DEXWORLD-BENCH: EVALUATING FINE-GRAINED CONTROL AND INTERACTION FOR DEXTEROUS WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World models for dexterous manipulation have served as learned simulators for anticipating action consequences and supporting robotic planning and policy learning. Their utility depends on faithfully capturing fine-grained finger motions and hand-object interactions, where small control differences can alter manipulation outcomes. However, existing evaluation methods provide limited systematic coverage of these dynamics. Similar overall motion or successful task completion can conceal errors in local control and interaction. We introduce **DexWorld-Bench**, a benchmark that evaluates dexterous world models along three complementary dimensions: **fine-grained control accuracy**, **interaction consistency**, and **task executability**. The benchmark comprises 220 full task sequences, 90 critical interaction clips that include both successful and failed interactions, and 70 controlled action comparisons that isolate the motion of a single arm or fingers. We evaluate multiple widely-used world models spanning general-purpose video generators, action-conditioned world models, and world action models under the same protocol. The results show that action conditioning alone does not ensure that models faithfully follow fine-grained action commands, and that fine-grained control and long-horizon executability remain key bottlenecks in current dexterous world models.

1 INTRODUCTION

World models for dexterous manipulation have recently attracted growing attention as learned simulators to anticipate the consequences of dexterous robot actions. Conditioned on robot actions, hand motions, and task instructions, these models predict future visual observations or environment states to support planning, policy evaluation, and policy improvement. Prior work has explored dexterous manipulation planning driven by future-state prediction (Goswami et al., 2025), video simulation with fine-grained action controllability (Feingold et al., 2026), and closed-loop policy learning that couples state prediction with outcome evaluation (Bi et al., 2026). Compared with parallel-jaw manipulation, dexterous manipulation involves higher-dimensional finger control and more complex, contact-rich hand-object interaction, where small deviations in individual finger motion can alter contact states and manipulation outcomes.

Recent evaluation of embodied world models has undergone a parallel expansion, moving beyond visual quality and temporal consistency to also assess physical executability, interaction correctness, and downstream functional utility (Shang et al., 2026; Jiang et al., 2026; Liu et al., 2026; Rong et al., 2026). However, benchmarks that address dexterous hands remain scarce, and even these evaluate primarily the consequences of interactions rather than the fidelity of the generated hand itself. In particular, no existing protocol evaluates responsiveness to sub-dimensional action inputs, such as commands that actuate only a single finger or a designated arm.

To this end, we introduce **DexWorld-Bench**, a benchmark for evaluating world models for robotic dexterous manipulation. Built on 11 tasks from DexJoCo (Wang et al., 2026), the benchmark comprises three complementary evaluation sets, namely full task sequences, critical interaction clips, and controlled action comparisons, spanning grasping, insertion, tool use and bimanual manipulation.

