (Figure 6).

Our contributions are as follows. (1) WorldEcho measures visual integrity and action-specific $\mathrm{SE}(3)$ trajectory alignment across demonstrated and off-expert queries. (2) Our evaluation characterizes off-expert failures across six world models, distinguishing action-inconsistent motion from visual degradation. (3) WorldSync improves action following through expanded coverage, AFE, and IE supervision, with downstream evaluation of policy improvement.

2 Related Work

2.1 Action-Conditioned Robotic World Models

AC-WMs advance action control through architecture (Zhu et al., 2025; Guo et al., 2026b; Zheng et al., 2026), broader pretraining (Gao et al., 2026; Wu & Gao, 2026; Huang et al., 2026b), and spatially grounded representations (Wu & Gao, 2026; Chen et al., 2026c;a; Yang et al., 2026b). Per-

turbations and counterfactual behaviors extend beyond standard replay, but provide mostly indirect or coarse-grained evidence of action following rather than ground-truth motion for each command (Zhu et al., 2025; Gao et al., 2026; Huang et al., 2026b; Feingold et al., 2026; Yang et al., 2026a).

2.2 World Models for Policy Evaluation and Improvement

AC-WMs support policy evaluation (Quevedo et al., 2026; Li et al., 2025b; Jeon et al., 2026; Ma et al., 2026) and improvement using synthetic experience (Yu et al., 2026; Xiao et al., 2026; Li et al., 2025a; Jiang et al., 2026b). To handle policy queries beyond expert-dominated data, systems use exploratory or corrective interactions (Yin et al., 2026; Li et al., 2026b; Xiao et al., 2026), generation filtering, or joint policy and world-model updates (Jiang et al., 2026b; Guo et al., 2026a; Liu et al., 2026). These downstream gains motivate direct evaluation of the queried actions' consequences.

2.3 Robotic World Model Evaluation

Benchmarks assess motion and physical consistency (Hu et al., 2025; Shang et al., 2026a;b), or recover actions to test executability (Jiang et al., 2026a; Fan et al., 2026). Most closely related, MiraBench shows that visually plausible predictions may ignore failure-inducing perturbations or exhibit optimism bias (Yang et al., 2026a). Beyond image-space motion, recovered-action execution, and task-level failure judgments, WorldEcho aligns generated end-effector trajectories with simulator-replayed references in $\mathrm{SE}(3)$ across broader numerical action queries.

3 Method

3.1 Problem Formulation

Given a current multi-view observation o_0, language instruction c, and H-step numerical action sequence $a_{1:H}$, an action-conditioned world model generates a future multi-view video

$$\hat{I}_{1:H} \sim p_\theta(\cdot \mid o_0, c, a_{1:H}). \tag{1}$$

Executing the same actions from the corresponding initial environment state gives the ground-truth future $I^{\mathrm{GT}}_{1:H}$. For a trajectory extractor Φ, the generated and reference end-effector trajectories are $\hat{\tau} = \Phi(\hat{I}_{1:H})$ and $\tau^{\mathrm{GT}} = \Phi(I^{\mathrm{GT}}_{1:H})$. Our goal is to evaluate and improve visual integrity and agreement with the action-specific reference trajectories beyond expert demonstrations.

3.2 Motivation

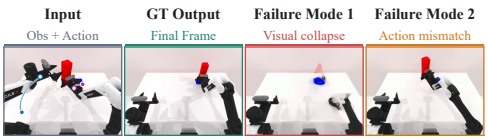

Figure 2: Off-expert failures. Left to right: input and actions, ground-truth final frame, visual collapse, and plausible but action-inconsistent motion.

Following prior work (Zhu et al., 2025; Guo et al., 2026b), we fine-tune Cosmos-Predict2.5 (NVIDIA et al., 2025) on RoboTwin expert trajectories (Mu et al., 2025; Chen et al., 2026b). From the same initial state, we compare predictions for expert and feasible off-expert actions with matched simulator futures. Expert actions are replayed accurately, but off-expert predictions lose visual integrity or show plausible, action-inconsistent motion (Figure 2). These failures motivate evaluating both visual integrity and action-induced motion beyond expert demonstrations.

3.3 WorldEcho

WorldEcho evaluates visual integrity and trajectory alignment against action-specific simulator references across demonstrated and off-expert queries (Figure 3).

3.3.1 Action Query Construction

Demonstrated Action replays the expert sequence for the current observation as an in-distribution baseline. Four off-expert categories probe beyond this support. *Cross-State Replay* transfers expert

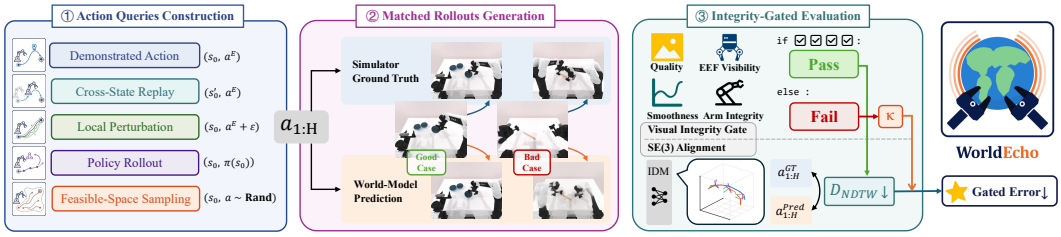

Figure 3: WorldEcho evaluation across five action-query categories, combining visual integrity with SE(3) end-effector trajectory alignment against matched simulator references.

actions to another initial state, testing context-dependent effects including failure. *Local Perturbation* applies bounded changes to expert actions to test sensitivity to nearby alternatives. *Policy Rollout* uses learned-policy actions to capture deviations during evaluation and improvement. Beyond these policy-dependent queries, *Feasible-Space Sampling* draws valid actions from the broader action space to test controllability. All off-expert queries undergo feasibility filtering before replay in RoboTwin.

3.3.2 Visual Integrity Assessment

We assess four aspects of visual integrity so trajectory comparisons concern valid depictions of the robot and scene. Following WorldArena (Shang et al., 2026a), image quality q uses MUSIQ (Ke et al., 2021), while motion smoothness m uses frame-interpolation consistency (Zhang et al., 2024). With $\mathbb{I}$ denoting an indicator, thresholds τ_q, τ_m give $G_{\text{quality}} = \mathbb{I}[q \geq \tau_q]$ and $G_{\text{motion}} = \mathbb{I}[m \geq \tau_m]$. For end-effector visibility, G_{EEF} passes when either SAM-based video tracking or framewise text-prompted detection meets its requirement (Carion et al., 2026). Arm integrity G_{arm} is assessed by a vision-language evaluator that detects blurred, broken, or disappearing arms (Bai et al., 2025). Using the same rules and thresholds for all models, we define

$$G_{\text{vis}} = G_{\text{quality}} \wedge G_{\text{motion}} \wedge G_{\text{EEF}} \wedge G_{\text{arm}}. \tag{2}$$

3.3.3 End-Effector Trajectory Alignment

Visual validity does not ensure that motion follows the queried actions, so we compare generated and ground-truth end-effector trajectories. We use an inverse dynamics model (IDM) Φ (Tan et al., 2026a) to extract end-effector positions $p_t^e \in \mathbb{R}^3$ and rotations $R_t^e \in \mathrm{SO}(3)$ from each video, where $e \in \{L, R\}$ denotes the left or right end effector. Across all 1,750 WorldEcho reference videos, frame-aligned IDM predictions have mean Euclidean position and geodesic rotation errors of 3.28 mm and 1.42° against simulator states, averaged over frames and both arms. For generated frame i and reference frame j, the local pose discrepancy is

$$d_e(i,j) = \left[w_p^2 \|\hat{p}_i^e - p_j^{e,\mathrm{GT}}\|_2^2 + w_R^2 d_{\mathrm{SO}(3)}^2(\hat{R}_i^e, R_j^{e,\mathrm{GT}}) \right]^{1/2}, \tag{3}$$

where hats and GT denote generated and reference poses, and w_p, w_R weight translation and rotation. The rotation distance is

$$d_{\mathrm{SO}(3)}(R_1, R_2) = \arccos(\mathrm{clip}(\xi, -1, 1)), \qquad \xi = \frac{\mathrm{tr}(R_1^\top R_2) - 1}{2}, \tag{4}$$

with tr denoting the trace and clip clamping to $[-1, 1]$. To accommodate temporal misalignment, we use pose-aware normalized dynamic time warping (NDTW) (Sakoe & Chiba, 1978; Salvador & Chan, 2007; Hu et al., 2025; Shang et al., 2026a):

$$D_{\mathrm{NDTW}} = \frac{1}{|\mathcal{A}|} \sum_{e \in \mathcal{A}} \frac{1}{|\pi_e|} \sum_{(i,j) \in \pi_e} d_e(i,j), \tag{5}$$

where $\mathcal{A}$ contains valid end effectors and π_e is the optimal warping path for e. We normalize by path length, but not by the reference trajectory's spatial extent, to preserve absolute pose error and avoid amplifying pose-estimation noise for short motions.

3.3.4 Evaluation Protocol

To retain diagnostic information, we compute raw NDTW D_n^{NDTW} for every query n, including visually invalid rollouts. Given its visual-gate result G_n, the official error instead assigns a fixed penalty κ to visual failures:

$$S_n = \begin{cases} D_n^{\mathrm{NDTW}}, & G_n = 1, \\ \kappa, & G_n = 0. \end{cases} \tag{6}$$

We first average S_n within each task, then equally across tasks, so unequal query counts do not dominate the ranking. We also report visual pass rate and raw NDTW. Algorithm 1 summarizes evaluation; Appendices F.1, F.2, and F.3 provide query-category results, metric distributions, and human comparisons, respectively.

Algorithm 1 Integrity-Gated Action-Following Evaluation

Require: AC-WM $\mathcal{W}_\theta$, action queries $\mathcal{Q}$, failure penalty κ

for each query n: $(o_0, c, a_{1:H}, I_{1:H}^{\mathrm{GT}}) \in \mathcal{Q}$ **do**
  Generate $\hat{I}_{1:H} \sim p_\theta(\cdot \mid o_0, c, a_{1:H})$
  Evaluate $G_n = G_{\mathrm{vis}}(\hat{I}_{1:H})$
  Extract $\hat{\tau} = \Phi(\hat{I}_{1:H})$ and $\tau^{\mathrm{GT}} = \Phi(I_{1:H}^{\mathrm{GT}})$
  Compute pose-aware NDTW error D_n^{NDTW}
  Set $S_n \leftarrow D_n^{\mathrm{NDTW}}$ if $G_n = 1$; otherwise $S_n \leftarrow \kappa$
end for
Aggregate $\{S_n\}$ within each task and then across tasks
return task-macro integrity-gated error, visual pass rate, and raw NDTW errors

3.4 WorldSync

To address these failures, WorldSync combines expanded action coverage with AFE feature grounding and IE supervision of action-dependent changes (Figure 4).

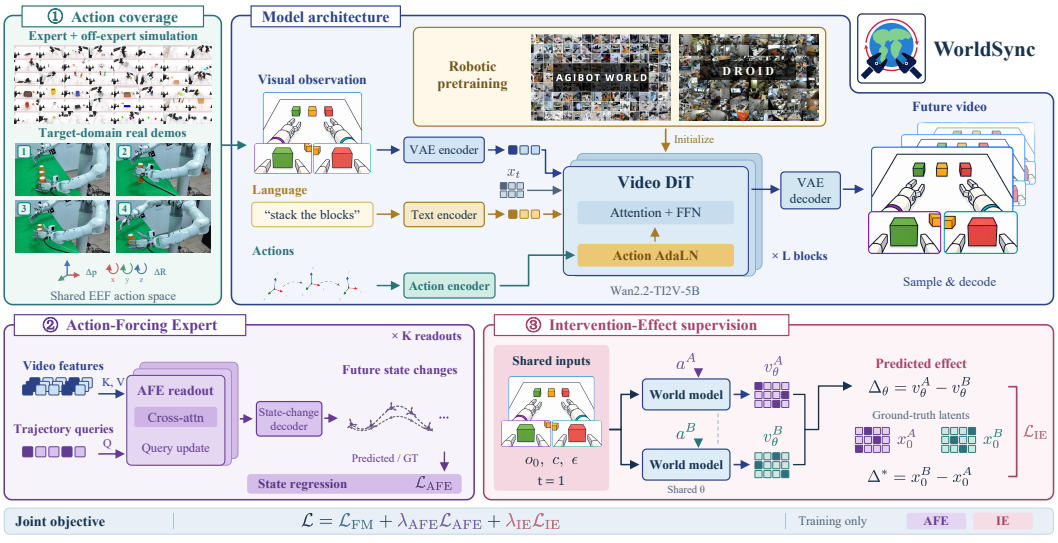

Figure 4: WorldSync overview: action coverage, the pretrained video backbone, and training-only supervision from the Action-Forcing Expert (AFE) and Intervention-Effect (IE) objectives.

We condition the video blocks of Wan2.2-TI2V-5B (Team Wan et al., 2025) on robot actions through Action AdaLN (Appendix D) and train the backbone by flow matching. For clean target-video latent x_0, Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, and flow time $t \in [0, 1]$, we construct $x_t = (1 - t)x_0 + t\epsilon$ and

optimize

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{x_0,\epsilon,t}\left[\|v_\theta(x_t, t \mid o_0, c, a_{1:H}) - (\epsilon - x_0)\|_2^2\right], \tag{7}$$

where v_θ is the world model's action-conditioned flow velocity.

Robotic Pretraining. For the additional robotic pretraining variant, we first optimize the backbone on approximately 1,011 hours of robot interaction data from AgiBot-World-Alpha (Team AgiBot-World, 2025) and DROID (Khazatsky et al., 2024) using $\mathcal{L}_{\mathrm{FM}}$. AFE and IE are introduced during subsequent WorldSync training. Dataset statistics and pretraining configurations are provided in Appendix B.

3.4.1 Action Coverage Expansion

We broaden expert-only training with multi-task simulated expert trajectories, local perturbations, cross-state replays, policy rollouts, and broad feasible actions. We mix in a small set of target-task real-robot demonstrations to preserve target-domain visual fidelity. Across simulated and real-robot data, actions use relative Cartesian end-effector pose displacements in the robot base frame and gripper opening. This shared representation allows the model to learn relationships between actions and their consequences across the two domains. Appendix E.2 examines the effect of training-data composition.

3.4.2 Action-Forcing Expert

Broader coverage alone does not ensure that intermediate video representations encode action-induced robot dynamics. To provide an auxiliary feature-level grounding signal, AFE maintains one learnable trajectory query per future step. These queries progressively cross-attend to the intermediate features of successive video blocks and predict future end-effector state changes relative to the initial state. The targets include both arms' position, rotation, and gripper state. Given standardized cumulative-change targets $y_{1:H}$ and predictions $\hat{y}_{1:H}$, we optimize

$$\mathcal{L}_{\mathrm{AFE}} = \mathrm{MSE}_{\mathrm{valid}}(\hat{y}_{1:H}, y_{1:H}), \tag{8}$$

where the masked mean squared error uses valid state labels. Appendix E.1 specifies target preprocessing and design choices. AFE neither reads actions directly nor writes back to the video stream; its loss updates the backbone through video features, and the expert is removed at inference time.

3.4.3 Intervention-Effect Supervision

While AFE grounds individual rollouts, it does not directly supervise how the generated future should change when the conditioned action changes. IE provides this complementary relational signal through paired trajectories that share the current observation and instruction but execute different actions. To isolate the effect of changing actions, we evaluate both branches at the noise endpoint $t = 1$, where shared noise gives identical input latents $x_1^A = x_1^B = \epsilon$. With the observation and instruction also shared, only the action inputs differ. The predicted intervention effect is $\Delta_\theta = v_\theta^A - v_\theta^B$. Since the shared noise cancels between the target velocities, the corresponding ground-truth effect is $\Delta^* = x_0^B - x_0^A$. We align these effects over future video latents:

$$\mathcal{L}_{\mathrm{IE}} = \|\Delta_\theta - \Delta^*\|_2^2. \tag{9}$$

3.4.4 Training Objective

Combining the flow-matching generation loss with the two auxiliary objectives gives

$$\mathcal{L} = \mathcal{L}_{\mathrm{FM}} + \lambda_{\mathrm{AFE}}\mathcal{L}_{\mathrm{AFE}} + \lambda_{\mathrm{IE}}\mathcal{L}_{\mathrm{IE}}. \tag{10}$$

We apply $\mathcal{L}_{\mathrm{AFE}}$ when future state labels are available and $\mathcal{L}_{\mathrm{IE}}$ to intervention pairs.

4 Experiments

4.1 Experimental Setup

Benchmarks and evaluation sets. We evaluate action following on 50 RoboTwin tasks (Mu et al., 2025; Chen et al., 2026b) using 1,750 queries from WorldEcho's five categories, and component

contributions on ten tasks under the same protocol. Policy improvement is evaluated in RoboTwin and on four real-robot tasks across three platforms (Appendix A).

Baselines. We compare WorldSync with CtrlWorld (Guo et al., 2026b), Cosmos-Predict2.5 (NVIDIA et al., 2025), Cosmos3 (NVIDIA, 2026), DreamDojo (Gao et al., 2026), Motus (Bi et al., 2025), and LingBotVA (Li et al., 2026a). Each baseline is trained with Expert Demonstrations or Expanded Action Coverage and evaluated on the same task split and queries. All baselines and both WorldSync variants are evaluated after 40k fine-tuning updates.

Metrics. Integrity-gated error is the primary metric. Raw pose-aware NDTW and visual pass rate distinguish action mismatch from visual failure. Raw and gated errors are task-macro averages. Visual pass rate is the fraction of accepted rollouts. Table 1 specifies video-metric aggregation. Video metrics are evaluated on all 1,750 queries.

4.2 Action-Following Evaluation

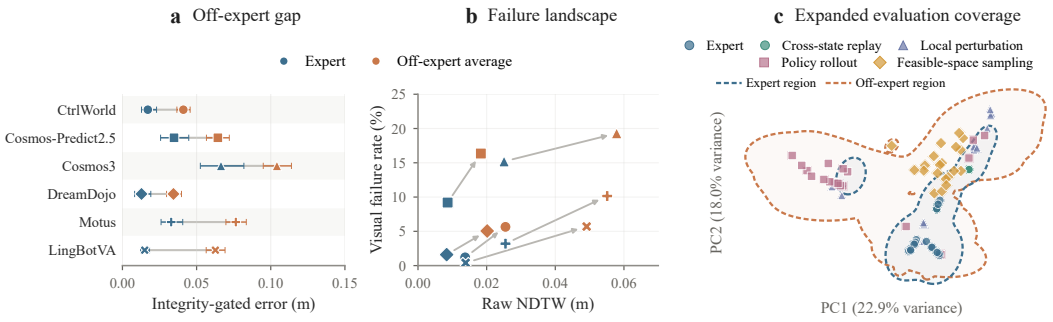

Figure 5: Off-expert action following. (a) Gated error with equally weighted off-expert categories within tasks. Error bars show 95% confidence intervals from task-level resampling. (b) Expert-to-off-expert changes in raw NDTW and visual failure rate. (c) Action PCA with 95% highest-density regions (off-expert: union across categories).

Expert-only evaluation understates off-expert failures. Across all six expert-trained models, off-expert queries increased gated error by 0.022–0.048 m (Figure 5a). This gap involved both trajectory inconsistency and visual degradation: raw NDTW increased by 0.010–0.035 m and visual failure rate by 3.45–7.15 percentage points across all models (Figure 5b). Their relative contributions varied by model, so either metric alone would miss part of the failure. The off-expert categories occupy a broader region than expert actions in the projected action space (Figure 5c). This expanded coverage exposes errors beyond expert replay, motivating both broader training support and stronger dependence on the queried actions.

Expanded-coverage models improve all three WorldEcho metrics. Compared with Expert Demonstrations, Expanded Action Coverage lowers gated error and raw NDTW and raises visual pass rates for all six baselines (Table 1), improving both trajectory alignment and visual integrity.

WorldSync achieves the lowest integrity-gated error. Without additional robotic pretraining, WorldSync achieves a lower gated-error point estimate than every baseline (0.0234 versus 0.0237 for the strongest baseline, expanded-coverage DreamDojo). Additional robotic pretraining further lowers this estimate to 0.0228. The pretrained and non-pretrained variants' visual pass rates were also highest (98.23% and 97.94%, versus 97.14% for expanded-coverage Motus). However, expanded-coverage Cosmos-Predict2.5 had lower raw NDTW (0.0127 versus 0.0180 and 0.0176), so the gated ranking does not imply the lowest trajectory error. Both WorldSync variants ranked among the top three on PSNR and FVD. Relative to the non-pretrained variant, pretraining improved visual pass rate but slightly increased raw NDTW. Appendix G presents qualitative comparisons and per-case metrics.

Table 1: Main WorldEcho comparison. Expert and Expanded denote Expert Demonstrations and Expanded Action Coverage. Bold marks the best scores. Underlining marks second-best WorldEcho scores. Emb. PT marks embodied-data pretraining before task-specific fine-tuning, excluding generic-only video pretraining (Appendix B). † Video metrics use reconstructed references, not verified as pixel-identical to the original RGB videos. Baseline video and WorldEcho scores use separate prediction batches at the same checkpoint. PSNR, SSIM, and LPIPS average over frames, clips, then tasks. FVD pools all clips.

			WorldEcho metrics			Video generation metrics †			
Model	**Emb. PT**	**Training regime**	**Gated error ↓**	**Raw NDTW ↓**	**Visual pass (%) ↑**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**FVD ↓**
CtrlWorld	✓	Expert	0.0379	0.0266	95.83	19.41	0.8466	0.1137	113.58
CtrlWorld	✓	Expanded	0.0293	0.0210	96.97	20.83	0.8665	0.0915	69.58
Cosmos-Predict2.5	✓	Expert	0.0551	0.0190	87.09	21.10	0.8750	0.0848	95.69
Cosmos-Predict2.5	✓	Expanded	0.0445	**0.0127**	88.80	**22.95**	0.8952	**0.0658**	51.89
Cosmos3	✓	Expert	0.0989	0.0572	82.51	14.45	0.6846	0.1833	122.58
Cosmos3	✓	Expanded	0.0799	0.0419	85.09	15.10	0.7003	0.1525	86.36
DreamDojo	✓	Expert	0.0313	0.0210	96.29	20.49	0.8909	0.1049	81.83
DreamDojo	✓	Expanded	0.0237	<u>0.0151</u>	96.97	21.17	**0.8959**	0.0949	**30.77**
Motus	✓	Expert	0.0708	0.0548	91.94	17.21	0.8164	0.1620	106.76
Motus	✓	Expanded	0.0371	0.0292	97.14	19.22	0.8448	0.1182	68.78
LingBotVA	✓	Expert	0.0572	0.0473	95.60	16.54	0.8084	0.1717	128.21
LingBotVA	✓	Expanded	0.0446	0.0340	96.06	17.60	0.8272	0.1450	81.16
WorldSync	–	Expanded	<u>0.0234</u>	0.0176	<u>97.94</u>	21.42	0.8743	0.0875	46.97
WorldSync w/ pretrain	✓	Expanded	**0.0228**	0.0180	**98.23**	21.36	0.8738	0.0887	45.61

4.3 Policy Improvement

We adapt the VLAW framework (Guo et al., 2026a) to compare WorldSync with expert-trained CtrlWorld under a shared policy-improvement procedure. Both conditions use the same initial policy and matched interaction, world-model rollout, and policy-training budgets within each task. Appendix C details the simulation procedure and condition-specific world-model adaptation.

Simulation. On Bin Dumping, both conditions started at 52% success. WorldSync reached 65% versus 57% for CtrlWorld after two rounds (Figure 6a). From 3,000 imagined rollouts per round, the same task-progress and action-consistency filters retained 553 versus 452 rollouts in Round 1 and 365 versus 323 in Round 2 (Appendix C.3). Thus, WorldSync provided 22.3% and 13.0% more usable imagined rollouts under the same generation budget, alongside larger policy gains.

Real-robot tasks. The four tasks span three manipulation platforms (Figure 6b–e and Appendix A.2). Cup Stacking uses two TianJi arms with Sharpa dexterous hands to nest three cups through sequential left- and right-hand placements. Box Packing uses the dual-arm Simplexity i7 Pro with Xense grippers to place an object in a box, close its flaps, and tape the top seam. On the single-arm Franka Research 3, Bread Placement transfers two slices from a toaster to a cutting board, while Pen Collection places two pens into a holder. These tasks cover coordinated dual-arm manipulation, dexterous-hand control, and sequential single-arm pick-and-place.

Real-robot data. The demonstration datasets contain 200 trajectories for Cup Stacking, 200 for Box Packing, 300 for Bread Placement, and 500 for Pen Collection (Appendix A.1). All three platforms provide workspace and wrist camera observations, with training trajectories sampled at 10 Hz. The TianJi platform controls 22 independently actuated degrees of freedom per hand, while the i7 Pro and Franka platforms use gripper-opening commands. Success requires completing the full task sequence, including closing and sealing the box or placing both bread slices or pens at their targets.

Real-robot results. Each success rate uses 25 trials. On Cup Stacking, both conditions started at 48%. WorldSync reached 68% versus 56% after two rounds. The other tasks use one round. Box Packing improved from 40% to 60% with WorldSync while CtrlWorld remained at 40%. Bread Placement improved from 76% to 96% versus 84%. Pen Collection improved from 72% to 88%

versus 64%. WorldSync therefore improved the initial policy on all four tasks, with final success rates 12–24 percentage points above CtrlWorld. The gains span both dexterous hands and grippers, including Box Packing and Pen Collection, where CtrlWorld yielded no improvement or a decline. Together with the simulation results, these experiments connect stronger WorldEcho performance with larger downstream policy gains.

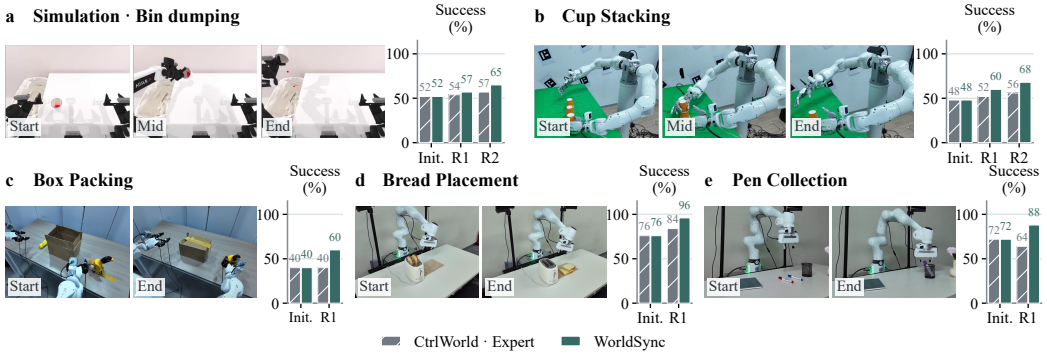

Figure 6: Policy improvement in simulation and four real-robot tasks. Each real-robot success rate uses 25 trials. Init. denotes the initial policy, and R1/R2 denote improvement rounds.

4.4 Component Contributions and Interactions

Table 2: Data and component ablation on ten RoboTwin tasks at 40k training steps. Bold denotes the best reported value, including ties. Video metrics use separate generations at the same checkpoint. PSNR, SSIM, and LPIPS average over frames, clips, then tasks. FVD pools all video pairs.

Variant	Gated ↓	Raw ↓	Pass (%) ↑	PSNR ↑	SSIM ↑	LPIPS ↓	FVD ↓
Expert Base	0.0352	0.0314	**98.57**	21.23	0.8585	0.1154	175.09
Base	0.0264	0.0204	97.86	20.66	0.8591	0.1087	141.10
+ IE	0.0214	0.0176	**98.57**	21.05	0.8665	0.0963	125.92
+ AFE	0.0232	0.0172	97.86	21.30	0.8683	0.0962	118.81
IE + AFE	**0.0193**	**0.0138**	97.86	**22.33**	**0.8821**	**0.0830**	**105.83**

In Table 2, both Base variants disable IE and AFE. Expert Base uses expert-only data. All other rows use Expanded data. Expanded Base lowers gated and raw errors by 25.0% and 35.0%, respectively, despite lower visual pass rate and PSNR. IE and AFE each further reduce both errors. Their combination achieves the lowest errors and best video metrics, reducing gated and raw errors by 9.57% and 21.68% relative to IE, with a 0.71-percentage-point drop in visual pass. Thus, both objectives contribute at this checkpoint, while visual pass does not improve uniformly (Appendix E.1).

5 Conclusion and Limitations

In this work, we introduced WorldEcho to evaluate action-conditioned world models beyond the narrow distribution of expert demonstrations, jointly measuring visual integrity and action-induced trajectory alignment. Our evaluation shows that the six expert-trained models degrade under feasible off-expert actions, exposing failures overlooked by expert-only protocols. Guided by this diagnosis, we proposed WorldSync, which combines distributional coverage, representational grounding, and intervention-effect alignment for faithful action-conditioned generation. WorldSync reduced integrity-gated error on RoboTwin and yielded larger policy gains than expert-trained CtrlWorld in simulation and on real robots. Although WorldEcho substantially broadens evaluation coverage, comprehensively probing long-horizon interactions across diverse embodiments and open-world environments remains a shared challenge for the field and an important direction for future work.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript formatting and language editing. All AI-assisted content was reviewed by the authors, who take responsibility for the final content of this work.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model, 2025. URL https://arxiv.org/abs/2512.13030.

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

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=r35clVtGzw.

Longyu Chen, Heng Li, Wei Yang, Manqi Zhao, and Dongsheng Jiang. ViPSim: Collaborating visual and parameter spaces for consistent long-horizon embodied world model. In *Robotics: Science and Systems*, 2026a. URL https://roboticsconference.org/program/papers/14/.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *Forty-third International Conference on Machine Learning*, 2026b. URL https://icml.cc/virtual/2026/poster/62192.

Yixiang Chen, Peiyan Li, Jiabing Yang, Keji He, Xiangnan Wu, Yuan Xu, Kai Wang, Jing Liu, Nianfeng Liu, Yan Huang, and Liang Wang. BridgeV2W: Bridging video generation models to

embodied world models via embodiment masks, 2026c. URL https://arxiv.org/abs/2602.03793.

Yuzhi Chen, Ronghan Chen, Dongjie Huo, Yandan Yang, Dekang Qi, Haoyun Liu, Tong Lin, Shuang Zeng, Junjin Xiao, Xinyuan Chang, Feng Xiong, Xing Wei, Zhiheng Ma, and Mu Xu. ABot-PhysWorld: Interactive world foundation model for robotic manipulation with physics alignment, 2026d. URL https://arxiv.org/abs/2603.23376.

Chun-Kai Fan, Xiaowei Chi, Xiaozhu Ju, Hao Li, Yong Bao, Yu-Kai Wang, Lizhang Chen, Zhiyuan Jiang, Kuangzhi Ge, Ying Li, Weishi Mi, Qingpo Wuwu, Peidong Jia, Yulin Luo, Kevin Zhang, Zhiyuan Qin, Yong Dai, Sirui Han, Yike Guo, Shanghang Zhang, and Jian Tang. Wow, wo, val!: A comprehensive embodied world model evaluation turing test, 2026. URL https://arxiv.org/abs/2601.04137.

Riccardo O. Feingold, Davide Liconti, Chenyu Yang, and Robert K. Katzschmann. Mask2Real-WM: Segmentation masks as a sim-to-real bridge for controllable dexterous world models, 2026. URL https://arxiv.org/abs/2607.04546.

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

Yanjiang Guo, Tony Lee, Lucy Xiaoyang Shi, Jianyu Chen, Percy Liang, and Chelsea Finn. VLAW: Iterative co-improvement of vision-language-action policy and world model. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=68zOaa2gOf.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A controllable generative world model for robot manipulation. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=748bHL2BAv.

Yue Hu, Siyuan Huang, Yue Liao, Shengcong Chen, Pengfei Zhou, Liliang Chen, Guanghui Ren, and Maoqing Yao. EWMBench: Evaluating scene, motion, and semantic quality in embodied world models. In *36th British Machine Vision Conference*. BMVA, 2025. URL https://bmva-archive.org.uk/bmvc/2025/assets/papers/Paper_736/paper.pdf.

Siqiao Huang, Partha Kaushik, Michael Chen, Hengkai Pan, Kaiwen Geng, Omar Chehab, Fernando Moreno-Pino, and Max Simchowitz. Nano World Models: A minimalist implementation of future video prediction, 2026a. URL https://arxiv.org/abs/2605.23993.

Ze Huang, Jiahui Zhang, Hairuo Liu, Chenxi Zhang, Ran Cheng, and Li Zhang. Learning transferable dynamics priors from action to world modeling, 2026b. URL https://arxiv.org/abs/2606.29501. Accepted to ECCV 2026; proceedings version not yet available as of 2026-07-15.

Byeongguk Jeon, Seonghyeon Ye, JaeHyeok Doo, Sungdong Kim, Minjoon Seo, Hyungmok Son, and Kimin Lee. RoboWorld: Fast and reliable neural simulators for generalist robot policy evaluation, 2026. URL https://arxiv.org/abs/2607.01060.

Feng Jiang, Yang Chen, Kyle Xu, Yuchen Liu, Haifeng Wang, Zhenhao Shen, Jasper Lu, Shengze Huang, Yuanfei Wang, Chen Xie, and Ruihai Wu. RoboWM-Bench: A benchmark for evaluating world models in robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 4455–4460, June 2026a. URL https://openaccess.thecvf.com/content/CVPR2026W/GigaBrainChallenge/html/Jiang_RoboWM-Bench_A_Benchmark_for_Evaluating_World_Models_in_Robotic_Manipulation_CVPRW_2026_paper.html.

Zhennan Jiang, Shangqing Zhou, Yutong Jiang, Zefang Huang, Mingjie Wei, Yuhui Chen, Tianxing Zhou, Zhen Guo, Hao Lin, Quanlu Zhang, Yu Wang, Haoran Li, Chao Yu, and Dongbin Zhao. WoVR: World models as reliable simulators for post-training vla policies with rl. *arXiv preprint arXiv:2602.13977*, 2026b. doi: 10.48550/arXiv.2602.13977. URL https://arxiv.org/abs/2602.13977.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. MUSIQ: Multi-scale image quality transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5128–5137, Montreal, QC, Canada, October 2021. IEEE. doi: 10.1109/ICCV48922.2021.00510. URL https://doi.org/10.1109/ICCV48922.2021.00510.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.120.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos Policy: Fine-tuning video models for visuomotor control and planning. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 71531–71552, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/748becc400a57c0e31cfe6a2e7951467-Paper-Conference.pdf.

Hengtao Li, Pengxiang Ding, Runze Suo, Yihao Wang, Zirui Ge, Dongyuan Zang, Kexian Yu, Mingyang Sun, Hongyin Zhang, Donglin Wang, and Weihua Su. VLA-RFT: Vision-language-action reinforcement fine-tuning with verified rewards in world simulators, 2025a. URL https://arxiv.org/abs/2510.00406.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. In *Proceedings of Robotics: Science and Systems*, 2026a. URL https://www.roboticsproceedings.org/rss22/p016.pdf.

Yaxuan Li, Yichen Zhu, Junjie Wen, Chaomin Shen, and Yi Xu. WorldEval: World model as real-world robot policies evaluator. *arXiv preprint arXiv:2505.19017*, 2025b. doi: 10.48550/arXiv.2505.19017. URL https://arxiv.org/abs/2505.19017.

Yaxuan Li, Zhongyi Zhou, Yefei Chen, Yanjiang Guo, Jiaming Liu, Shanghang Zhang, Jianyu Chen, and Yichen Zhu. Hi-WM: Human-in-the-world-model for scalable robot post-training, 2026b. URL https://arxiv.org/abs/2604.21741.

Xiaokang Liu, Zechen Bai, Hai Ci, Kevin Yuchen Ma, and Mike Zheng Shou. World-VLA-Loop: Closed-loop learning of video world model and vla policy, 2026. URL https://arxiv.org/abs/2602.06508.

Chong Ma, Taiyi Su, Jian Zhu, Jianjun Zhang, Zitai Huang, Yi Xu, and Hanli Wang. PiL-World: A chunk-wise world model for vla policy-in-the-loop evaluation, 2026. URL https://arxiv.org/abs/2606.05773.

Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, Lunkai Lin, Zhiqiang Xie, Mingyu Ding, and Ping Luo. RoboTwin: Dual-arm robot benchmark with generative digital twins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27649–27660. IEEE, 2025. doi: 10.1109/CVPR52734.2025.02575. URL https://doi.org/10.1109/CVPR52734.2025.02575.

NVIDIA. Cosmos 3: Omnimodal world models for physical AI, 2026. URL https://arxiv.org/abs/2606.02800.

NVIDIA, Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, Prithvijit Chattopadhyay, Mike Chen, Yongxin Chen, Yu Chen, Shuai Cheng, Yin Cui, Jenna Diamond, Yifan Ding, Jiaojiao Fan, Linxi Fan, Liang Feng, Francesco Ferroni, Sanja Fidler, Xiao Fu, Ruiyuan Gao, Yunhao Ge, Jinwei Gu, Aryaman Gupta, Siddharth Gururani, Imad El Hanafi, Ali Hassani, Zekun Hao, Jacob Huffman, Joel Jang, Pooya Jannaty, Jan Kautz, Grace Lam, Xuan Li, Zhaoshuo Li, Maosheng Liao, Chen-Hsuan Lin, Tsung-Yi Lin, Yen-Chen Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Yifan Lu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Seungjun Nah, Yashraj Narang, Abhijeet Panaskar, Lindsey Pavao, Trung Pham, Morteza Ramezanali, Fitsum Reda, Scott Reed, Xuanchi Ren, Haonan Shao, Yue Shen, Stella Shi, Shuran Song, Bartosz Stefaniak, Shangkun Sun, Shitao Tang, Sameena Tasmeen, Lyne Tchapmi, Wei-Cheng Tseng, Jibin Varghese, Andrew Z. Wang, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Jiashu Xu, Dinghao Yang, Xiaodong Yang, Haotian Ye, Seonghyeon Ye, Xiaohui Zeng, Jing Zhang, Qinsheng Zhang, Kaiwen Zheng, Andrew Zhu, and Yuke Zhu. World simulation with video foundation models for physical AI, 2025. URL https://arxiv.org/abs/2511.00062.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi Jim Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick Tree Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shuran Song, Sichun

Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10611477. URL https://doi.org/10.1109/ICRA57147.2024.10611477.

Mingjie Pan, Siyuan Feng, Qinglin Zhang, Xinchen Li, Jianheng Song, Chendi Qu, Yi Wang, Chuankang Li, Ziyu Xiong, Zhi Chen, Yi Liu, and Jianlan Luo. SOP: A scalable online post-training system for vision-language-action models, 2026. URL https://arxiv.org/abs/2601.03044.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachy Groom, Hunter Hancock, Karol Hausman, Gashon Hussein, Brian Ichter, Szymon Jakubczak, Rowan Jen, Tim Jones, Ben Katz, Liyiming Ke, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Yao Lu, Vishnu Mano, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Charvi Sharma, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Alex Swerdlow, James Tanner, Marcel Torne, Quan Vuong, Anna Walling, Haohuan Wang, Blake Williams, Sukwon Yoo, Lili Yu, Ury Zhilinsky, and Zhiyuan Zhou. $\pi^*_{0.6}$: a VLA that learns from experience, 2025. URL https://arxiv.org/abs/2511.14759.