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

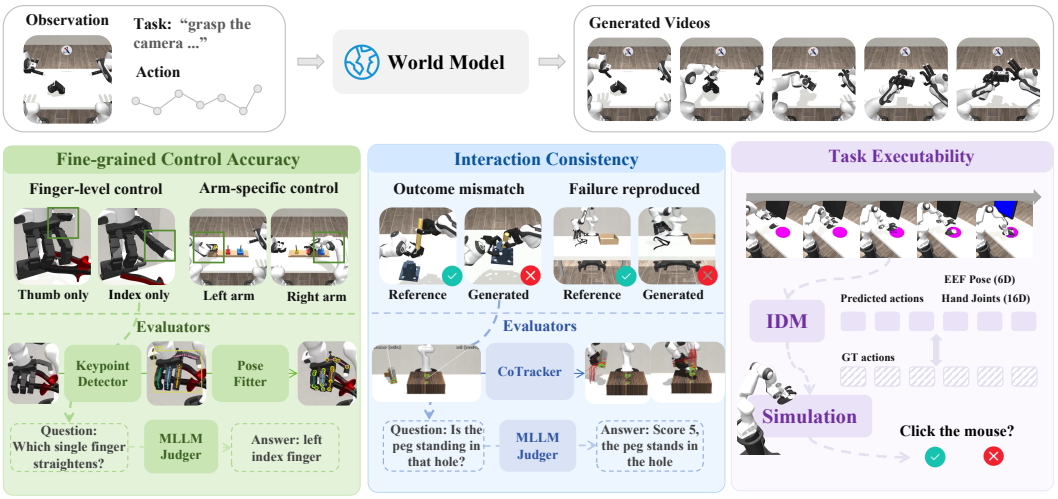

Figure 1: **Overview of DexWorld-Bench.** Conditioned on the initial observation, task instruction, and robot actions, a dexterous world model generates future videos. DexWorld-Bench evaluates these videos along three dimensions. **Fine-grained control accuracy** probes whether commanded motions appear only in the intended finger or arm. **Interaction consistency** checks whether object trajectories and interaction outcomes, including failures, match the reference. **Task executability** replays actions recovered by an inverse dynamics model in simulation to test whether the task is completed.

These sets provide task-level, event-level, and intervention-level evidence for evaluating models along three dimensions.

First, **fine-grained control accuracy** tests whether the commanded motion appears in the intended effector, and whether the generated hand remains geometrically valid, covering hand geometry accuracy, acting-arm selection, and finger-level responsiveness. Second, **interaction consistency** measures whether generated videos reproduce the reference object motion and the resulting interaction outcomes, covering object trajectory fidelity, trajectory consistency horizon, and outcome consistency. Finally, **task executability** recovers action sequences from generated videos and replays them open-loop in simulation, covering action-recovery fidelity and task completion.

We evaluate five representative world models on DexWorld-Bench. Action conditioning improves interaction prediction over the general-purpose video baseline, raising the object trajectory score from 0.520 to up to 0.878. However, it does not guarantee fine-grained command following. On arm-side control, IRASim and Ctrl-World reach 0.98, whereas FastWAM and Cosmos3 score 0.50 and 0.48, which is the level expected from a model that ignores the command. On finger-level control, no model exceeds 0.30. The best execution success rate is only 0.650, and visual quality metrics correlate only weakly with interaction outcomes.

We summarize our main contributions below.

- We introduce **DexWorld-Bench**, a benchmark dedicated to world models of robotic dexterous manipulation. Designed around the high-dimensional and intrinsically complex dynamics of dexterous hands, it evaluates fine-grained control accuracy, interaction consistency, and task executability.
- We construct **three complementary evaluation sets** comprising 220 full task sequences, 90 critical interaction clips, and 70 controlled action comparisons. Covering grasping, insertion, tool use, and bimanual manipulation, they support task-level, event-level, and intervention-level analysis.
- We conduct a **systematic evaluation of widely-used world models**, spanning general-purpose video generators, action-conditioned world models, and world action models, under a unified protocol. These results reveal that current world models remain deficient in fine-grained control and long-horizon executability, and highlight promising directions for future work.

Table 1: **Comparison of embodied world-model benchmarks.** ✓ denotes a dedicated evaluation, △ denotes partial or indirect coverage, and × denotes no dedicated evaluation.

Benchmark	Dexterous-Hand Evaluation	Single-Finger Response	Arm-Side Response	Controlled Action Comparisons	Interaction Outcomes	Simulation Replay
WorldArena (Shang et al., 2026)	×	×	×	×	△	×
WMBench (Team et al., 2026)	×	×	×	×	△	×
RoboWM-Bench (Jiang et al., 2026)	×	×	×	×	△	✓
KineBench (Liu et al., 2026)	×	×	×	×	×	✓
WorldSimProbe (Co et al., 2026)	×	×	×	✓	✓	×
H2R-Bench (Rong et al., 2026)	✓	×	×	×	✓	×
DexWorld-Bench (Ours)	✓	✓	✓	✓	✓	✓

2 Related Work

2.1 World Models for Embodied Intelligence

World models support embodied decision-making by predicting the dynamics of the environment. Latent-space approaches include DreamerV3 (Hafner et al., 2023), which optimizes policies through imagined trajectories, and V-JEPA 2 (Assran et al., 2025), which combines video representation pretraining with action-conditioned post-training for visually goal-conditioned robotic planning. Among video-generative approaches, UniSim (Yang et al., 2023) is conditioned on instructions or low-level controls. Ctrl-World (Guo et al., 2026) uses frame-wise action conditioning and multi-view prediction for policy evaluation and data generation, and DreamDojo (Gao et al., 2026) transfers interaction priors from large-scale human videos to robotic action spaces for evaluation and planning. Building on these advances, we focus on dexterous manipulation and examine whether world models faithfully predict the interaction outcomes induced by fine-grained control inputs.

2.2 World Models for Dexterous Manipulation

Dexterous manipulation requires jointly modeling arm motion, finger configurations, and hand–object contact, as even subtle finger movements can determine interaction outcomes. DexWM (Goswami et al., 2025) represents actions using finger keypoints from human videos, predicts future latent states, and introduces hand-consistency constraints to improve finger-configuration modeling. DWM (Kim et al., 2026) generates hand-action-driven interaction videos conditioned on 3D scene renderings and egocentric hand motion. DexAC-WM (Yuan et al., 2026b) uses structured action conditioning to address scale disparities among components of high-dimensional actions and better capture subtle movements, while Mask2Real-WM (Feingold et al., 2026) decouples segmentation-mask prediction from appearance generation to learn robotic-hand action responses from simulation. Despite these advances, each method is evaluated under its own data and protocol, and there is still no shared benchmark that tests whether dexterous world models respond correctly to fine-grained hand commands and predict the resulting interaction outcomes.

2.3 Benchmarks for Embodied World Models

World-model evaluation extends beyond visual quality to action fidelity and functional utility. VBench (Huang et al., 2024; 2025) assesses perceptual quality, while VBench-2.0 (Zheng et al., 2025) and VideoPhy (Bansal et al., 2025) evaluate physical and commonsense plausibility. WorldArena (Shang et al., 2026) evaluates utility for data generation, policy evaluation, and planning. GigaWorld-1 (Team et al., 2026) introduces WMBench to examine whether policy evaluations conducted in learned worlds agree with real-robot outcomes. RoboWM-Bench (Jiang et al., 2026) and KineBench (Liu et al., 2026) assess executability and kinematic feasibility through simulated execution. More targeted evaluations include WorldSimProbe (Co et al., 2026), which probes action responsiveness, interaction grounding, and dynamics through controlled interventions. H2R-Bench (Rong et al., 2026) evaluates action events, functional contacts, and embodiment consistency in human-to-robot video translation. Table 1 summarizes the evaluation dimensions covered by these benchmarks. DexWorld-Bench brings them together for dexterous manipulation. It tests whether finger- and arm-specific commands produce the intended motions, whether object responses match the reference, and whether actions recovered from generated videos complete the task in simulation.

3 THE DEXWORLD-BENCH BENCHMARK

3.1 BENCHMARK DESIGN

DexWorld-Bench evaluates dexterous world models in terms of *fine-grained control accuracy*, *interaction consistency*, and *task executability*. It uses three evaluation sets, namely controlled action comparisons, critical interaction clips, and full task sequences, built from the original DexJoCo dataset (Wang et al., 2026) and from Cosmos 3 and Fast-WAM policy rollouts (Agarwal et al., 2026; Yuan et al., 2026a).

Controlled action comparisons. To test whether generated videos realize a command only in the targeted body part, we construct arm-level and finger-level comparisons in which all non-target parts are held still. For the arm level, we build 50 comparisons (25 left–right pairs) from five episodes of each of five bimanual tasks, namely Assembly, Hanoi, Microwave Cook, Photograph, and Unlock iPad, by action-space editing. The two clips of a pair share an identical prefix. After a short interpolated bridge, one replays the recorded right-arm segment and the other the left-arm segment, with the opposite arm frozen at its initial pose. The edited actions are re-rendered in the simulator, so that the clips of a pair differ only in the acting arm. For the finger level, starting from frames of Pinch Tongs and Photograph in which all fingers are clearly visible, we fully extend one finger at a time while keeping the arm and the remaining fingers stationary.