Boxiang Qiu, Liliang Chen, Yue Liao, Nan Wang, Lintao Wang, Jiayi Luo, Wenzhi Zhao, Shengcong Chen, Di Chen, Ye Li, Chen Gao, Shuicheng Yan, Si Liu, Maoqing Yao, and Guanghui Ren. GE-Sim 2.0: A roadmap towards comprehensive closed-loop video world simulators for robotic manipulation, 2026. URL https://arxiv.org/abs/2605.27491.

Julian Hector Quevedo, Ansh Kumar Sharma, Yixiang Sun, Varad Suryavanshi, Percy Liang, and Sherry Yang. WorldGym: World model as an environment for policy evaluation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=hidBHy1CAw.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, February 1978. doi: 10.1109/TASSP.1978.1163055. URL https://doi.org/10.1109/TASSP.1978.1163055.

Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007. doi: 10.3233/IDA-2007-11508. URL https://doi.org/10.3233/IDA-2007-11508.

Yu Shang, Zhuohang Li, Yiding Ma, Weikang Su, Xin Jin, Ziyou Wang, Lei Jin, Xin Zhang, Yinzhou Tang, Haisheng Su, Chen Gao, Wei Wu, Xihui Liu, Dhruv Shah, Zhaoxiang Zhang, Zhibo Chen, Jun Zhu, Yonghong Tian, Tat-Seng Chua, Wenwu Zhu, and Yong Li. WorldArena: A unified benchmark for evaluating perception and functional utility of embodied world models. *arXiv*

preprint arXiv:2602.08971, 2026a. doi: 10.48550/arXiv.2602.08971. URL https://arxiv.org/abs/2602.08971.

Yu Shang, Yinzhou Tang, Yiding Ma, Zhuohang Li, Lei Jin, Weikang Su, Xin Jin, Zhaolu Wang, Ziyou Wang, Xin Zhang, Haisheng Su, Weizhen He, Wei Wu, Haoyi Duan, Gordon Wetzstein, Xihui Liu, Dhruv Shah, Zhaoxiang Zhang, Zhibo Chen, Jun Zhu, Yonghong Tian, Tat-Seng Chua, Wenwu Zhu, Chen Gao, and Yong Li. WorldArena 2.0: Extending embodied world model benchmarking on modality, functionality and platform, 2026b. URL https://arxiv.org/abs/2605.17912.

Hengkai Tan, Yao Feng, Xinyi Mao, Shuhe Huang, Guodong Liu, Zhongkai Hao, Hang Su, and Jun Zhu. AnyPos: Automated task-agnostic actions for bimanual manipulation, 2026a. URL https://arxiv.org/abs/2507.12768.

Huajie Tan, Sixiang Chen, Yijie Xu, Zixiao Wang, Cheng Chi, Yuheng Ji, Yaoxu Lyu, Zhongxia Zhao, Xiansheng Chen, Peterson Co, Shaoxuan Xie, Guocai Yao, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. General process reward modeling for robotic reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22412–22422, June 2026b. URL https://openaccess.thecvf.com/content/CVPR2026/html/Tan_General_Process_Reward_Modeling_for_Robotic_Reinforcement_Learning_CVPR_2026_paper.html.

Team AgiBot-World. AgiBot World Colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3549–3556. IEEE, 2025. doi: 10.1109/IROS60139.2025.11247088. URL https://doi.org/10.1109/IROS60139.2025.11247088.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models, 2025. URL https://arxiv.org/abs/2503.20314.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NxoFmGgWC9.

Zhuoyuan Wu and Jun Gao. OSCAR: Omni-embodiment action-conditioned world model for robotics, 2026. URL https://arxiv.org/abs/2606.04463.

Junjin Xiao, Yandan Yang, Xinyuan Chang, Ronghan Chen, Feng Xiong, Mu Xu, Wei-Shi Zheng, and Qing Zhang. RehearseVLA: Simulated post-training for vlas with physically-consistent world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20867–20877, June 2026.

Tianzhuo Yang, Zihan Shen, Zirui Mi, Zhaoyi Zhang, Jiayi Zhou, Jiaming Ji, Juntao Dai, Jiawei Chen, Boyuan Chen, and Yaodong Yang. MiraBench: Evaluating action-conditioned reliability in robotic world models. *arXiv preprint arXiv:2605.29360*, 2026a. doi: 10.48550/arXiv.2605.29360. URL https://arxiv.org/abs/2605.29360.

Zhaoyang Yang, Yurun Jin, Lizhe Qi, Cong Huang, and Kai Chen. EA-WM: Event-aware generative world model with structured kinematic-to-visual action fields, 2026b. URL https://arxiv.org/abs/2605.06192.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William

Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World action models are zero-shot policies, 2026. URL https://arxiv.org/abs/2602.15922.

Tenny Yin, Zhiting Mei, Zhonghe Zheng, Miyu Yamane, David Wang, Jade Sceats, Samuel M. Bateman, Lihan Zha, Apurva Badithela, Ola Shorinwa, and Anirudha Majumdar. PlayWorld: Learning robot world models from autonomous play, 2026. URL https://arxiv.org/abs/2603.09030.

Anlan Yu, Zaishu Chen, Peili Song, Zhiqing Hong, Haotian Wang, Desheng Zhang, Tian He, Yi Ding, and Daqing Zhang. WM-DAgger: Enabling efficient data aggregation for imitation learning with world models. *arXiv preprint arXiv:2604.11351*, 2026. doi: 10.48550/arXiv.2604.11351. URL https://arxiv.org/abs/2604.11351.

Guozhen Zhang, Chunxu Liu, Yutao Cui, Xiaotong Zhao, Kai Ma, and Limin Wang. VFIMamba: Video frame interpolation with state space models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 107225–107248. Curran Associates, Inc., 2024. doi: 10.52202/079017-3405. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/c1e9db5e1b04322963af91ac0c943568-Abstract-Conference.html.

Haoyu Zhen, Zixian Gao, Qiao Sun, Yilin Zhao, Yuncong Yang, Yilun Du, Pengsheng Guo, Tsun-Hsuan Wang, Yi-Ling Qiao, and Chuang Gan. Action Images: End-to-end policy learning via multiview video generation, 2026. URL https://arxiv.org/abs/2604.06168.

Zirui Zheng, Jiaqian Yu, Xiongfeng Peng, Jun Shi, Mingyi Li, Chao Zhang, Weiming Li, Dong Wang, Huchuan Lu, and Xu Jia. Mem-World: Memory-augmented action-conditioned world models for persistent robot manipulation, 2026. URL https://arxiv.org/abs/2606.18960.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. IRASim: A fine-grained world model for robot manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9834–9844, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Zhu_IRASim_A_Fine-Grained_World_Model_for_Robot_Manipulation_ICCV_2025_paper.html.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

APPENDIX

A REAL-ROBOT CONFIGURATION

This appendix describes the real-robot platforms and manipulation tasks. Section A.1 details the hardware, action spaces, data collection, and camera views for each platform. Section A.2 introduces the task instructions and manipulation sequences, illustrated through third-person and robot-camera views.

A.1 ROBOT PLATFORMS

A.1.1 DUAL TIANJI-SHARPA (ROBOT 1)

Hardware. Our first platform uses two TianJi robotic arms, each with seven degrees of freedom (DoF), equipped with Sharpa Wave dexterous hands. Each hand has 22 independently actuated DoF. A ZED 2i head camera provides a workspace view, while Insta360 GO 3S cameras provide wrist views. VIVE Ultimate Trackers control the arm end-effector motion during teleoperation, and MANUS Metagloves Pro Haptic provide the operator's hand motion for controlling the dexterous hands. Figure 7 identifies the platform hardware and teleoperation devices.

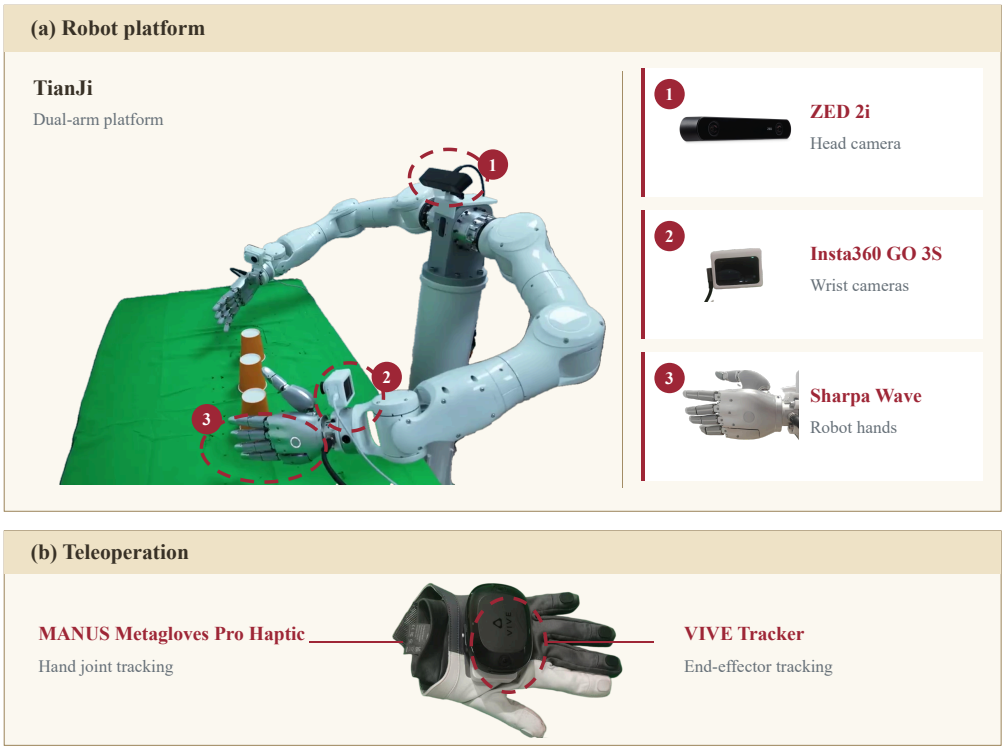

Figure 7: TianJi hardware and teleoperation setup. Numbered markers match the installed devices to the equipment list; the lower panel identifies the hand-motion and end-effector tracking devices.

Action Space. We retain WorldSync's relative Cartesian end-effector pose commands (Delta EE) for each arm and extend its hand control to dexterous manipulation. For each hand, the single gripper-opening component is replaced by 22 absolute joint-position targets. The combined command is

$$a_t = \left(\Delta T_t^L, q_t^L, \Delta T_t^R, q_t^R\right), \qquad \Delta T_t^e \in \mathrm{SE}(3), \quad q_t^e \in \mathbb{R}^{22}, \quad e \in \{L, R\}. \tag{11}$$

Here, ΔT_t^e specifies the commanded end-effector pose change and q_t^e specifies the corresponding hand-joint targets.

Data Collection. We collect 200 teleoperated demonstrations of Cup Stacking for the initial baseline dataset, before iterative policy improvement. Observations and robot commands are recorded at 30 Hz and downsampled to 10 Hz for training. The demonstrations pair camera observations with end-effector motion commands and dexterous-hand joint targets.

Camera Views. The head view shows the tabletop and both hands, while the wrist views show the nearby hand and objects during manipulation. The RGB frames have a resolution of 640×360 pixels and retain the full stored field of view.

A.1.2 SIMPLEXITY I7 PRO (ROBOT 2)

Hardware. Our second platform is a Simplexity i7 Pro dual-arm robot equipped with Xense grippers. The base and torso remain fixed during the task. An Orbbec Gemini 335Lg head camera provides a workspace view, and Orbbec Gemini 305g cameras provide left and right wrist views. Figure 8 identifies the installed devices and the PICO 4 Ultra teleoperation interface.

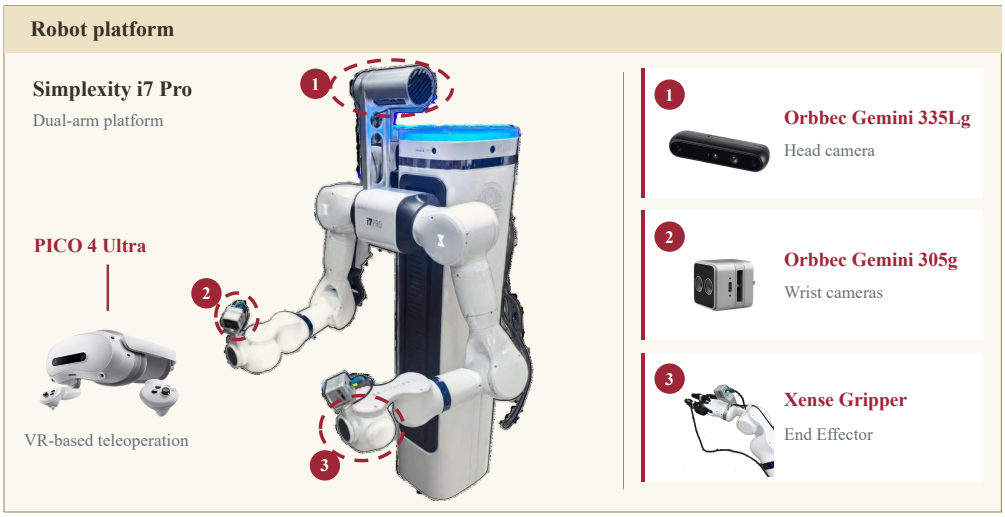

Figure 8: i7 Pro hardware and teleoperation setup. Numbered markers link the installed cameras and grippers to the equipment list; the left side shows the PICO teleoperation device.

Action Space. Following WorldSync's shared end-effector action formulation, we control both arms through relative Cartesian end-effector pose commands (Delta EE) and a continuous gripper-opening command for each arm. The combined command is

$$a_t = \left(\Delta T_t^L, g_t^L, \Delta T_t^R, g_t^R\right), \qquad \Delta T_t^e \in \mathrm{SE}(3), \quad g_t^e \in \mathbb{R}, \quad e \in \{L, R\}. \tag{12}$$

Here, ΔT_t^e specifies the commanded end-effector pose change and g_t^e specifies the corresponding gripper opening.

Data Collection. We collect 200 demonstrations of Box Packing through PICO teleoperation. Camera observations and robot commands are recorded at 30 Hz and downsampled to 10 Hz for training.

Camera Views. The head camera observes the tabletop and both arms, while the wrist cameras show the grippers and nearby objects during manipulation. The three camera views are synchronized.

A.1.3 FRANKA RESEARCH 3 (ROBOT 3)

Hardware. Our third platform uses a Franka Research 3 (FR3) robotic arm with seven degrees of freedom (DoF), equipped with a parallel-jaw gripper. Two Intel RealSense D435i cameras provide

complementary external views of the workspace. An Intel RealSense D405 mounted near the end effector provides a close-range wrist view. Figure 9 identifies the platform and installed cameras.

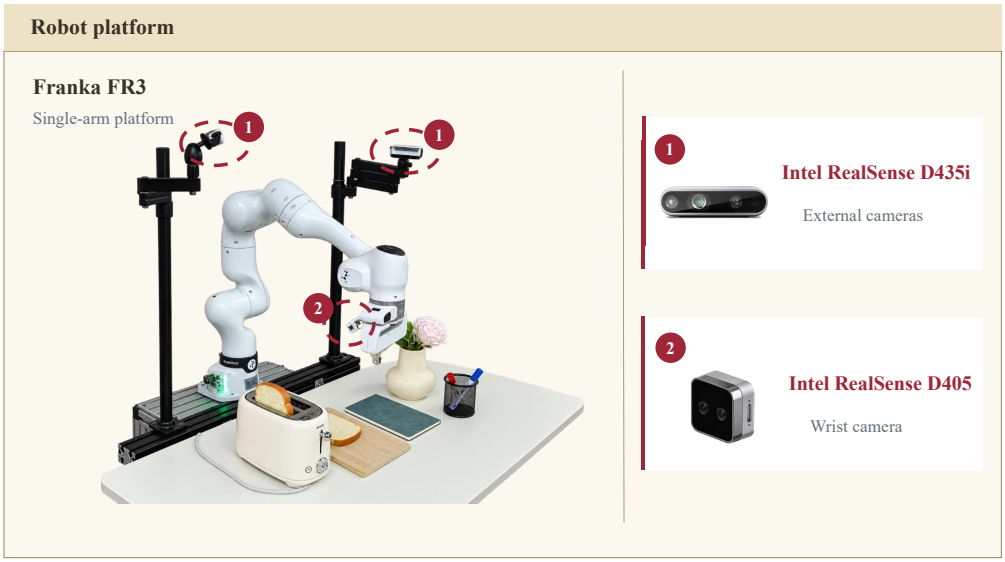

Figure 9: Franka FR3 hardware and camera setup. Numbered markers link the camera locations to the equipment list.

Action Space. We control the FR3 through relative Cartesian end-effector pose commands (Delta EE) and a continuous gripper-opening command, following WorldSync's shared end-effector action formulation. The single-arm command is

$$a_t = \big(\Delta T_t, g_t\big), \qquad \Delta T_t \in \mathrm{SE}(3), \quad g_t \in \mathbb{R}. \tag{13}$$

Here, ΔT_t specifies the commanded end-effector pose change and g_t specifies gripper opening. Both FR3 tasks use this action representation.

Data Collection. We collect 300 demonstrations for Bread Placement and 500 demonstrations for Pen Collection. The recorded trajectories are downsampled to 10 Hz for training. Each demonstration pairs the three camera observations with the corresponding relative end-effector pose commands and continuous gripper-opening commands.

Camera Views. The two external cameras observe the workspace from complementary viewpoints, while the wrist-mounted camera shows the gripper and nearby objects during manipulation.

A.2 TASKS

A.2.1 CUP STACKING

Instruction. "Stack the three upside-down cups into a single stack at the center of the table."

Task Description. Three paper cups are initially placed upside down on the table in front of the TianJi platform. The robot first uses its left hand to place the left cup over the center cup. It then uses its right hand to place the right cup over the resulting stack. The task ends with all three cups nested in a single stack at the center position.

Task Visualization. Figures 10 and 11 illustrate Cup Stacking from third-person and robot-camera views, respectively.

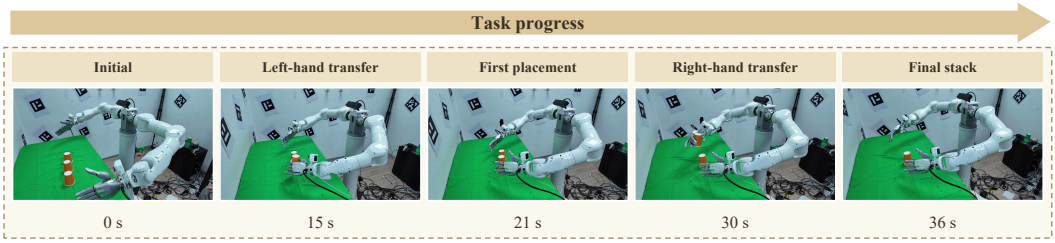

Figure 10: Cup Stacking from a third-person viewpoint. Read left to right; timestamps mark selected frames from one rollout.