Critical interaction clips. To evaluate how faithfully world models predict critical hand-object interactions and their outcomes, we construct a short-clip evaluation set covering grasping, insertion, tool use and bimanual manipulation. We select successful samples from the original DexJoCo dataset and failed samples from policy rollouts. We then manually identify the start and end points of each clip, extracting approximately 1-14 seconds (median 3.6 s) of the critical interaction. Together, successful and failed samples test whether generated videos preserve the interaction dynamics and outcomes of their corresponding reference sequences. Faithfully reproducing a failed interaction therefore also demonstrates predictive fidelity.

Full task sequences. To evaluate behavior prediction and task executability throughout complete manipulation episodes, we select 10 training-set and 10 held-out sequences from each of 11 tasks, yielding two evaluation subsets of 110 sequences each. The held-out sequences comprise demonstration episodes 90-99 of each task, excluded from the episodes 0–89 used to fine-tune the evaluated models. They test generalization to new episodes of the same tasks. Complete task sequences preserve successive manipulation stages and their temporal context, providing task-level evidence of how local control and interaction errors affect subsequent behavior and task completion.

3.2 FINE-GRAINED CONTROL ACCURACY

Hand Geometry Accuracy. Hand geometry accuracy measures whether the dexterous hand in a generated video can be explained by a kinematically valid Allegro hand. A YOLOv8-pose detector (Jocher et al., 2023), fine-tuned on simulator-rendered frames, localizes 20 keypoints per hand. A pose fitter then optimizes the 16 joint angles and the palm pose so that the forward kinematics of the simulator's Allegro model (MJCF) reprojects onto the detected 2D keypoints. We report the hand detection rate, the reprojection residual of the fit, the fraction of observable joints saturated at their limits, and the self-collision depth. Cases with a detection rate below 0.8 are marked undecidable for the latter three, avoiding the survivorship bias of scoring only a few clear frames. For the benchmark score, each term is normalized by its ground-truth value (model/GT for the detection rate and GT/model for the three error terms, so that ground truth scores 1), and the four ratios are averaged with equal weight (details in Appendix A.1).

Arm-Side Control Accuracy. Arm-side control accuracy measures whether a generated video exhibits the intended motion only on the side specified by the instruction. The judgment is made by an MLLM judge (Zheng et al., 2023; Chen et al., 2024). To avoid leaking the answer, the judge observes only the video without being told the motion instruction. It separately determines on which side of the view the manipulation occurs, which of the robot's own hands performs it, and which disk size is moved. Each question is answered over multiple samples with a majority vote, and the votes are compared against the instructed target side to obtain the accuracy.

Finger Control Accuracy. Finger control accuracy measures whether a generated video produces the intended motion only in the instructed finger, while the remaining fingers and the arm stay still. An MLLM judge observes the video without being told that any finger will move, answers which finger extends with a joint hand–finger label, and the judgment is made by a majority vote over multiple samples. This joint label captures both the response of the target finger and the isolation of the remaining body parts. The judgment is compared against ground-truth labels recorded during construction to obtain the accuracy, requiring no manual annotation. Unperturbed original clips, for which no finger extension is commanded, serve as negative controls to test whether the judge fabricates non-existent motion.

3.3 Interaction Consistency

Object Trajectory Score. The object trajectory score (OTS) measures whether a generated video reproduces the reference object motion. We project 24 mesh vertices per object using the recorded simulator poses as reference trajectories and track them through the generated video with CoTracker3 (Karaev et al., 2025). Each object's median tracking error is linearly normalized between two anchors. The error on ground-truth videos maps to 1, and the error of a frozen video (for moving objects) or a drift of ten times the tracking noise floor (for stationary objects) maps to 0. A survival penalty prevents models from lowering the error by making objects disappear, and scores are averaged over objects, cases, and tasks (full definition in Appendix A.2).

Trajectory Consistency Horizon. The trajectory consistency horizon (TCH) measures how long object motion keeps following the reference. A moving object diverges once its per-frame error persistently exceeds three times the noise floor, and the case horizon is the fraction of the video before the earliest divergence. Since tracker noise causes even ground-truth videos to diverge, we normalize by the per-task ground-truth horizon, so that ground truth scores 1.

Interaction Outcome Consistency. Interaction outcome consistency measures how consistently the interaction outcome presented in a generated video matches that of the reference sequence. An MLLM judge rates the degree of interaction execution in the video on a 1–5 scale against task-specific anchors that describe observable interaction outcomes in the reference clips. The absolute difference between the scores of a generated video and its reference sequence gives the fidelity error of the sample, which is mapped to a per-sample score of $1 - |s_{\mathrm{gen}} - s_{\mathrm{ref}}|/4$, so that 1 indicates a matching outcome. Successful and failed samples are compared under the same criterion, and scores are averaged over all samples.

3.4 Task Executability

Execution Success Rate. Execution success rate measures the task-level executability of the action sequence implied by a generated video. Specifically, the generated video is divided into contiguous 16-step segments. For each segment, the first and last frames of the ego view are given to the Inverse Dynamics Model (IDM) (Jang et al., 2025), which infers the corresponding 16-step actions. The per-segment predictions are then concatenated into a complete action sequence. This sequence is then replayed open-loop in the MuJoCo (Todorov et al., 2012) simulator from the task's initial state, and task completion is determined by the task's predefined physical criteria. The metric reports the fraction of sequences whose replay succeeds, reflecting whether accumulated control and interaction errors still permit task completion. Because per-task IDMs are used, the task executability dimension, comprising this metric and the action execution score below, is evaluated on three tasks, namely click mouse, water plant, and pick bucket (Appendix A.3).

Action Execution Score. The action execution score compares the actions inferred by the IDM (Jang et al., 2025) from a generated video with the recorded reference actions, aligned step by step. Errors are computed separately for each action group. The end-effector position error is measured as the Euclidean RMSE, the arm rotation error as the geodesic RMSE on SO(3), and the finger error as the RMSE over the 16 joint targets. The three error terms are reported separately to distinguish whether deviations arise from coarse arm motion or from fine-grained finger control. For the benchmark score, each error term is normalized by the corresponding error of the IDM applied to ground-truth videos (the achievable floor, scored 1), and the three ratios are averaged equally.

Table 2: Main results on DexWorld-Bench across fine-grained control accuracy, interaction consistency, and task executability. ↑ indicate higher is better. Best results are in **bold**. N/A: Wan2.2-TI2V is not conditioned on robot actions, so the controlled action comparisons do not apply.

Model	Fine-Grained Control Accuracy			Interaction Consistency			Task Executability	
	Hand Geom. Accuracy ↑	Arm-Side Ctrl. Accuracy ↑	Finger Ctrl. Accuracy ↑	Object Traj. Score ↑	Traj. Cons. Horizon ↑	Outcome Consistency ↑	Execution Success Rate ↑	Action Exec. Score ↑
Wan2.2-TI2V (Wan et al., 2025)	0.297	N/A	N/A	0.520	0.455	0.638	0.000	0.164
IRASim (Zhu et al., 2025)	0.334	**0.98**	**0.30**	0.762	0.708	**0.737**	**0.650**	**0.508**
Ctrl-World (Guo et al., 2026)	0.437	**0.98**	0.10	0.686	0.659	0.716	0.633	0.442
FastWAM (Yuan et al., 2026a)	0.531	0.50	0.05	**0.878**	**0.719**	0.716	0.583	0.308
Cosmos3 (Agarwal et al., 2026)	**0.537**	0.48	0.10	0.683	0.631	0.674	0.383	0.257

3.5 Auxiliary Visual Metrics

Beyond the three core dimensions, five conventional visual quality metrics are included as auxiliary references, each computed frame by frame and averaged over the sequence. PSNR and SSIM (Wang et al., 2004) measure the pixel-level and structural similarity between generated and reference frames, and LPIPS (Zhang et al., 2018) measures the perceptual distance with AlexNet features. All three are computed after frame-by-frame alignment with the reference video, excluding the first frame that serves as the generation condition. The imaging quality score, based on the MUSIQ (Ke et al., 2021) model, provides a no-reference assessment of per-frame imaging quality. In addition, the cosine similarity of DINO (Caron et al., 2021) features to the first frame and to the previous frames measures the temporal consistency of video appearance, conflating long-term drift and short-term jitter into a single score.