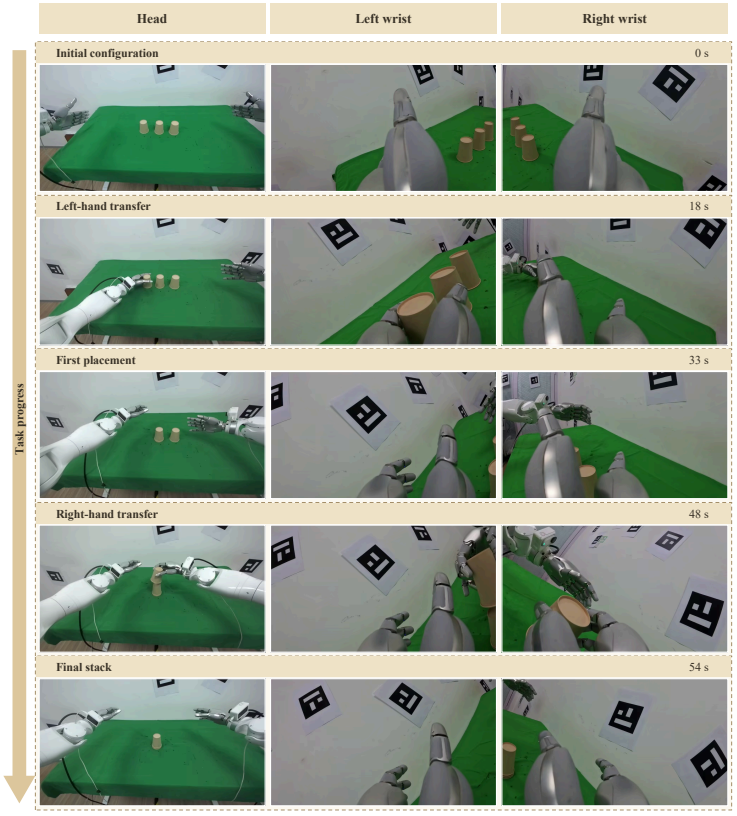

Figure 11: Cup Stacking through the robot's cameras. Columns show the head, left wrist, and right wrist; rows advance in time, with synchronized views within each row. This rollout is separate from Figure 10; timestamps are relative to each recording.

A.2.2 BOX PACKING

Instruction. "Place the mango in the box, close the flaps, and seal the box with tape."

Task Description. A mango model, an open cardboard box, and a tape dispenser are initially placed on the table in front of the i7 Pro platform. The robot places the mango in the box, folds the top flaps closed, and uses the tape dispenser to apply tape across the top seam. The task ends with the mango inside the closed box and tape applied across the seam.

Task Visualization. Figures 12 and 13 illustrate Box Packing from third-person and robot-camera views, respectively.

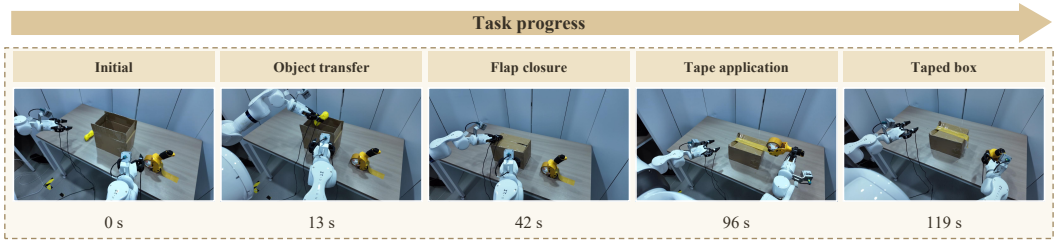

Figure 12: Box Packing from a third-person viewpoint. Read left to right; timestamps mark selected frames from one rollout.

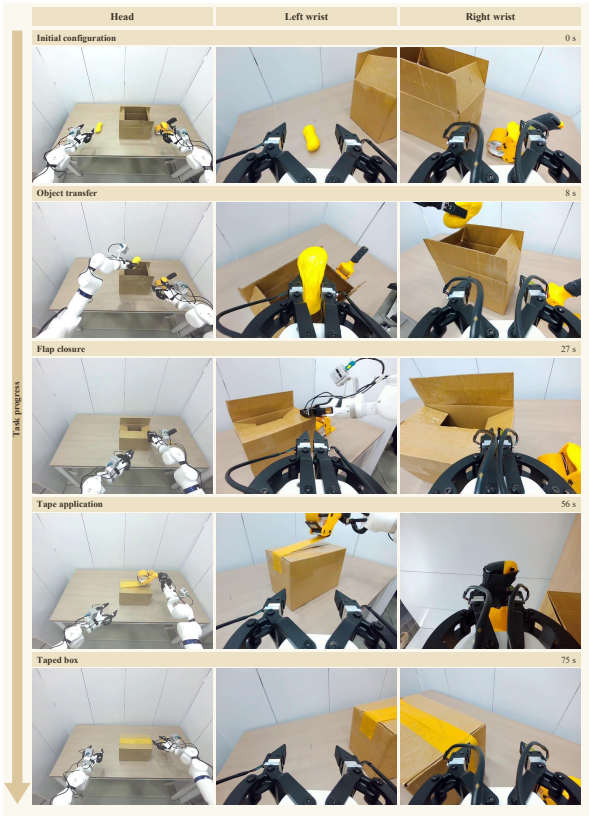

Figure 13: Box Packing through the robot's cameras. Columns show the head, left wrist, and right wrist; rows advance in time, with synchronized views within each row. This rollout is separate from Figure 12; timestamps are relative to each recording.

A.2.3 BREAD PLACEMENT

Instruction. The recorded task instruction is "Take the bread out of the toaster and place it on the cutting board."

Task Description. Two slices of bread are initially placed in the toaster in front of the FR3 platform. The robot sequentially grasps each slice, removes it from the toaster, transfers it to the cutting board, and releases it. The task ends when both slices have been successfully placed on the cutting board.

Task Visualization. Figures 14 and 15 illustrate Bread Placement from third-person and robot-camera views, respectively.

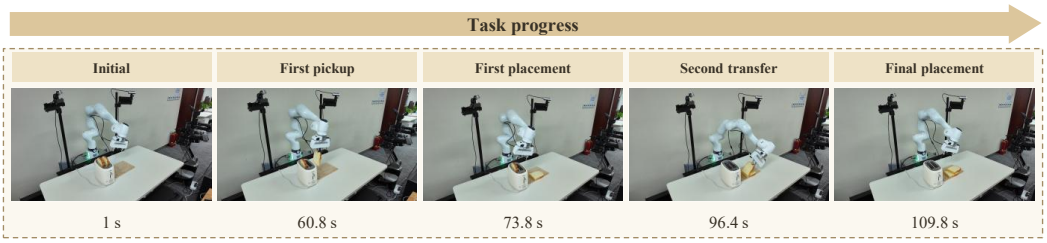

Figure 14: Bread Placement from a third-person viewpoint. Read left to right; timestamps mark selected frames from one rollout.

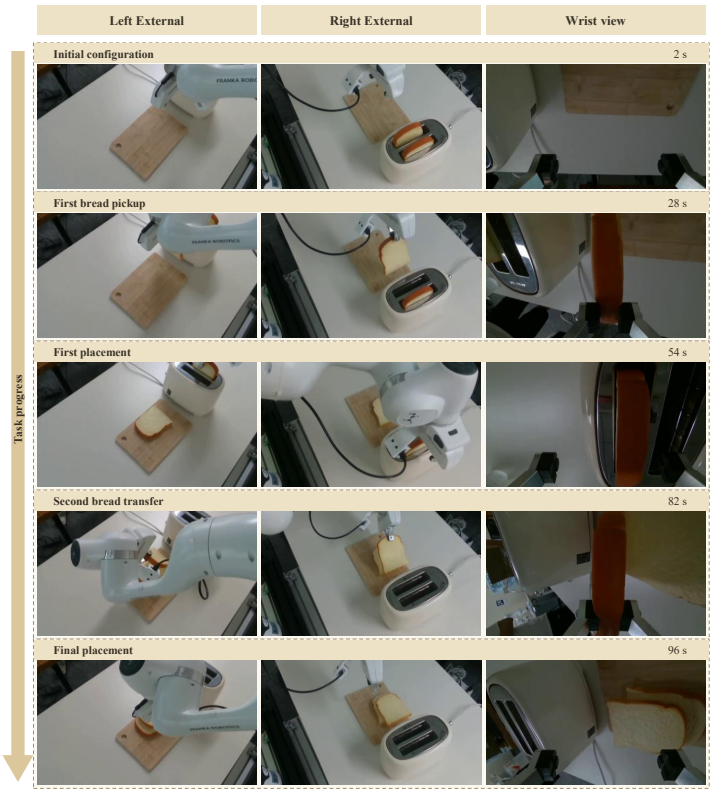

Figure 15: Bread Placement through the robot's cameras. Columns show the left external, right external, and wrist views; rows advance in time, with corresponding views within each row. This rollout is separate from Figure 14; timestamps are relative to each recording.

A.2.4 PEN COLLECTION

Instruction. The recorded task instruction is "Pick up the pens one at a time and place them into the pen holder."

Task Description. Two pens are initially placed on the tabletop in front of the FR3 platform. The robot grasps the pens one at a time, transfers each pen toward the pen holder, and releases it into the holder. The task ends when both pens have been successfully placed inside the pen holder.

Task Visualization. Figures 16 and 17 illustrate Pen Collection from third-person and robot-camera views, respectively.

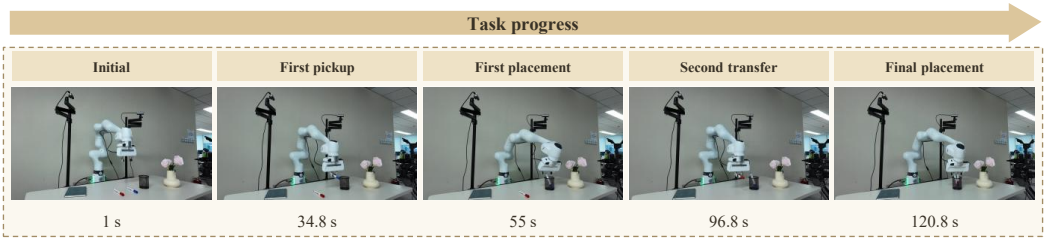

Figure 16: Pen Collection from a third-person viewpoint. Read left to right; timestamps mark selected frames from one rollout.

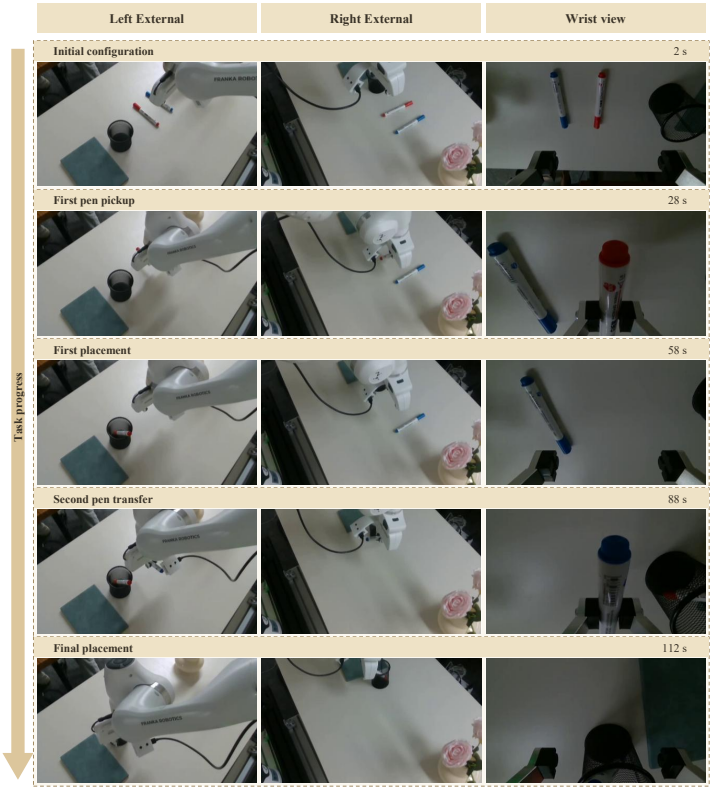

Figure 17: Pen Collection through the robot's cameras. Columns show the left external, right external, and wrist views; rows advance in time, with corresponding views within each row. This rollout is separate from Figure 16; timestamps are relative to each recording.

B ROBOTIC PRETRAINING

Data. The additional robotic pretraining variant of WorldSync uses AgiBot-World-Alpha (Team AgiBot-World, 2025) and DROID (Khazatsky et al., 2024). After cleaning, we retain 32,836 AgiBot episodes: 31,953 gripper episodes and 883 dexterous-hand episodes. From DROID's cleaned subset of 92,998 episodes, we exclude 81 that are shorter than the required 33-frame clip, leaving 92,917 episodes. Table 3 summarizes the retained data.

Table 3: Robotic pretraining data after filtering. Steps count recorded timesteps, including episode endpoints, rather than optimizer updates or sampled clips; synchronized camera views are counted once. Hours give the corresponding video duration. Mean length is the number of recorded steps per episode, with the total-row mean computed over all retained episodes.

Dataset	Episodes	Steps	Mean length (steps)	Hours
AgiBot-World-Alpha	32,836	55,255,078	1,682.76	511.62
DROID	92,917	26,959,694	290.15	499.25
Total	125,753	82,214,772	653.78	1,010.87

Action Space. Both datasets use a unified 20-dimensional action representation with ten dimensions per arm: three for translation increments, six for relative rotation, and one for gripper opening at the next timestep. Pose increments are computed between adjacent observed states in the robot base frame. Given successive end-effector rotations R_t and R_{t+1}, we form the relative rotation $\Delta R_t = R_{t+1} R_t^\top$. We use a 6D rotation representation by concatenating the first two columns of ΔR_t. We apply min–max normalization only to translation. The six rotation components are already bounded in $[-1, 1]$; we retain their original values to preserve the unit-length, orthogonal column structure without dataset-dependent rescaling. Gripper opening ranges from zero (closed) to one (open).

We adapt this representation to each dataset. DROID's single-arm actions use the ten components allocated to the right arm. The unused left-arm components encode zero translation, identity rotation, and a zero gripper value. For AgiBot dexterous-hand episodes, we retain end-effector motion and set the gripper components to zero without encoding individual finger motion.

Video and Sampling. Each training clip contains one current frame, 32 future frames, and 32 actions. All three views are resized to 320×256 pixels. AgiBot uses head and wrist views, while DROID's two external cameras and wrist camera are assigned to the three view inputs in that order. Language conditions use episode descriptions; the 18,863 retained DROID episodes without descriptions use an empty string.

Clips are sampled with a stride of 32, with a final overlapping clip covering each episode's tail. All clips stay within episode boundaries and contain real future frames. This yields 1,741,651 AgiBot clips and 884,670 DROID clips. We shuffle and traverse these clips without replacement, giving each dataset a sampling share proportional to its clip count.

Training Configuration. We initialize from the official Wan2.2-TI2V-5B checkpoint (Team Wan et al., 2025) and train the video diffusion transformer (DiT) and Action AdaLN modules described in Appendix D, using the future-video flow-matching objective, keeping the VAE and text encoder frozen. AFE and IE are disabled during pretraining. We train for one epoch on 16 NVIDIA A100 GPUs over 10 days, accumulating gradients over four microbatches per optimizer update for a global batch size of 128. Training uses bfloat16 precision, ZeRO stage 2, and gradient checkpointing.

We use AdamW with a learning rate of 10^{-5}, 500 warmup steps, and cosine decay to 5×10^{-7}. The optimizer uses $(\beta_1, \beta_2) = (0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 10^{-8}, and gradient clipping at a norm of 1.0. The random seed is 45. Model weights are saved every 5,000 updates and at the end of pretraining.

Baseline Pretraining. In Table 1, Emb. PT indicates training on curated robot or human egocentric interaction data before our task-specific fine-tuning, with or without explicit action labels. A checkmark includes such data in an earlier backbone-training stage; a dash denotes generic video pretraining only, not training from scratch. Our CtrlWorld baseline starts from its DROID-trained checkpoint, whereas WorldSync without additional robotic pretraining starts from Wan2.2-TI2V-5B. The other baseline initializations are Cosmos-Predict2.5-2B's action-conditioned checkpoint, Cosmos3-Nano, DreamDojo-2B's GR-1 post-trained checkpoint, Motus's stage-2 checkpoint, and LingBotVA's base checkpoint.

The reported corpora provide context for these initializations. CtrlWorld (Guo et al., 2026b) is trained on DROID with approximately 95k trajectories across 564 scenes. Cosmos-Predict2.5 (NVIDIA et al., 2025) includes approximately 481k robot-domain videos, obtained by summing the camera-view counts in its robotics data table; these are not unique episodes or hours. Cosmos3 (NVIDIA, 2026) reports 61.3k hours for action mid-training across egocentric motion, robotics, driving, and camera motion, including 41.3k hours of egocentric motion and 5.36k hours of robotics. DreamDojo (Gao et al., 2026) reports 44,711 hours of egocentric human video before robot post-training, while LingBotVA (Li et al., 2026a) reports approximately 16k hours of robot manipulation data. Our additional robotic pretraining uses 1,010.87 hours from AgiBot-World-Alpha and DROID (Table 3), counting synchronized views once. These figures describe the reported data stages, not cumulative training exposure or compute. Differences in data domains, counting rules, supervision, and training schedules preclude interpreting their ratios as controlled data-efficiency comparisons.

C Policy Improvement Approach

Following the diagnose–improve–validate workflow in the Introduction, we evaluate world-model quality through downstream policy improvement. While WorldEcho directly measures visual integrity and action following, policy success after post-training assesses the value of generated experience for learning to complete a task. We build on VLAW (Guo et al., 2026a) to compare world models through a common policy-improvement procedure. §C.1 examines failures in our reproduction and explains the resulting changes to the pipeline. §C.2 describes the shared experimental protocol and implementation, and §C.3 reports rollout retention and policy success over two improvement rounds.

C.1 Diagnosis and Adaptations

Reproduction and diagnosis. VLAW (Guo et al., 2026a) alternates world-model adaptation and policy improvement using online and imagined experience. In our reproduction with CtrlWorld and Qwen-OFT on RoboTwin's `rotate_qrcode` task, policy success decreased from 59% to 46% and then 14% (Table 4). This decline motivates examining whether imagined rollouts provide reliable observation–action supervision. We use this reproduction to diagnose failures before comparing world models under the adapted pipeline on the bin-dumping task below.

Table 4: **Reproduction of the VLAW framework.** Policy success rates (%) on `rotate_qrcode`, measured over 100 episodes at each stage. Both CtrlWorld and Qwen-OFT are updated each round.

Setting	Initial	Round 1	Round 2
CtrlWorld + Qwen-OFT	59	46	14

Two weaknesses undermine this supervision. First, generated rollouts may fail to advance the task or depict apparent progress through visual artifacts, such as distorted arms or disappearing grippers (Figure 2). Second, visually plausible motion may not follow the conditioning actions. Figure 18 shows a bin-dumping example: the policy's actions fail to pick up the cup and empty its contents in the simulator, yet the world model predicts successful lifting and pouring. This mismatch is consistent with a bias toward successful expert behavior when responding to off-expert actions. This discrepancy motivates screening both predicted task progress and action consistency before using imagined rollouts for policy training.

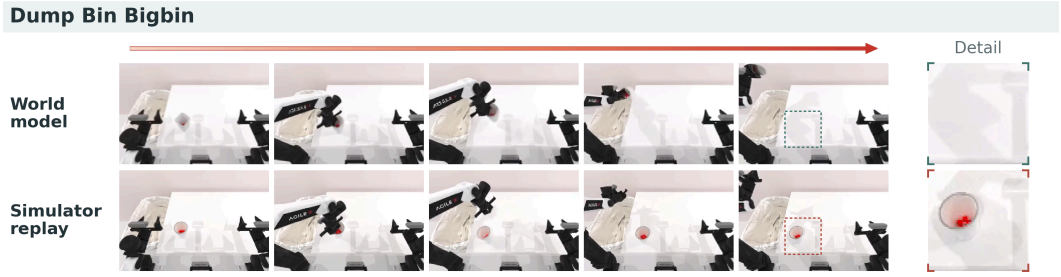

Figure 18: **Action mismatch under identical conditioning actions.** World-model prediction (top) and simulator replay (bottom) on `dump_bin_bigbin`, using the same actions from matched initial conditions. The prediction depicts lifting and pouring, whereas the simulator replay leaves the cup and its contents in place. The arrow indicates temporal order; rightmost panels magnify the boxed final-frame regions.