4 Experiments

4.1 Experimental Setup

Datasets. All experiments are conducted on three evaluation sets of DexWorld-Bench (Section 3.1), which are constructed from 11 manipulation tasks designed on the DexJoCo platform (Wang et al., 2026). The full-task set contains 110 training-set and 110 held-out sequences and provides task-level evidence. The critical-interaction set contains 90 clips of 1–14 seconds covering grasping, insertion, tool use and bimanual manipulation, including both successful and failed samples. The controlled-comparison set contains 50 left-right arm comparisons and 20 single-finger comparisons, including negative controls without any commanded motion, and provides intervention-level evidence.

Baselines. We evaluate five representative world models, including a general-purpose video generator (Wan2.2-TI2V (Wan et al., 2025)), two action-conditioned world models (IRASim (Zhu et al., 2025) and Ctrl-World (Guo et al., 2026)), and two world action models (FastWAM (Yuan et al., 2026a) and Cosmos3 (Agarwal et al., 2026)). For a fair comparison, all baselines are fine-tuned on the same DexJoCo training data. Wan2.2-TI2V is conditioned only on the initial frame and the task instruction, without robot actions, and is therefore excluded from the arm-side and finger-level control evaluations. All methods use the same frame rate and number of camera viewpoints for evaluation.

4.2 Main Results

4.2.1 Overall Performance on DexWorld-Bench

Table 2 reveals two main findings. First, the action-conditioned world models show better interaction prediction than the general-purpose video baseline. In particular, IRASim achieves the best outcome consistency (0.737) and leads both task-executability metrics, reaching 0.650 in execution success and 0.508 in action execution. FastWAM performs best on the trajectory-based metrics, achieving an object trajectory score of 0.878 and a trajectory consistency horizon of 0.719. Second, action conditioning alone does not guarantee fine-grained action control. Only IRASim and Ctrl-World reliably follow arm-side commands, both reaching 0.98, whereas FastWAM and Cosmos3

remain near 0.5, reproducing the training-time motion pattern rather than responding to the command. Finger-level control is even harder, with all action-conditioned models scoring at or below 0.30. Moreover, the best execution success rate is only 0.650, indicating that local prediction errors accumulate over complete task sequences. These results identify fine-grained action response and long-horizon executability as the central bottlenecks of current dexterous world models.

4.2.2 Fine-Grained Control Accuracy

Results reveal limited geometric precision and weak action grounding, particularly for individual fingers.

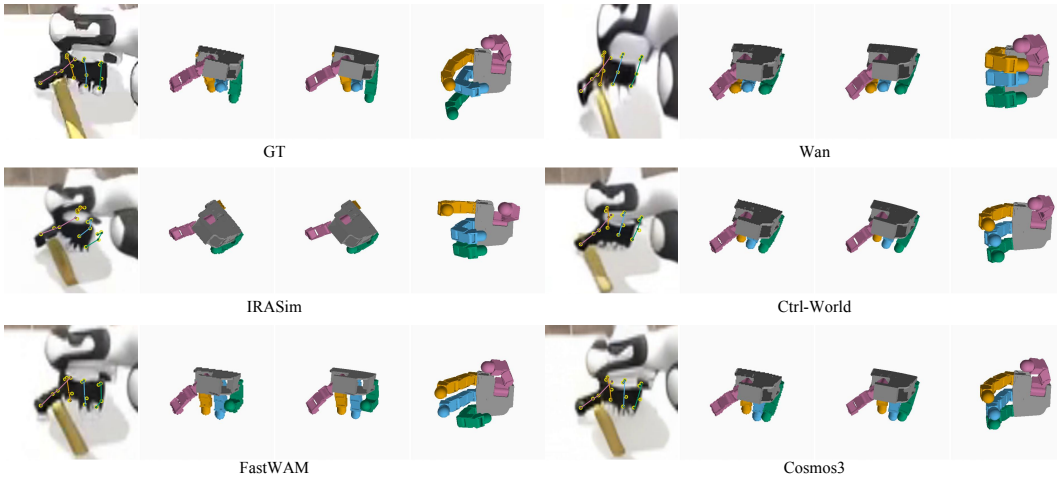

Figure 2: Qualitative comparison of hand geometry accuracy. For each model, the columns show the generated frame with detected 2D keypoints, a crop at the original simulator resolution, and re-renderings of the fitted hand from the original and an alternative camera view.

Hand Geometry Accuracy. As shown in Fig. 2, geometric distortions in the generated hands displace the detected keypoints, and the fitted poses, while preserving the coarse hand structure, deviate from the ground truth in finger flexion. Cosmos3 and FastWAM achieve the highest scores of 0.537 and 0.531, whereas Wan2.2-TI2V obtains only 0.297, suggesting that current world models capture coarse hand geometry but not fine-grained articulation.

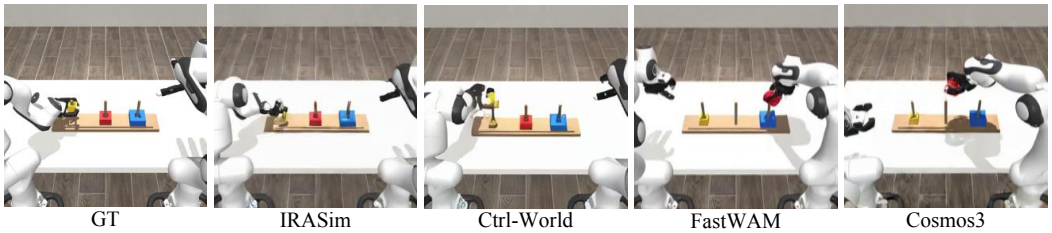

Figure 3: Qualitative comparison of the final frames generated by different models in the arm-side control accuracy evaluation, given an action command specifying left-arm motion. The first column shows the ground-truth frame.

Arm-Side Control Accuracy. Each task contains equal numbers of left- and right-arm commands, so a model that ignores the command and generates the same motion regardless of the input scores 0.5. In training, the two arms either move in a fixed order, as in Hanoi where the right arm always moves first, or move together, whereas each test clip moves only one arm. A score near 0.5 thus indicates that the model reproduces the training motion pattern rather than responding to the action. IRASim and Ctrl-World both reach 0.98. As shown in Fig. 3, they follow the left-arm command to grasp the yellow block while keeping the right arm still. FastWAM and Cosmos3 score 0.50 and

0.48 and, in the Hanoi example, move the right arm, which always moves first in training, instead of the commanded left arm.

Finger Control Accuracy. All action-conditioned models respond poorly to individual-finger commands. IRASim achieves the highest accuracy of only 0.30, while the others score at most 0.10. The generated hands mainly reproduce coordinated whole-hand motions from the demonstrations rather than independently moving the specified finger. This likely results from the subtle visual changes of individual fingers and the strong coupling of finger motions in the training data.

4.2.3 Interaction Consistency

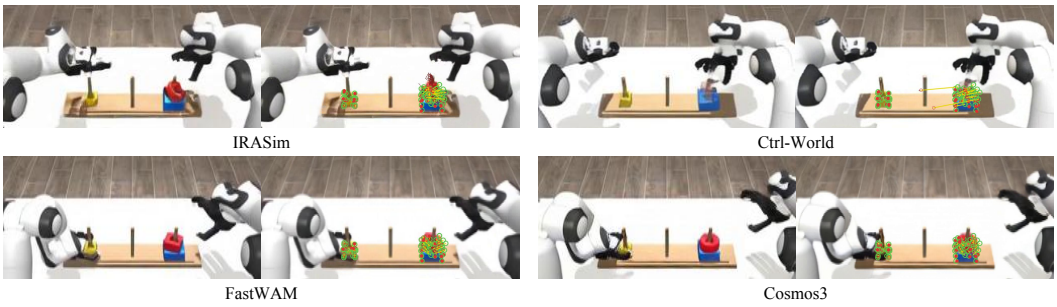

Figure 4: Qualitative visualization of object trajectories estimated by CoTracker (Karaev et al., 2025). For each model, the left panel shows a generated frame, while the right panel overlays the tracked object points and their accumulated trajectories.