Adaptations to VLAW. We retain VLAW's iterative organization of online collection, world-model adaptation, closed-loop imagination, and supervised policy updates. Within this organization, we refine how imagined experience is selected and adapt the policy update to Qwen-OFT. We use Robo-Dopamine (Tan et al., 2026b) as the task-progress reward model $\mathcal{R}$ to remove rollouts with

insufficient predicted progress. Because apparent progress can coexist with action mismatch, action-consistency filtering then combines video–action rotation agreement with a paired-expert reference feature, using AnyPos (Tan et al., 2026a) as the pose extractor Φ. The first stage selects predicted progress, while the second checks the associated motion. Neither score replaces the simulator's success label.

Online world-model adaptation addresses the changing policy distribution. It uses current-round successes and failures together with offline data, while successful online rollouts accumulate for policy training. For policy training, an L1 behavior-cloning objective combines those online successes with retained imagined rollouts and expert demonstrations. We apply the resulting pipeline to both world-model conditions under the shared budgets specified in §C.2.

Imagined rollouts are regenerated each round, and both updated models initialize the next round. Algorithm 2 specifies the update sequence. Here π_k and $\mathcal{W}_{\theta_k}$ denote the policy and world model before round $k+1$, and $\mathcal{E}$ is the simulator.

Algorithm 2 Policy Improvement with Filtered Imagined Rollouts

Require: Initial models $(\pi_0, \mathcal{W}_{\theta_0})$, offline data $\mathcal{D}_{\text{off}}$, expert demonstrations $\mathcal{D}_{\text{exp}}$, environment $\mathcal{E}$, reward model $\mathcal{R}$, pose extractor Φ, fixed filter calibration and evaluation seeds $\mathcal{S}_{\text{eval}}$

Initialize cumulative online successes $\mathcal{D}^{+}_{\text{on}} \leftarrow \emptyset$
Evaluate π_0 on fixed evaluation seeds $\mathcal{S}_{\text{eval}}$
for $k = 0, 1$ **do**
  (a) Online collection and world-model adaptation
  Collect online rollouts $\mathcal{D}^{k}_{\text{on}}$ using π_k in $\mathcal{E}$
  Accumulate simulator-verified successes in $\mathcal{D}^{+}_{\text{on}}$
  Update $\mathcal{W}_{\theta_k} \rightarrow \mathcal{W}_{\theta_{k+1}}$ using current $\mathcal{D}^{k}_{\text{on}}$ and $\mathcal{D}_{\text{off}}$ (ordinary sampling $1:1$), with the condition-specific objective and auxiliary data in §C.2
  (b) Imagined rollout generation
  Generate imagined rollouts $\mathcal{D}^{k}_{\text{syn}}$ from expert starts using π_k and $\mathcal{W}_{\theta_{k+1}}$ in closed loop
  (c) Sequential filtering
  Select $\mathcal{D}^{k}_{\mathcal{R}} \leftarrow \{z \in \mathcal{D}^{k}_{\text{syn}} : s_{\mathcal{R}}(z) > 0.8\}$
  Extract visual poses with Φ and reconstruct trajectories from conditioning actions
  Compute action-consistency scores $q_k(z)$ using Eq. (15)
  Select $\mathcal{D}^{k}_{\text{keep}} \leftarrow \{z \in \mathcal{D}^{k}_{\mathcal{R}} : q_k(z) \le \eta\}$
  (d) Policy update and evaluation
  Construct policy-training examples from $\mathcal{D}^{k}_{\text{keep}}$
  Fine-tune $\pi_k \rightarrow \pi_{k+1}$ on $\mathcal{D}^{+}_{\text{on}}$, $\mathcal{D}^{k}_{\text{keep}}$, and $\mathcal{D}_{\text{exp}}$ (sampling $1:1:1$)
  Evaluate π_{k+1} on $\mathcal{S}_{\text{eval}}$
end for
return $(\pi_2, \mathcal{W}_{\theta_2})$

C.2 Experimental Setup

Task, initialization, and controls. We detail the RoboTwin `dump_bin_bigbin` task, hereafter the bin-dumping task, evaluated in Figure 6. All conditions start from the same Qwen-OFT policy checkpoint and their respective world-model checkpoints. We compare WorldSync with CtrlWorld trained on expert demonstrations. The policy initialization, expert demonstrations and starts, filter settings, per-round budgets, and evaluation seeds are shared across conditions (Table 5). Online observations and retained imagined rollouts may differ as the policies evolve, while the number of policy optimizer updates remains fixed. World-model architectures, training objectives, and offline data remain condition-specific, as described below.

Each arm's action contains three translation increments, a 6D rotation representation of the relative rotation, and one gripper value. Together, the two arms give a 20-dimensional action (Appendix B).

Online collection and world-model adaptation. To follow the updated policy's state–action distribution, each round adapts the world model using all 30 current online rollouts, including failures. Offline replay preserves the original training distribution: WorldSync uses its original expert/off-

Table 5: **Shared simulation protocol.** Counts are per condition and per round unless stated otherwise. Sampling ratios do not denote unique rollout counts.

Setting	Configuration
Improvement rounds	2
Online collection	30 rollouts, including successes and failures
Imagined rollouts	$3{,}000 = 50$ expert starts $\times 60$
Action representation	20-dimensional actions, 32 actions per chunk
Generation horizon	7 chunks (224 actions), 20 denoising steps per chunk
World-model update	1,000 updates, global batch 16, learning rate 10^{-5}
World-model sampling	Current online : offline $= 1 : 1$ for ordinary sampling
Policy sampling	Cumulative online successes : current retained imagined : expert $= 1 : 1 : 1$
Policy batch size	Global batch 32
Policy optimizer / loss	AdamW, learning rate 10^{-5}, gradient clip 1, L1 behavior cloning
Evaluation	Same ordered 100 seeds, initially and after each round

expert mixture, whereas CtrlWorld uses expert data. WorldSync retains video flow matching, AFE, and IE. Its intervention-pair sampling fraction is 0.25, separate from the ordinary online:offline mixture. CtrlWorld retains its native video-denoising objective.

Sequential rollout filtering. To select useful experience without relying on apparent task completion alone, we apply the complementary filters motivated in §C.1. First, we use the Robo-Dopamine-GRM-2.0-8B-Preview checkpoint as $\mathcal{R}$ to score predicted task progress from the instruction, generated multi-view observations, and paired expert terminal image. It evaluates frame pairs at a nominal stride of eight, up to eight pairs. For longer videos, it samples nine uniformly spaced frames including the endpoints. With incremental scoring disabled, the rollout score $s_{\mathcal{R}}(z)$ is the maximum over evaluated frames and forward/backward scoring modes. Only rollouts with $s_{\mathcal{R}}(z) > 0.8$ proceed to action-consistency filtering.

For each such rollout z, Φ estimates both end-effector poses from the generated head-camera video. We reconstruct action-conditioned trajectories from the paired initial robot state using $p_t = p_{t-1} + \Delta p_t$ and $R_t = \Delta R_t R_{t-1}$. Both trajectories are expressed in physical units and aligned at video timestamps. Their relative rotations, $\bar{R}_t = R_t R_0^\top$, define a geodesic error $e_{\mathrm{rot}}(z)$ in degrees, averaged over both arms and aligned frames. We normalize this error within the reward-passing set $\mathcal{D}^k_{\mathcal{R}}$:

$$r_k(z) = \frac{e_{\mathrm{rot}}(z)}{\mathrm{median}_{z' \in \mathcal{D}^k_{\mathcal{R}}}\, e_{\mathrm{rot}}(z')}. \tag{14}$$

An expert-referenced feature provides additional information about the action-conditioned motion. We select the arm with the longer expert Cartesian path, choosing the left arm in a tie. For candidate and expert separately, closure is the first frame of a four-frame window containing at least three gripper values below 0.5, or the final frame if none exists. The feature $d(z)$ is the Euclidean distance in meters between the candidate's action-integrated position and the expert position at their respective closure frames. It is a closure-position proxy, not a measured physical-contact event. The joint retention rule is

$$q_k(z) = \frac{r_k(z) - \mu_r}{s_r} - \lambda \frac{d(z) - \mu_d}{s_d}, \qquad \mathcal{D}^k_{\mathrm{keep}} = \{z \in \mathcal{D}^k_{\mathcal{R}} : q_k(z) \le \eta\}. \tag{15}$$

Here μ and s are calibration medians and interquartile ranges. These statistics, the closure-feature weight λ, and the retention threshold η remain fixed across conditions and rounds. Only the rotation batch median in Eq. (14) is recomputed for each condition and round. Because the implemented joint score assigns a negative coefficient to $d(z)$, the rule does not impose separate upper bounds on both features. Filtering stops if pose extraction is incomplete or feature statistics are invalid.

C.3 RESULTS

Sequential rollout filtering. Table 6 tracks each round's 3,000 imagined rollouts through both filters, reporting rollout counts and retention rates.

Table 6: **Sequential filtering counts and retention rates.** Stage 1 applies task-progress filtering with $s_{\mathcal{R}}(z) > 0.8$, followed by action-consistency filtering with $q_k(z) \leq \eta$ in Stage 2. Stage-wise rates use the preceding stage's count, whereas overall rates use all 3,000 generated rollouts.

		Rollout counts: $N_0 \to N_1 \to N_2$			**Retention (%)**		
Method	**Round**	**Initial** N_0	**After S1** N_1	**After S2** N_2	**Stage 1** N_1/N_0	**Stage 2** N_2/N_1	**Overall** N_2/N_0
WorldSync	1	3,000	1,114	553	37.1	49.6	18.4
WorldSync	2	3,000	1,088	365	36.3	33.5	12.2
CtrlWorld	1	3,000	962	452	32.1	47.0	15.1
CtrlWorld	2	3,000	909	323	30.3	35.5	10.8

Under the same generation budget and filtering criteria, WorldSync retained more imagined rollouts than CtrlWorld in both improvement rounds. The final counts were 553 versus 452 in Round 1 and 365 versus 323 in Round 2, corresponding to 22.3% and 13.0% more retained rollouts, respectively. Thus, WorldSync supplied more imagined rollouts that met both the task-progress and action-consistency criteria for policy training from the same number of generation attempts. We next examine whether this higher data yield was accompanied by larger policy gains under the fixed policy-training budget.

Downstream policy success. Table 7 gives the exact simulation values plotted in Figure 6. Under the shared pipeline and budgets, WorldSync achieved higher policy success after both rounds than CtrlWorld. It reached 65% after two rounds, gaining 13 percentage points, whereas CtrlWorld gained 5 points and finished at 57%. The higher yield of accepted imagined rollouts and larger policy gains together support higher world-model quality for downstream policy learning under the evaluated protocol. Combined with the direct measurements in WorldEcho, these results provide complementary evidence for improved action following and policy-learning utility.

Table 7: **Policy success over two improvement rounds.** Success rates use the same ordered set of 100 evaluation seeds. Gains are percentage-point changes from each condition's measured initial rate.

Method	**Initial (%)**	**Round 1 (%)**	**Round 2 (%)**	**Gain (pp)**
WorldSync	52	57	**65**	**+13**
CtrlWorld	52	54	57	+5

D ANALYSIS OF ACTION-CONDITIONING MECHANISMS

D.1 ACTION-CONDITIONING MECHANISMS

Action conditioning is central to robot world models because the model must translate commanded robot motion into its visual consequences. For video-DiT-based world models, this requirement raises two coupled design questions of how actions should be represented and where they should interact with video features. We therefore systematically compare nine mechanisms selected from representative works and reimplemented within the same backbone, organizing them along four design axes of action representation, temporal or spatial alignment, injection site, and fusion. Table 8 defines these mechanisms, while Figure 19 illustrates their corresponding data flows through the DiT backbone.

Six numerical-based mechanisms retain actions as low-dimensional signals. Additive follows NanoWM (Huang et al., 2026a) and adds frame-aligned action features directly to the patch features. AdaLN Fusion keeps the same frame alignment but routes actions through the native AdaLN

Table 8: Action-conditioning mechanisms and representative works. All variants use the same video diffusion backbone.

Mechanism	Representative work	Alignment	Injection site	Fusion
Numerical-based				
Additive	NanoWM (2026a)	Frame-aligned	Patch features	Element-wise addition
AdaLN Fusion	Cosmos-Predict2.5 (2025)	Frame-aligned	Native AdaLN path	Native-condition fusion
Action AdaLN	IRASim (2025)	Frame-aligned	Per-block attention/FFN	Low-rank shift/scale/gate
Cross-Attention	CtrlWorld (2026b)	Frame-aligned, local	Native attention	Gated action attention
Shared-KV Attention	Motus (2025)	Global sequence	Native self-attention	Shared video/action K/V; video Q
Latent Frame	Cosmos Policy (2026)	Global chunk	Latent condition slot	Native attention across slots
Map-based (shared RGB trajectory maps)				
Channel Concatenation	GE-Sim 2.0 (2026)	Latent-aligned	Patch input	Channel concatenation
Residual Injection (ControlNet-style)	ABot-PhysWorld (2026d)	Latent-aligned	Selected DiT blocks	Zero-initialized residuals
Sequence Concatenation	Action Images (2026)	Full map video	Action/video sequence	Temporal concatenation; video-only loss

path, where the action embedding is fused into the timestep pathway that drives the backbone's adaptive layer normalization, as in Cosmos-Predict2.5 (NVIDIA et al., 2025) and DreamDojo (Gao et al., 2026). Action AdaLN then moves this modulation from this shared conditioning path into each block; following IRASim (Zhu et al., 2025), it produces low-rank shift, scale, and gate offsets for the attention and feed-forward sublayers. The next two mechanisms instead introduce actions through attention. Cross-Attention follows CtrlWorld (Guo et al., 2026b) and lets video tokens query a separate action branch through a gated attention output. Shared-KV Attention follows Motus (Bi et al., 2025) and merges video and action keys and values in the native self-attention while retaining video-only queries. DreamZero (Ye et al., 2026) shares this joint video–action attention design, but further assigns action tokens their own queries and prediction targets. Finally, Latent Frame follows Cosmos Policy (Kim et al., 2026) and reshapes a global action chunk to fill a dedicated latent-frame-sized slot in the native DiT sequence. Together, these representative mechanisms span feature addition, adaptive modulation, attention-based fusion, and latent-frame conditioning.

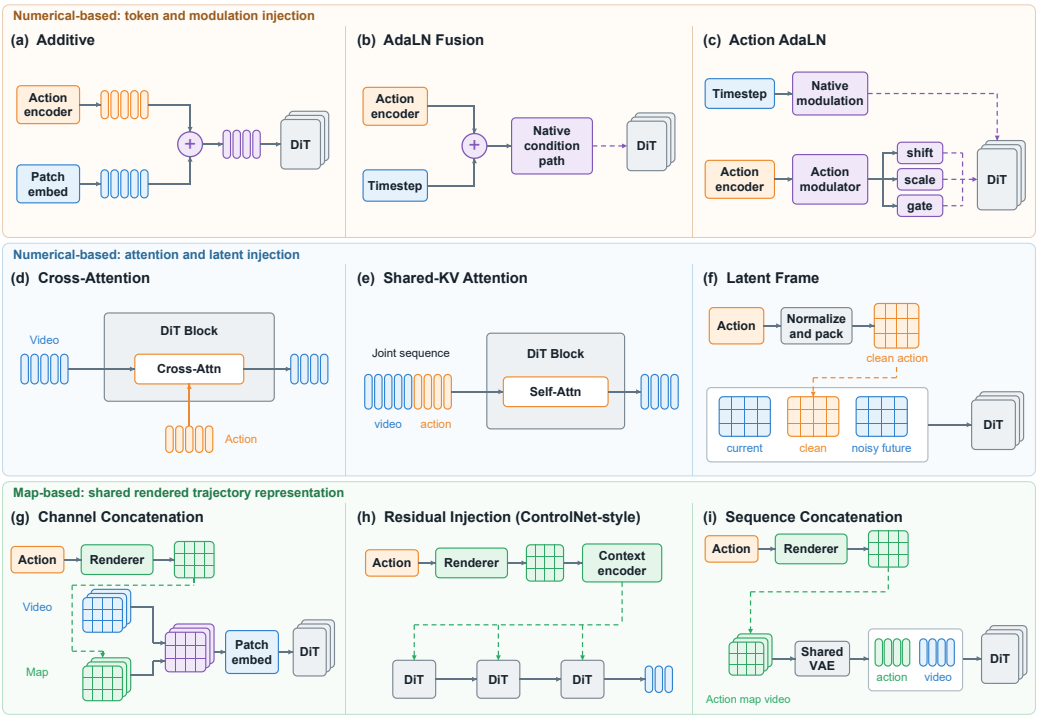

Figure 19: Overview of the evaluated action-conditioning mechanisms. Numerical-based methods use feature addition, modulation, attention, or latent frames, whereas map-based methods use rendered action maps. Blue, orange, green, purple, and gray denote video features, numerical actions, action maps, fusion operations, and DiT blocks, respectively.

However, compact numerical signals do not explicitly indicate where the commanded motion should appear in each camera view. Some studies instead project robot motion into image-aligned maps, providing a direct spatial correspondence between actions and predicted visual changes. We therefore evaluate three map-based mechanisms that share the RGB trajectory representation used by GE-Sim 2.0 (Qiu et al., 2026). As shown in Figure 20, we project the end-effector poses into each camera view and render trajectories over the prediction horizon. Channel Concatenation follows GE-Sim 2.0 and injects these maps at the patch input. Residual Injection adapts the ControlNet-style pathway of ABot-PhysWorld (Chen et al., 2026d), adding map features to selected backbone blocks through zero-initialized projections. Sequence Concatenation follows Action Images (Zhen et al., 2026) and places action-map and video latents in a joint temporal sequence. By making spatial alignment explicit, these mechanisms aim to give the model a more direct cue for where action-induced changes should appear. Holding the action-map representation fixed further isolates the effect of their injection paths.

D.2 Experimental Setup

All nine mechanisms are implemented on the same video diffusion backbone, Wan2.2-TI2V-5B (Team Wan et al., 2025), a 5B-parameter video DiT initialized from the official checkpoint. All variants are trained with the same flow-matching objective on the same simulated trajectory mixture, which spans demonstrated and off-expert actions. Each model generates 32 future frames across three camera views from the current observation, conditioned on a 32-step action chunk in the shared end-effector action space used by WorldSync. Only the representation and injection of the action condition differ across variants. We save checkpoints every 10k training steps up to 40k; the fixed 40k checkpoint supports the primary comparison in Table 9, and Figure 21 examines the full checkpoint series.

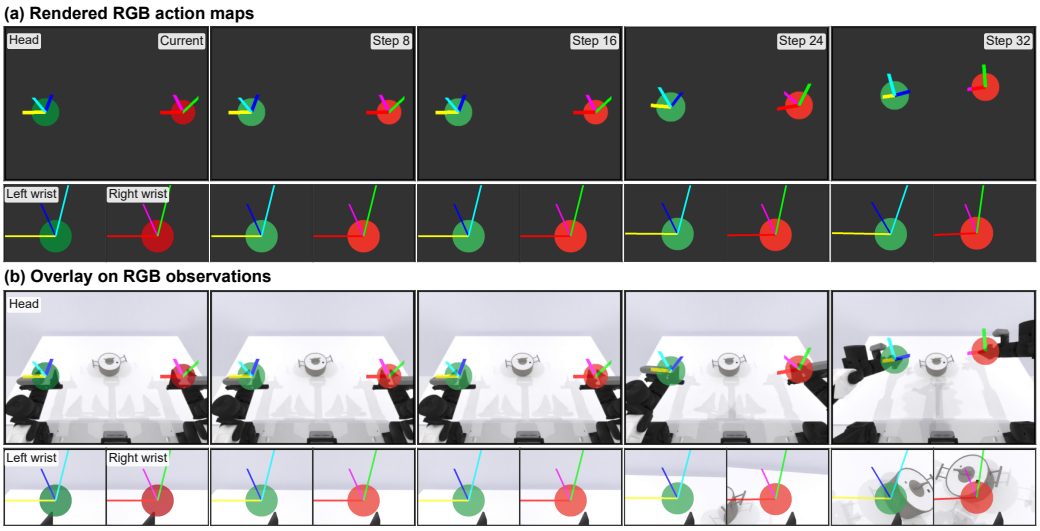

Figure 20: RGB action-map representation. (a) Rendered action maps at the current frame and four future steps. (b) Corresponding overlays on the RGB observations. Each column shows the head, left-wrist, and right-wrist views; colored markers encode the projected end-effector poses.

Table 9: Comparison at the last training checkpoint. Params and FLOPs denote additional trainable parameters and inference compute relative to unconditioned video generation. Best performance is bold.

Mechanism	Raw NDTW ↓	Gated ↓	Pass (%) ↑	Params (M)	FLOPs (T)
Additive	0.0320	0.0403	96.43	568.07	0.04
AdaLN Fusion	0.0330	0.0389	97.86	568.07	4.93
Action AdaLN	**0.0137**	**0.0176**	**98.57**	572.01	0.09
Cross-Attention	0.0302	0.0354	97.86	566.49	0.82
Shared-KV Attention	0.0332	0.0370	**98.57**	566.49	1.27
Latent Frame	0.0239	0.0297	97.86	0	69.39
Channel Concatenation	0.0342	0.0379	**98.57**	0.04	< 0.01
Residual Injection (ControlNet-style)	0.0338	0.0413	97.14	88.17	7.61
Sequence Concatenation	0.0318	0.0392	97.14	0	520.90

All variants are evaluated following the WorldEcho protocol (Algorithm 1), generating each rollout with 20 denoising steps. Each variant is evaluated on four RoboTwin tasks and all five action-query categories. We report raw pose-aware NDTW, integrity-gated error, and visual pass rate. Raw NDTW measures the agreement between the generated and ground-truth end-effector trajectories, the integrity-gated error assigns a fixed penalty κ=0.3 to visually invalid rollouts, and the visual pass rate isolates the frequency of valid rollouts; we interpret the three jointly rather than as a single ranking. Table 9 additionally reports the trainable parameters added by each mechanism and the additional inference compute relative to unconditioned video generation, estimated at the operator level over 20 denoising steps. These performance and cost dimensions are interpreted separately.

D.3 Comparison Results

Table 9 reports the common 40k checkpoint under the WorldEcho evaluation protocol. Action AdaLN attains the lowest raw NDTW (0.0137) and integrity-gated error (0.0176), and shares the highest visual pass rate (98.57%) with Shared-KV Attention and Channel Concatenation. Latent

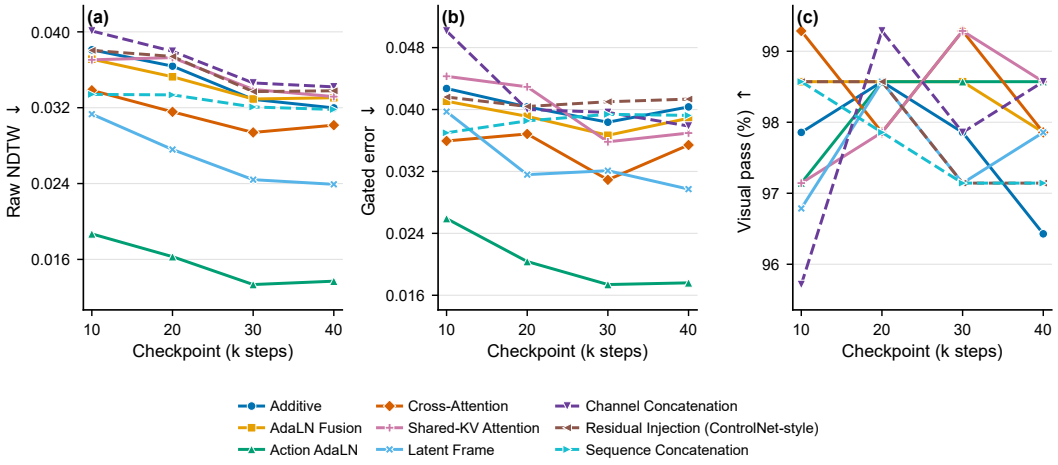

Figure 21: Performance across training checkpoints under the WorldEcho evaluation protocol. The panels report raw NDTW, integrity-gated error, and visual pass rate for all nine mechanisms. Raw and gated errors are task-macro averages; pass rate counts samples. Solid and dashed lines denote numerical-based and map-based mechanisms, respectively. Latent Frame uses equal-weight aggregation of its component results.

Frame has the second-lowest raw NDTW (0.0239) and gated error (0.0297), with a 97.86% pass rate. The remaining seven mechanisms have raw NDTW values of 0.0302–0.0342. Visual pass rates are similar across mechanisms, while substantial differences in trajectory agreement remain.

Integrity-gated error combines trajectory agreement with a fixed penalty κ=0.3 for visually rejected clips. At 40k, pass rates range from 96.43% to 98.57%; the remaining failures still increase gated error above raw NDTW. Channel Concatenation has the lowest gated error among map-based methods (0.0379), despite having their highest raw NDTW (0.0342). We therefore read raw NDTW and pass rate separately and gated error as their synthesis. In this comparison, the three map-based mechanisms have higher raw NDTW than Action AdaLN despite their explicit image-aligned spatial correspondence. Their raw NDTW values are close to those of Additive, AdaLN Fusion, Cross-Attention, and Shared-KV Attention.

Figure 21 shows that these relationships are stable across training. Action AdaLN attains the lowest raw NDTW and gated error at every displayed checkpoint, although its errors bottom out near 30k and rise slightly at 40k, so its advantage does not grow monotonically with longer training. Latent Frame's raw NDTW decreases across the displayed checkpoints, while its gated error is not strictly monotonic. Visual pass rates across all 36 points range from 95.71% to 99.29%, and no single mechanism leads this metric at every checkpoint. We therefore interpret the curves as descriptive evidence of trend stability and trade-offs, without claiming statistical significance from them.

D.4 Computational Cost

The nine mechanisms introduce overhead through three distinct profiles (Table 9). The additive, modulation, and attention variants are parameter-dominated, with Additive, AdaLN Fusion, Action AdaLN, Cross-Attention, and Shared-KV Attention each adding 566.49–572.01M trainable parameters but little additional inference compute (0.04–4.93 TFLOPs). The sequence-extending variants are compute-dominated. Latent Frame and Sequence Concatenation add no trainable parameters, yet their additional latent tokens are processed by the full backbone at every denoising step, costing 69.39 and 520.90 TFLOPs, respectively. Channel Concatenation is nearly free, with 0.04M parameters and less than 0.01 TFLOPs, but has higher trajectory error than Action AdaLN and Latent Frame. Parameter count and inference compute are therefore decoupled, since a mechanism can be parameter-free yet expensive per rollout or parameter-heavy yet cheap to run. Neither number alone summarizes the cost of a design.

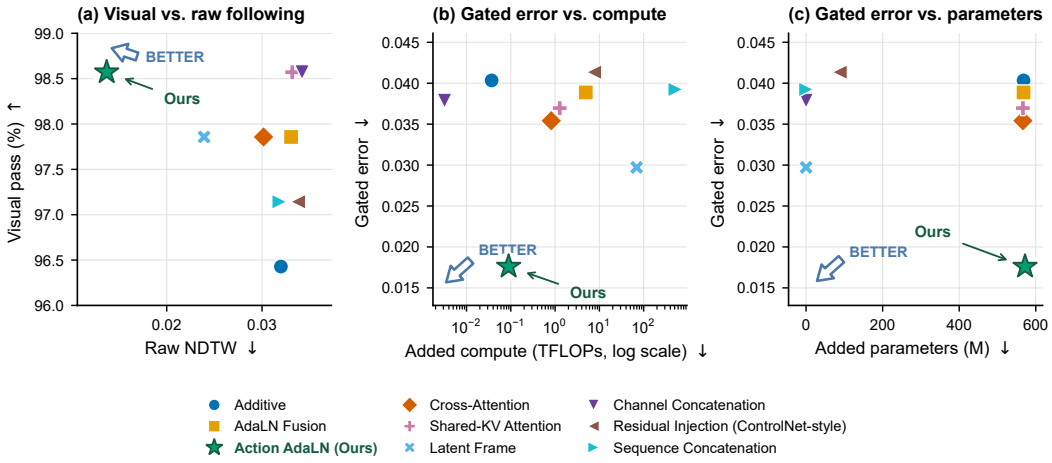

Figure 22: Performance–efficiency trade-offs at the common 40k checkpoint under the WorldEcho evaluation protocol. (a) Visual pass rate versus raw NDTW. (b) Integrity-gated error versus additional inference compute on a logarithmic scale. (c) Integrity-gated error versus additional parameters. Arrows indicate preferred directions, and the star marks Action AdaLN (Ours).

Figure 22 relates these costs to performance. In panel (a), Action AdaLN has the lowest raw NDTW and ties for the highest visual pass rate; Latent Frame has the second-lowest raw NDTW. Panels (b) and (c) show that no mechanism is simultaneously the cheapest and the most accurate. Sequence Concatenation pays the highest additional compute for a gated error of 0.0392, the parameter-free variants are not automatically stronger, and Action AdaLN pairs the lowest gated error with only 0.09 TFLOPs of additional compute although it is not the most parameter-efficient design. We note that the reported FLOPs are operator-level estimates over 20 denoising steps and do not directly measure wall-clock latency.

D.5 Discussion and Design Choice

Two conclusions emerge from this study. First, no action-conditioning mechanism resolves action following on its own. Across feature addition, adaptive modulation, attention-based fusion, latent-frame conditioning, and rendered action maps, every variant still trades trajectory agreement against visual integrity to some degree, and every variant adds parameters, inference compute, or both. Improving the injection pathway shifts the operating point but does not close the gap. Our full model therefore pairs its injection mechanism with broader action coverage and auxiliary training objectives rather than relying on injection design alone.

Second, Action AdaLN achieves the lowest errors with a simple modulation design. It modulates the attention and feed-forward sublayers of each DiT block through low-rank shift, scale, and gate offsets, without extra tokens, a separate attention branch, or rendered maps. Yet this standard modulation design attains the lowest raw NDTW and integrity-gated error in Table 9, retains a 98.57% visual pass rate, and adds only 0.09 TFLOPs of inference compute. The more elaborate alternatives, including extra latent slots, joint action–video sequences, ControlNet-style residual branches, and explicit spatial maps, fail to surpass this strong baseline on either error metric.

We therefore adopt Action AdaLN in our full model. This choice should not be read as uniform superiority: Shared-KV Attention and Channel Concatenation match its visual pass rate, while Latent Frame adds no trainable parameters. What Action AdaLN offers is the most favorable balance of accurate action following, visual integrity, inference compute, and architectural simplicity among the nine candidates, which is the combination that matters for a general-purpose robot world model.

E Additional Ablation Studies

E.1 Action-Forcing Expert Ablations

Motivation and design space. Intervention-Effect (IE) supervision compares generated futures under paired actions. AFE adds a complementary constraint within each rollout: intermediate video features should retain the robot-state changes induced by the actions. This training-only objective raises three design questions. *Where to read?* Features at different backbone depths may differ in their usefulness for state prediction. *When to read?* Restricting a future-step query to its corresponding video time may focus supervision but remove useful context. *What to predict?* Absolute state, stepwise change, and initial-relative cumulative change emphasize different aspects of robot motion. We compare these choices to select the AFE supervision.

Design choices. AFE architecture. Figure 23(a) shows the two input paths. Actions condition the Wan video backbone; AFE reads its intermediate video-token features. Ground-truth robot states enter only the supervision path, where they define targets for the auxiliary loss. They are not inputs to AFE. Wan-VAE compresses the current frame and 32 future frames into nine latent time positions. The backbone processes these noisy latents with visual, language, and action conditioning.

AFE maintains $H = 32$ learnable queries q_h, one per future step, arranged as $Q^0 = [q_1; \ldots; q_H] \in \mathbb{R}^{H\times d}$, where d is the query feature dimension. The $(i+1)$-th of eight sequential blocks reads the zero-based backbone layer $\ell_i = 8 + 2i$ and updates Q^i to Q^{i+1}, for $0 \le i < 8$. As shown in Figure 23(b), each block uses global query self-attention, video-feature cross-attention, and a feed-forward network, with pre-normalization and residual connections. A shared LayerNorm–linear projection maps the backbone features into the AFE feature space.

Let $F_j^{(\ell)}$ contain all spatial tokens from the three camera views at backbone layer ℓ and latent time j; $j = 0$ denotes the current frame. Temporal alignment restricts query q_h's video-feature cross-attention to $F^{(\ell)}_{\lceil h/4\rceil}$. Equivalently, the queries $\{q_h : 4j-3 \le h \le 4j\}$ read future latent time j, for $1 \le j \le 8$ (Figure 23(c)). These queries predict distinct future-step targets despite sharing the same video features. A shared LayerNorm–linear head maps the final queries to the 32×20 prediction $\hat{y}_{1:H}$.

The selected target is *initial-relative cumulative change*: the future end-effector state s_h minus the same current state s_0, rather than the previous step's state. Each $s_h \in \mathbb{R}^{20}$ contains both arms' XYZ position, rotation encoded using the 6D rotation representation, and gripper state. The State target column of Table 10 writes this semantic target as $s_h - s_0$; the actual training target applies coordinate preprocessing and scaling:

$$y_h = \frac{\rho(s_h) - \rho(s_0)}{\sigma_\Delta}, \tag{16}$$

where ρ min–max normalizes XYZ and retains the coordinates of the 6D rotation representation and binary gripper coordinates, and σ_Δ is a frozen per-coordinate standard deviation of the cumulative state changes after applying ρ. Division is element-wise, with $\sigma_\Delta \in \mathbb{R}^{20}$. The queries predict $\hat{y}_{1:H}$ directly and in parallel; no recursive sum of predicted increments is used. A masked mean squared error loss supervises the predictions using valid state labels. This target construction concerns auxiliary supervision, while the world model is conditioned on actions in the shared delta end-effector action space. The loss gradients update AFE and the backbone through the intermediate features. AFE does not write activations into the video stream and is removed at inference. All results below therefore evaluate generated videos rather than auxiliary state predictions.

Design axes. We vary the feature layers, temporal feature access, and state target. For layers, $\ell_i = \lfloor 29i/7 + 1/2 \rfloor$ spans the 30-layer backbone, whereas $\ell_i = 8 + 2i$ selects its middle layers; both use $0 \le i < 8$. For temporal access, the *Latent times* column in Table 10 contrasts unrestricted access ($0 \le j \le 8$) with temporal alignment ($j = \lceil h/4 \rceil$). Both retain all spatial tokens from the permitted times and global query self-attention. For the target, we compare absolute state s_h, stepwise change $s_h - s_{h-1}$, and initial-relative cumulative change $s_h - s_0$. These are semantic descriptions before preprocessing. Absolute-state supervision uses $\rho(s_h)$, whereas stepwise-change supervision uses $[\rho(s_h) - \rho(s_{h-1})]/\sigma_{\text{inc}}$. The selected initial-relative cumulative-change target is

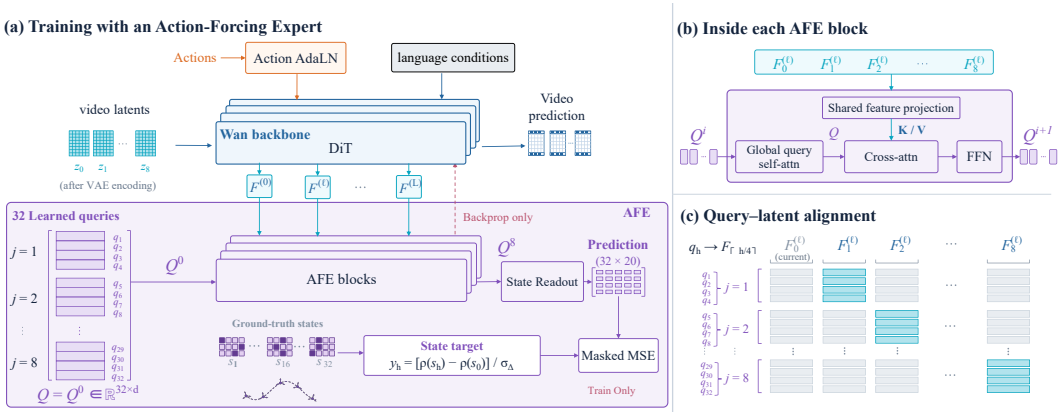

Figure 23: AFE architecture. (a) Training-only state supervision from intermediate video features. (b) Query updates within an AFE block. (c) Temporal alignment between future-step queries and video features. AFE updates the backbone through loss gradients and is removed at inference.

y_h in Equation (16). Each change target has its own frozen per-coordinate standard deviations, estimated after ρ and floored at 10^{-4}, without subtracting a mean.

Experimental setup. We evaluate AFE design choices on the same queries across four RoboTwin tasks. **Gated error is the primary metric.** It assigns each rollout its pose-aware normalized dynamic time-warping error (NDTW) if it passes all visual integrity criteria, and a fixed failure penalty otherwise, then averages within each task and across tasks. NDTW compares generated and GT trajectories using position and rotation, with the accumulated alignment cost divided by the warping-path length. We also report *raw NDTW* and the percentage of rollouts passing all visual integrity criteria (*Visual pass*). Errors are reported in meter-equivalent units and pass rates as percentages. Bold denotes the best unrounded value in each column.

Table 10 reports the design ablation, with IE disabled and all configurations evaluated at the common 40k checkpoint. Its Baseline is the same action-conditioned Wan model trained without either auxiliary objective; every other row adds eight AFE blocks. The individual and joint contributions of AFE and IE are reported in Table 2. The design ablation assesses end-effector visibility using segmentation masks, whereas the component ablation combines masks with text-based detection at every second frame. Their Gated errors and Visual pass rates are therefore not directly comparable.

The qualitative cases use Baseline and AFE checkpoints trained for 5k steps in the design ablation.

Results and analysis. **Design ablation.** At 40k, middle-layer features improve all three metrics relative to spanning layers under the same absolute-state target and unrestricted temporal access (Table 10). Restricting cross-attention to aligned latent times further improves all three metrics for this target. Stepwise-change supervision lowers Raw NDTW relative to absolute-state supervision, but increases Gated error and lowers Visual pass. Cumulative-change supervision achieves the lowest Gated error and Raw NDTW and the highest Visual pass among the tested configurations. Relative to Baseline, it reduces Gated error by 13.02% and Raw NDTW by 27.33%, supporting its selection as AFE.

This sweep selects a configuration, without isolating temporal alignment or target semantics: the change targets also alter coordinate scaling, and cumulative targets with unrestricted temporal access were not tested.

Generated-video evidence. Figure 24 compares GT, Baseline, and AFE for matched initial observations and action sequences in the design ablation. Terminal frames and identical hand-region crops reveal reduced hand-position drift in the four selected cases. The curves report framewise position errors estimated using AnyPos and averaged over both arms, whereas each crop shows one hand. The errors use corresponding frame indices without DTW alignment, exclude rotation, and

Table 10: AFE design ablation at 40k training steps with IE disabled. State targets are shown before preprocessing. Cumulative-change supervision achieves the lowest Gated error and Raw NDTW.

Layers ℓ_i	Latent times j	State target	Gated error ↓	Raw NDTW ↓	Visual pass ↑
Baseline	–	–	0.0717	0.0208	82.14
$\lfloor 29i/7 + 1/2 \rfloor$	$0 \le j \le 8$	s_h	0.0816	0.0210	78.57
$8 + 2i$	$0 \le j \le 8$	s_h	0.0784	0.0197	79.29
$8 + 2i$	$\lceil h/4 \rceil$	s_h	0.0702	0.0191	82.14
$8 + 2i$	$\lceil h/4 \rceil$	$s_h - s_{h-1}$	0.0714	0.0159	80.71
$8 + 2i$	$\lceil h/4 \rceil$	$s_h - s_0$	**0.0624**	**0.0151**	**83.57**

are reported in millimeters. They complement, rather than reproduce, the pose-aware NDTW metric. For each task, we select the query with the largest absolute Raw NDTW reduction among those passing both models' visual integrity criteria. All images use the head camera at frame 32, with identical crops across methods showing the right hand in panels a–c and the left hand in panel d. The curves include the initial frame and 32 future frames.

These cases illustrate large improvements selected within each task, not average behavior or evidence for IE+AFE. Gripper, rotation, and opposite-arm errors remain visible, particularly in pot lifting and QR-code rotation. Reduced position error alone does not establish task completion. The aggregate evidence for the AFE design comes from Table 10.

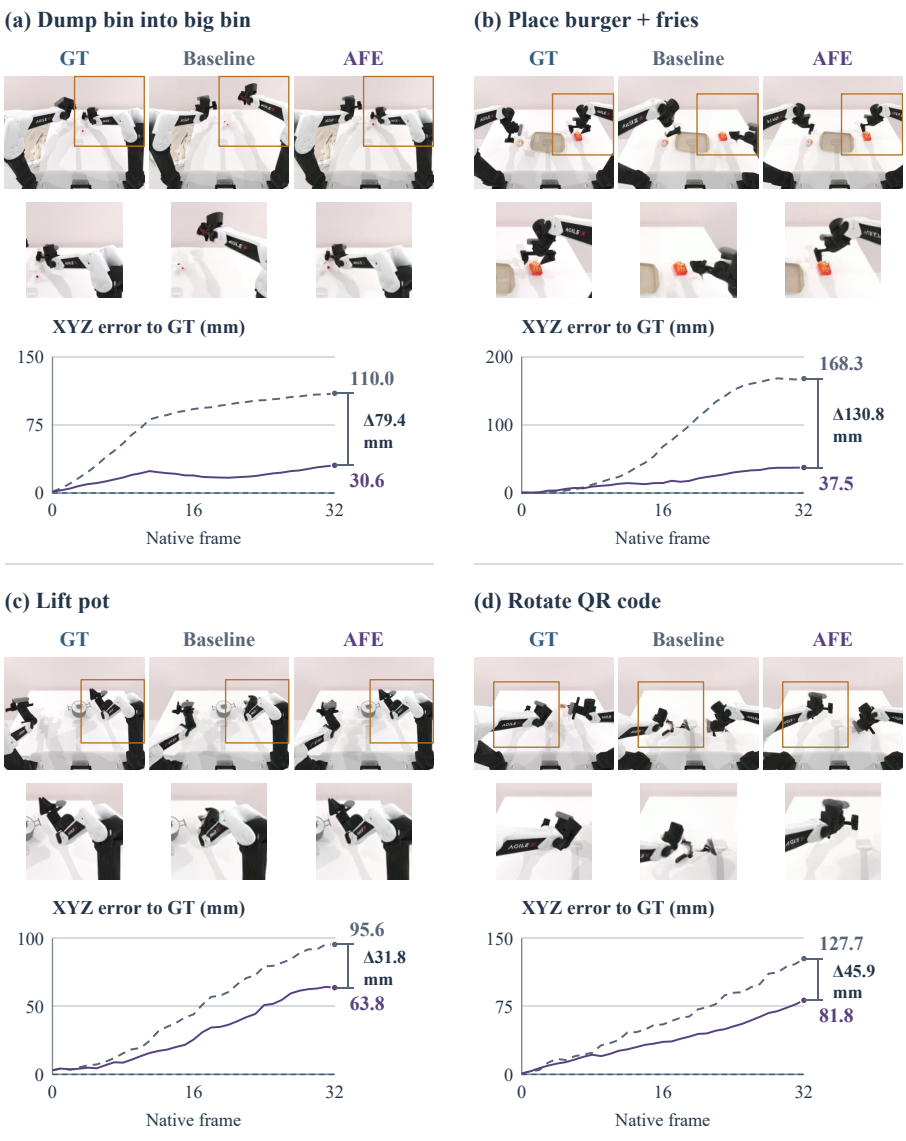

Figure 24: Selected Baseline and AFE video predictions at 5k training steps. Images show final frames and hand details; curves report position errors averaged over both arms in millimeters. Endpoint labels show final-frame errors, and brackets mark the Baseline–AFE differences. Cases illustrate improvements rather than average performance.

E.2 Data Composition Ablations

Setup. We examine how progressively broadening the training-data composition affects action following. We use WorldSync with IE and AFE disabled to isolate the role of data composition. Starting from expert demonstrations, we successively include feasible-space sampling, local-perturbation, cross-state-replay, and policy-rollout data. All configurations are evaluated at the 40k checkpoint on four RoboTwin tasks with three camera views and the shared end-effector action representation, following the WorldEcho evaluation protocol defined in the main text.

Table 11: Incremental data-composition ablation at 40k with IE and AFE disabled. Each row retains all data families in the preceding row and adds the named family. The final row includes all four off-expert families and constitutes Expanded Action Coverage. Bold denotes the best value in each metric column.

Training data	Gated error ↓	Raw NDTW ↓	Visual pass (%) ↑
Expert Demonstrations	0.0352	0.0314	**98.57**
+ Feasible-Space Sampling	0.0336	0.0282	97.86
+ Local Perturbation	0.0330	0.0270	97.86
+ Cross-State Replay	0.0300	0.0242	97.86
+ Policy Rollout	**0.0264**	**0.0204**	97.86

Results. Table 11 shows that the complete mixture reduces integrity-gated error from 0.0352 to 0.0264 and raw NDTW from 0.0314 to 0.0204 relative to expert-only training, corresponding to reductions of 25.0% and 35.0%, respectively. Visual pass rate decreases from 98.57% to 97.86%. Both errors decrease at every stage of the incremental comparison. Adding feasible-space data lowers both errors despite a decrease in visual pass rate; subsequent additions of local-perturbation and cross-state-replay data further reduce both errors while maintaining visual pass rate. The final addition of policy-rollout data yields the largest successive reduction in both errors while maintaining visual pass rate at 97.86%.

These results support broadening training beyond expert demonstrations under the evaluated configuration. The successive gains are conditional on the data families already present: they characterize this incremental composition path rather than establish an independent contribution ranking among families.

F Additional Quantitative Results

F.1 Results across Query Categories

Evaluation scope. We complement the overall comparison in Table 1 with results for each of the five action-query categories under the same evaluation protocol. Figure 25 reports all model configurations, while Figures 26 and 27 visualize their category-level profiles. Raw NDTW and gated error retain task-macro aggregation, and visual pass rate counts accepted rollouts.

Expanded action coverage. Expanded Action Coverage reduces both raw NDTW and gated error on Policy Rollout and Feasible-Space Sampling for all six baselines (Figures 25 and 26). These gains extend across actions produced during policy execution and actions sampled from the broader feasible space, providing category-level support for the overall benefits of expanded training coverage.

WorldSync across categories. Both WorldSync variants achieve the two lowest gated-error point estimates on Demonstrated Action, Cross-State Replay, and Feasible-Space Sampling (Figure 25). With additional robotic pretraining, the errors are 0.0058, 0.0103, and 0.0213, respectively. These correspond to reductions of 50.6%, 34.7%, and 22.3% relative to expanded-coverage DreamDojo, the lowest-scoring baseline in each category. The gains therefore span demonstrated actions, replay from a different initial state, and sampling from the broader feasible action space.

Trajectory agreement and visual integrity. The raw NDTW and visual pass results further characterize these advantages (Figures 25 and 27). On Demonstrated Action, the two WorldSync variants achieve the lowest two raw errors and both pass every visual check on all evaluated rollouts. On Feasible-Space Sampling, the pretrained variant combines the lowest gated error with the highest visual pass rate of 99.6%. These results support the gated-error improvements with specific evidence of trajectory agreement and visual integrity in the corresponding categories.

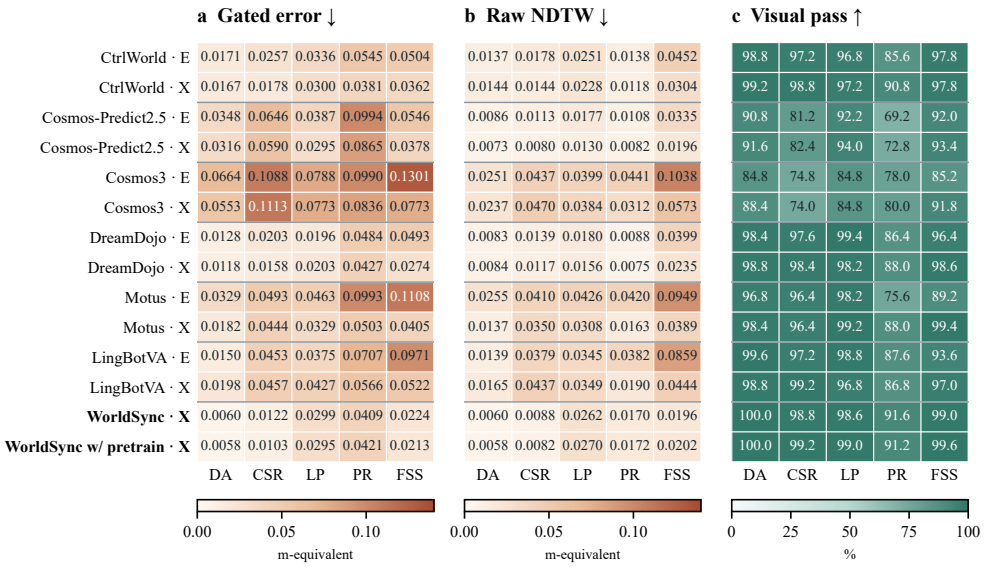

Figure 25: Results across query categories. Panels show gated error, raw NDTW, and visual pass rate. E/X denote Expert Demonstrations/Expanded Action Coverage. DA, CSR, LP, PR, and FSS denote Demonstrated Action, Cross-State Replay, Local Perturbation, Policy Rollout, and Feasible-Space Sampling. Cells give point estimates, and both error panels share an absolute color scale.

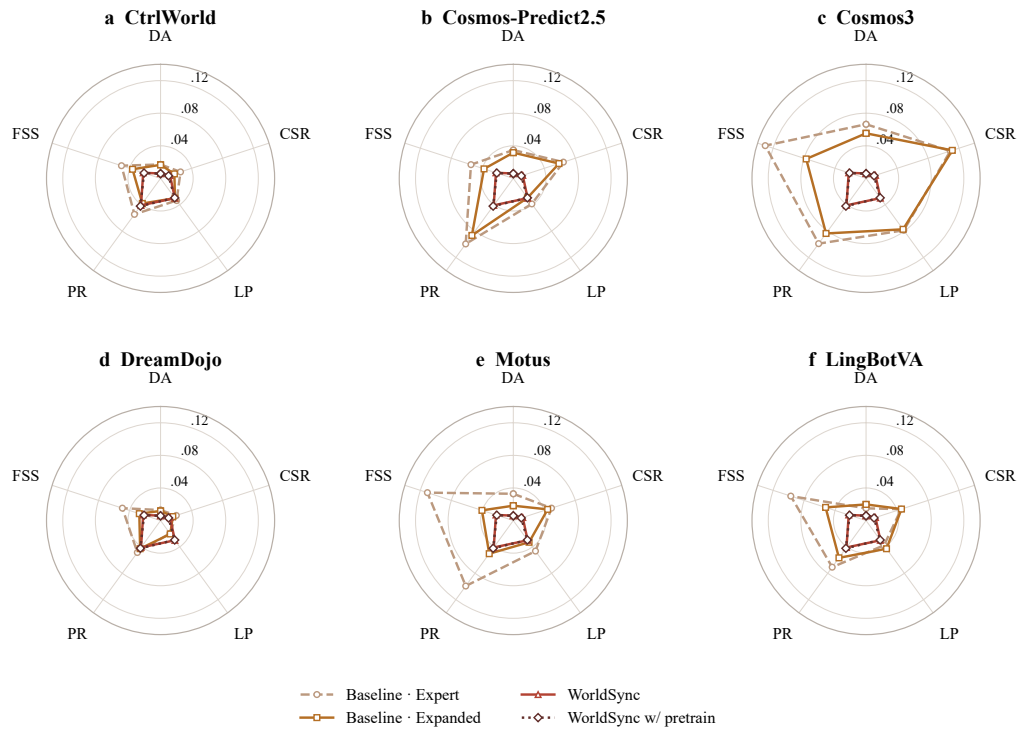

Figure 26: Gated error by query category. Each panel compares one baseline under both training regimes with the two WorldSync variants, repeated as common references. All axes share a 0–0.14 m-equivalent scale, with lower error closer to the center. Category abbreviations follow Figure 25.

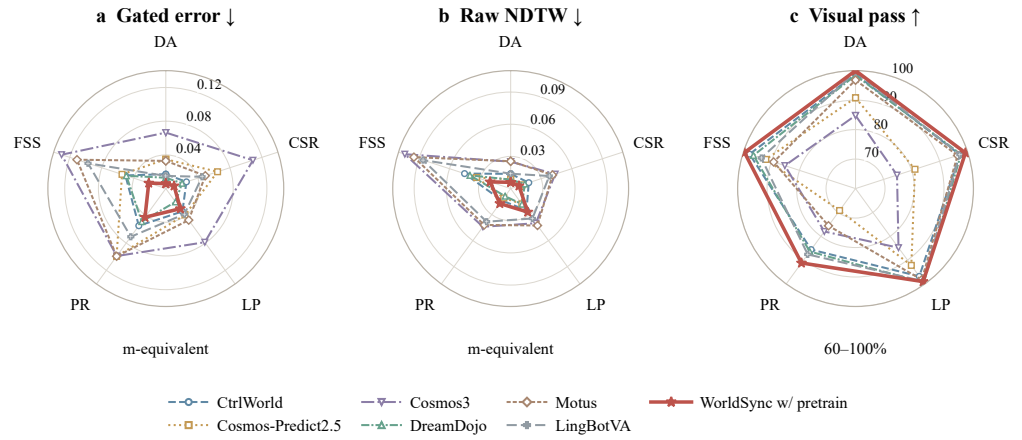

Figure 27: Three-metric profiles of expert-trained baselines and WorldSync with additional robotic pretraining and Expanded Action Coverage. Error axes start at zero, while visual pass spans 60–100%. Lower error lies closer to the center and higher pass rate farther out. Abbreviations follow Figure 25.

F.2 Metric Distributions and Analysis

To support policy evaluation, world-model metrics must distinguish differences in visual integrity and action following. Blurred imagery, distorted robot structures, and motion inconsistent with the commanded actions can each make generated rollouts unreliable for policy learning. We therefore examine whether WorldEcho scores and gate decisions correspond to observable differences in generated videos. We combine quantitative score distributions and per-check pass/fail rates with qualitative examples across score ranges to connect the metric values to visual integrity and action following.

F.2.1 Metric Distributions

Figure 28 shows metric distributions for over 10k generated clips from early training checkpoints, spanning 50 tasks and all five action-query categories. Each category includes over 200 distinct action queries. We retain all evaluated clips when plotting distributions, including visually rejected clips. The image-quality and motion-smoothness thresholds are $\tau_q = 0.33$ and $\tau_m = 0.45$. To visualize the binary EEF visibility check, we normalize the detection counts from tracking and text-prompted detection by their respective required counts and plot the larger value; a value of at least 1 passes the check. Arm integrity is binary, while raw NDTW is continuous and has no pass/fail threshold.

EEF visibility has the highest rejection rate in this evaluation set (Figure 28(f)): it rejects 7.3% of clips, compared with 0.9% for image quality, 0.4% for motion smoothness, and 3.5% for arm integrity. Reporting these outcomes separately reveals the frequency of violations in each visual dimension alongside the overall pass/fail decision. Each check is counted separately, so a clip may fail multiple checks. Clips with no dynamic frame pairs pass the motion check and are omitted from its score histogram.

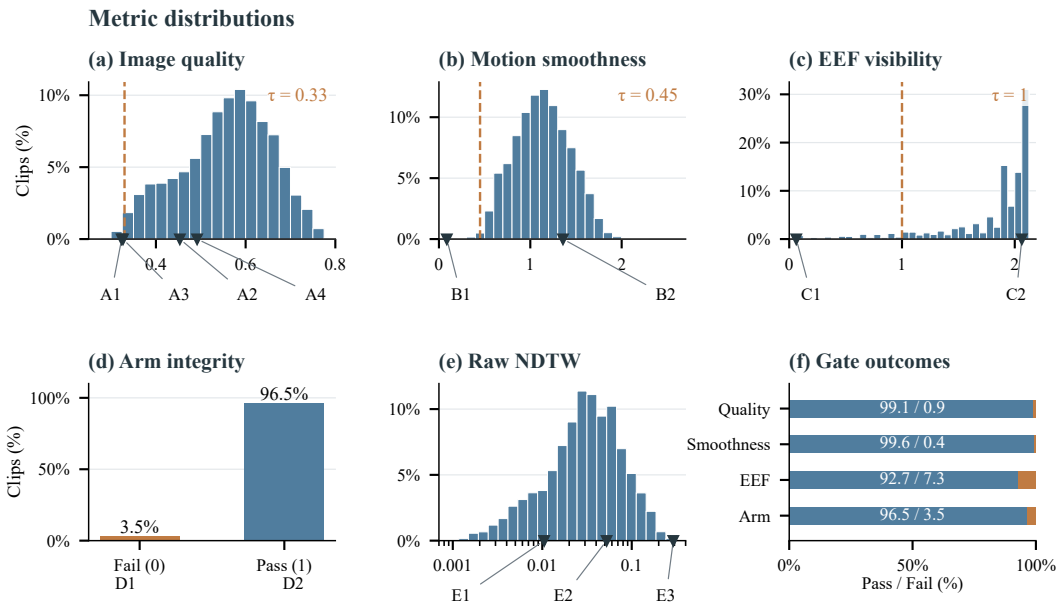

Figure 28: Metric distributions of generated clips. Panels (a–e) show the five metrics; case labels correspond to Figure 29, and dashed lines mark thresholds. Panel (f) reports each check's Pass/Fail percentages (blue/orange), with overlapping failures allowed. Percentages use the full evaluation set, including clips omitted from motion scoring.

F.2.2 METRIC ANALYSIS

Figure 29 connects the metric values to appearance, robot structure, and motion agreement. For image quality, each pair (A1/A2 and A3/A4) uses the same action query and native resolution. The lower-scoring examples show wrist blur (A1, 0.3236; A3, 0.3266), whereas A2 (0.4533) and A4 (0.4918) are clearer and exceed 0.33.

Following WorldArena (Shang et al., 2026a), motion smoothness is assessed through frame-interpolation reconstruction, weighted by motion magnitude. B1 and B2 receive sequence-level scores of 0.0864 and 1.3576, respectively. The displayed start and end frames provide visual context for scores computed over the full sequences.

For EEF visibility, C1 shows the right EEF out of view, while C2 shows both EEFs; their normalized detection counts are 0.062 and 2.062. These examples use different queries. For arm integrity, D1 and D2 share the same query: D1 shows arm distortion and fails the check, whereas D2 preserves arm structure and passes.

For action following, E1–E3 share one query and have raw NDTW errors of 0.0105, 0.0524, and 0.2913. The estimated position paths show increasing mismatch with trajectories estimated from ground-truth videos. The position paths visualize the spatial component of the pose discrepancy, while raw NDTW jointly measures position and rotation agreement.

Together, these examples connect the WorldEcho scores to appearance, robot visibility and structure, and action following, providing an interpretable diagnosis of generated rollout quality.

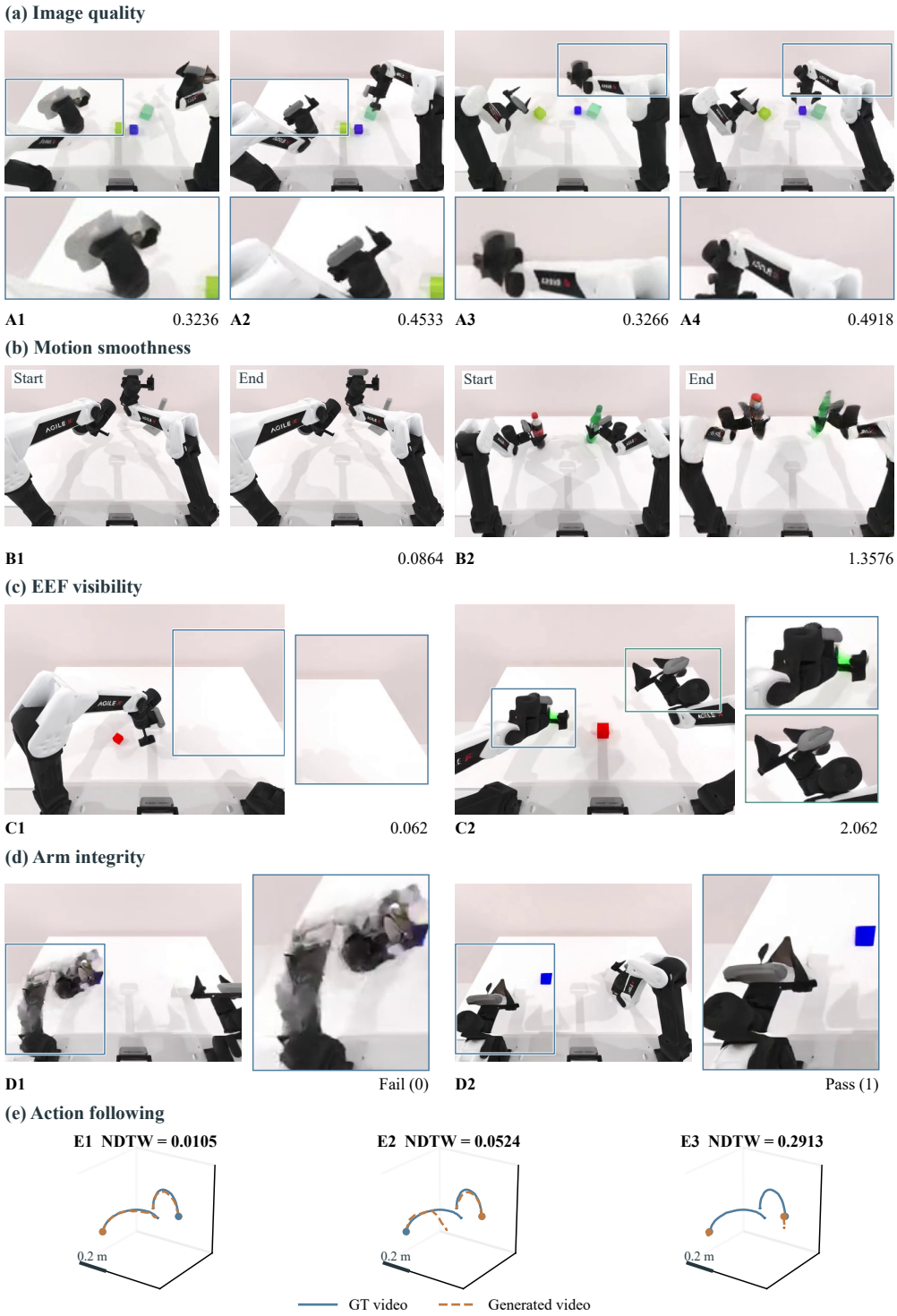

Figure 29: Examples corresponding to the case labels in Figure 28. Scores are computed on full videos. Colored boxes locate the enlarged regions; C2 enlarges the left and right EEFs above and below, respectively. In (e), solid blue and dashed orange paths are estimated from ground-truth (GT) and generated videos on shared 3D axes; dots mark starts.

F.3 Agreement with Human Evaluation

We examined how WorldEcho's automatic measurements correspond to human judgments, complementing the metric distributions and examples in Appendix F.2. We first assess whether raw NDTW captures human judgments of action following, then examine the four visual checks used for preliminary screening.

Evaluation protocol. Eleven raters provided 544 ratings for 100 clips, with five or six ratings per clip. Forty action clips received ratings on a five-point scale, with higher scores indicating greater action agreement. The remaining 60 clips received binary pass/fail judgments, with 15 clips each for image quality, motion smoothness, EEF visibility, and arm integrity. We retained all observed ratings, using the median for each action clip and a strict majority for each visual clip. No visual clip had a tied vote.

Sampling and analysis. The action set covers four tasks and five action-query categories, with two clips per combination. The tasks are `dump_bin_bigbin`, `lift_pot`, `place_burger_fries`, and `rotate_qrcode`. For action following, we computed Spearman's correlation between raw NDTW and median human ratings. A 95% percentile interval was obtained from 10,000 bootstrap resamples of clips within each task, retaining each clip's observed ratings. For visual metrics, we compared the recorded automatic decisions with human majority labels using accuracy, pass recall, failure recall, and Cohen's κ. Krippendorff's α measures inter-rater agreement, using ordinal distances for action scores and nominal distances for visual labels.

Action-following metric. Raw NDTW showed a strong negative association with median human action ratings (Figure 30(a)). Spearman's ρ was -0.826, with a 95% CI of $[-0.911, -0.679]$, indicating that lower trajectory errors corresponded to higher perceived action agreement. This result supports trajectory alignment as the central measure of action following in WorldEcho across the sampled tasks and action-query categories. Ordinal Krippendorff's α was 0.594, so the correlation reflects agreement with aggregated human judgments despite variation among individual raters.

Visual metrics. The visual checks are intended to screen for basic visual integrity while preserving clips for action-following evaluation. Automatic decisions matched human majority labels on 80.0%, 80.0%, 93.3%, and 86.7% of image-quality, motion-smoothness, EEF-visibility, and arm-integrity clips, respectively (Figure 30(b)). Each check retained 90.0%–100.0% of human-passed clips, while rejecting 50.0%–80.0% of human-failed clips (Table 12). This asymmetry is consistent with the intended permissive screening, which limits visual rejection so that retained clips can be evaluated for action following. For clips passing all four checks, raw NDTW determines the official error, keeping trajectory agreement central to the evaluation. Some human-rated visual failures nevertheless passed the automatic checks, so this screening does not guarantee visual integrity. These per-check results describe 15 sampled clips per dimension.

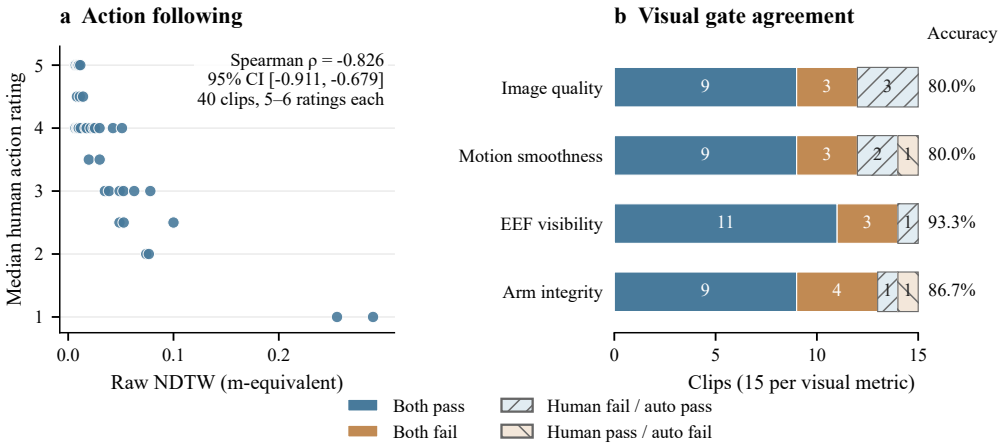

Figure 30: Automatic measurements and human judgments. (a) Raw NDTW versus median human ratings for 40 clips with five or six ratings each. Points retain exact coordinates, and marker area scales with the number of coincident clips. The interval is a task-stratified bootstrap interval for Spearman's ρ. (b) Joint human and automatic pass/fail decisions for 15 clips per visual metric. Human labels use a strict majority of five or six ratings. Solid segments indicate agreement, and hatched segments indicate disagreement. Segment labels give clip counts, and percentages give total agreement.

Table 12: Visual classification and inter-rater agreement. Each metric uses 15 clips with five or six ratings per clip. Pass recall is the fraction of human-passed clips accepted automatically. Failure recall is the fraction of human-failed clips rejected automatically. Cohen's κ compares automatic decisions with majority labels, while nominal Krippendorff's α compares individual raters.

Visual metric	Accuracy (%)	Pass recall (%)	Failure recall (%)	Cohen's κ	Rater α
Image quality	80.0	100.0	50.0	0.545	0.815
Motion smoothness	80.0	90.0	60.0	0.526	0.605
EEF visibility	93.3	100.0	75.0	0.815	0.651
Arm integrity	86.7	90.0	80.0	0.700	0.888

G Additional Qualitative Results

Figures 31 and 32 compare WorldSync with six baseline models on simulation tasks from WorldEcho. For each comparison, both models receive the same initial observation and action sequence, with the simulator replay providing ground truth. Rows show ground truth, WorldSync, and the corresponding baseline at matched frame indices. The final column enlarges the marked region using the same relative image coordinates across rows, highlighting local pose and shape differences. Figure 33 reports the corresponding WorldEcho metrics, computed on the complete generated videos rather than the displayed frames or enlarged regions.

Figure 31 shows differences in arm structure and gripper motion. In *Move playing card*, CtrlWorld produces distorted arm and gripper shapes. WorldSync better preserves their structure and follows the downward movement in the ground-truth replay. In *Move pill bottle*, Cosmos-Predict2.5 largely retains the initial right-gripper rotation, whereas WorldSync captures part of the wrist rotation. In *Place dual shoes*, Cosmos3 moves the grippers toward the shoe box. WorldSync keeps them to its sides, more closely matching the ground-truth replay.

Figure 32 shows differences in wrist motion and arm placement. In *Stack blocks (three)*, DreamDojo largely preserves the initial left-wrist pose. WorldSync follows the direction of the wrist motion, although its amplitude remains smaller than in the ground-truth replay. In *Grab roller*, Motus retracts the extended arm, while WorldSync more closely maintains its ground-truth placement. In *Place bread basket*, LingBotVA moves the grippers toward the basket, whereas WorldSync better preserves their placement relative to ground truth.

The per-case metrics in Figure 33 distinguish visual validity from motion agreement. WorldSync passes all four visual checks in all six cases and has lower raw NDTW in each comparison. CtrlWorld, Motus, and LingBotVA fail the end-effector visibility check; Motus also fails arm integrity. Their integrity-gated errors therefore take the fixed value of 0.3. Cosmos-Predict2.5, Cosmos3, and DreamDojo pass all visual checks, but their raw NDTW errors remain higher than WorldSync's (0.0714 versus 0.0251, 0.2532 versus 0.0256, and 0.0604 versus 0.0224, respectively). These cases show that passing the visual gate does not ensure agreement with the conditioned motion.

The individual scores also expose differences that the overall gate does not rank. For example, DreamDojo has higher image quality than WorldSync (0.679 versus 0.484), despite its larger trajectory error. Likewise, a passing arm-integrity decision does not rule out local shape distortions, as the CtrlWorld example illustrates. The visualizations and per-case metrics therefore provide complementary evidence.

These selected examples illustrate closer agreement with ground-truth motion, alongside remaining differences in wrist rotation and object appearance. They were chosen after assessing trajectory error, visual integrity, and the complete generated sequences. The comparisons illustrate specific visual discrepancies; aggregate performance over the full evaluation set is reported in Table 1.



Figure 31: Qualitative comparisons of WorldSync with CtrlWorld, Cosmos-Predict2.5, and Cosmos3. Rows show ground truth, WorldSync, and the baseline for each task. The last column enlarges the marked final-frame region.

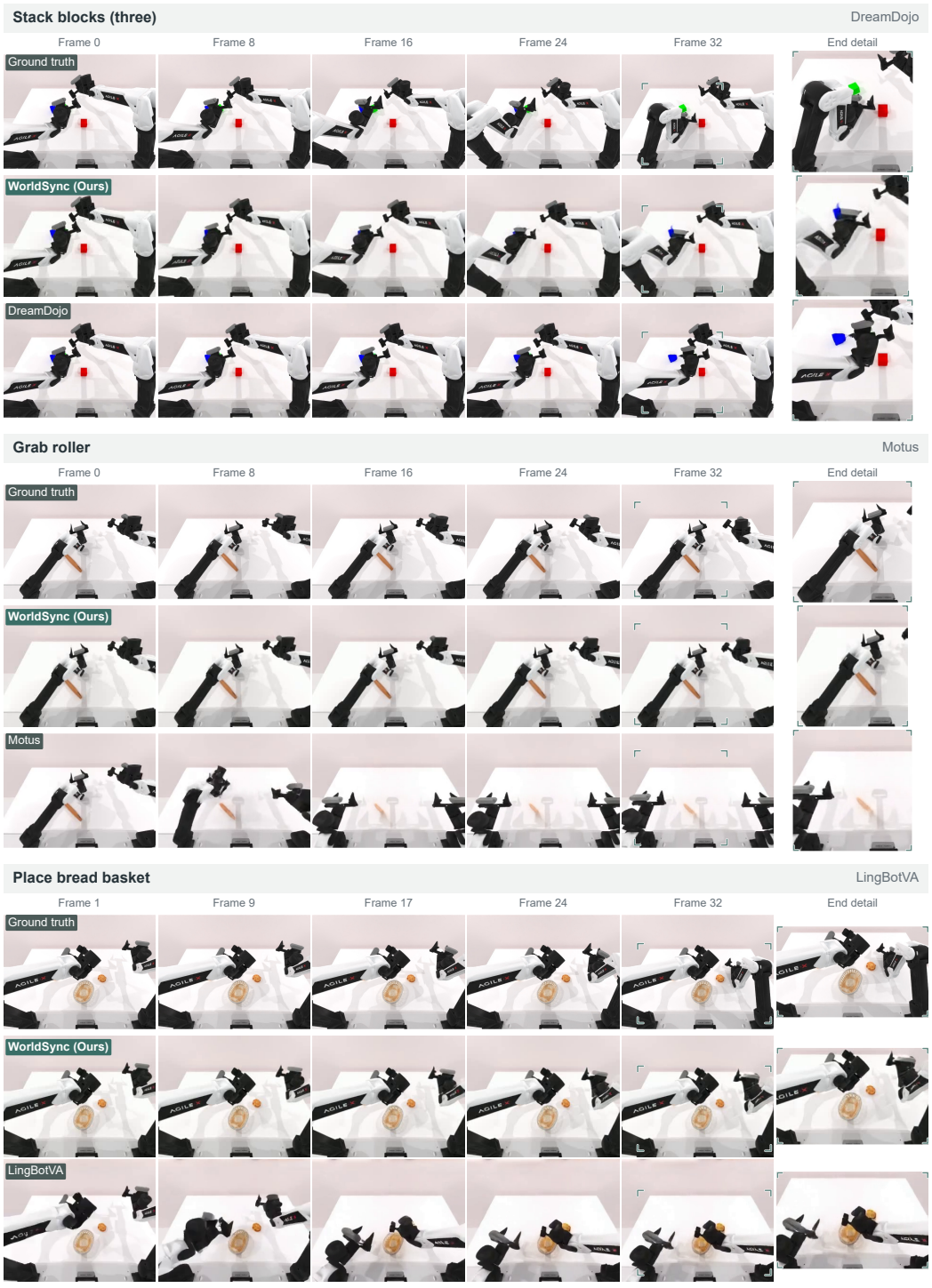

Figure 32: Qualitative comparisons of WorldSync with DreamDojo, Motus, and LingBotVA. Rows show ground truth, WorldSync, and the baseline for each task. The last column enlarges the marked final-frame region.

a Visual gates and trajectory errors

✓ Pass ✗ Fail

Model	Image quality ↑	Motion smoothness ↑	EEF visibility ↑	Arm integrity	Overall gate	Raw NDTW ↓	Gated error ↓
1 Move playing card							
CtrlWorld	0.472 ✓	1.009 ✓	0.235 ✗	1 ✓	✗	0.0990	0.3000
WorldSync	0.456 ✓	1.103 ✓	1.000 ✓	1 ✓	✓	**0.0222**	**0.0222**
2 Move pill bottle							
Cosmos-Predict2.5	0.627 ✓	0.860 ✓	0.970 ✓	1 ✓	✓	0.0714	0.0714
WorldSync	0.591 ✓	1.110 ✓	1.000 ✓	1 ✓	✓	**0.0251**	**0.0251**
3 Place dual shoes							
Cosmos3	0.393 ✓	1.429 ✓	0.529 ✓	1 ✓	✓	0.2532	0.2532
WorldSync	0.705 ✓	1.257 ✓	0.848 ✓	1 ✓	✓	**0.0256**	**0.0256**
4 Stack blocks (three)							
DreamDojo	0.679 ✓	0.944 ✓	1.000 ✓	1 ✓	✓	0.0604	0.0604
WorldSync	0.484 ✓	1.190 ✓	1.000 ✓	1 ✓	✓	**0.0224**	**0.0224**
5 Grab roller							
Motus	0.501 ✓	1.573 ✓	0.000 ✗	0 ✗	✗	0.2550	0.3000
WorldSync	0.572 ✓	0.779 ✓	1.000 ✓	1 ✓	✓	**0.0121**	**0.0121**
6 Place bread basket							
LingBotVA	0.635 ✓	1.592 ✓	0.250 ✗	1 ✓	✗	0.2051	0.3000
WorldSync	0.572 ✓	0.754 ✓	1.000 ✓	1 ✓	✓	**0.0216**	**0.0216**

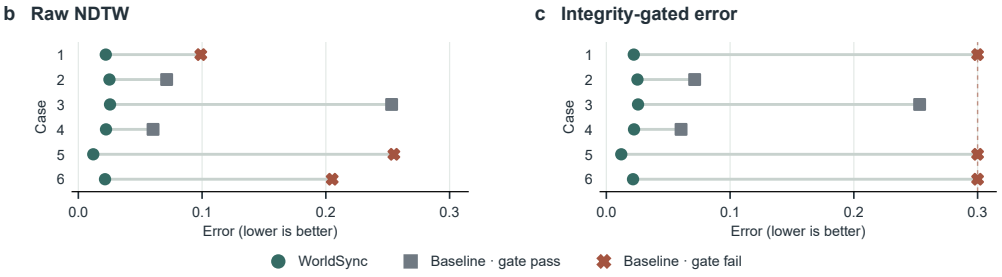

Figure 33: WorldEcho metrics for the six cases in Figures 31 and 32. (a) Scores and gate decisions. (b,c) Raw NDTW and integrity-gated error. Image quality and motion smoothness pass at ≥ 0.33 and ≥ 0.45, respectively. EEF visibility passes when either tracking or text-prompted detection meets its requirement; arm integrity must equal 1. The displayed EEF fraction comes from tracking when that branch passes and from text detection otherwise; the fractions do not share a universal 0.5 cutoff. The overall gate requires all four checks; gated error equals raw NDTW when passed and 0.3 otherwise.