Object Trajectory Consistency. We evaluate object motion using the Object Trajectory Score, which measures overall trajectory similarity, and the Trajectory Consistency Horizon, which measures how long this similarity is maintained. FastWAM performs best on both metrics, achieving 0.878 and 0.719, followed by IRASim with 0.762 and 0.708, respectively. As illustrated in Fig. 4, lower-performing models exhibit greater trajectory drift and mismatched object displacement.

Outcome Consistency. IRASim achieves the highest score of 0.737, followed by Ctrl-World and FastWAM tied at 0.716, while Cosmos3 (0.674) and Wan2.2-TI2V (0.638) trail. The different rankings between outcome consistency and trajectory metrics indicate that accurate local object motion does not necessarily preserve the final interaction outcome.

4.2.4 Task Executability

As shown in Table 2, IRASim performs best on both task-executability metrics, achieving 0.650 and 0.508, followed by Ctrl-World with 0.633 and 0.442. FastWAM obtains 0.583 and 0.308. Cosmos3 drops further to 0.383 and 0.257. Wan2.2-TI2V fails all tasks despite an action score of 0.164. These results suggest that errors in the generated visual dynamics propagate to the recovered actions and accumulate during execution, ultimately preventing successful task completion.

4.2.5 Visual quality and manipulation fidelity

Fig. 5 (left) reports auxiliary visual metrics, macro-averaged over the training-set and held-out test sets. FastWAM leads all three GT-paired metrics, achieving a PSNR of 21.30, an SSIM of 0.826, and an LPIPS of 0.200. Cosmos3 scores highest on imaging quality. Ctrl-World and FastWAM show the strongest subject consistency, with scores of 0.930 and 0.929, respectively, close to the real-video score of 0.927. Wan2.2-TI2V ranks last across all five metrics.

Fig. 5 (right) shows Spearman correlations between the auxiliary visual metrics and the manipulation metrics, where Motor averages hand geometry accuracy and action execution score, Traj. Fid. averages the object trajectory score and trajectory consistency horizon, and † marks task-level correlations. Imaging quality correlates strongly with hand geometry accuracy at 0.96, potentially because clearer hand images facilitate keypoint detection and pose fitting. However, it shows little association with interaction outcomes, execution success, or action accuracy. By contrast, PSNR

Model	Reference-Free		Paired with GT Frame		
	Imaging Quality ↑	Subject Consistency ↑	PSNR ↑	SSIM ↑	LPIPS ↓
Wan2.2-TI2V	0.426	0.881	13.98	0.662	0.442
IRASim	0.453	0.921	20.09	0.788	0.248
Ctrl-World	0.495	**0.930**	19.02	0.781	0.251
FastWAM	0.524	0.929	**21.30**	**0.826**	**0.200**
Cosmos3	**0.538**	0.918	18.45	0.781	0.239
Real video (reference)	0.533	0.927	–	–	–

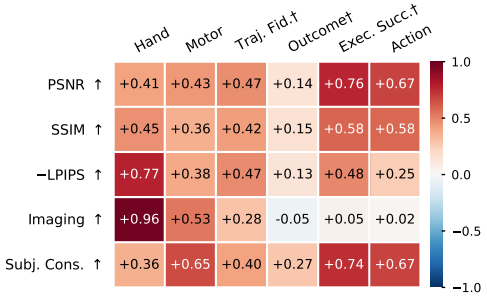

Figure 5: **Left:** auxiliary visual metrics of the five baselines. The best model in each column is in **bold**. **Right:** correlation between those visual metrics (rows) and manipulation metrics (columns), sign-aligned so that positive (red) always means better visual quality goes with better manipulation quality.

correlates with execution success and action accuracy at 0.76 and 0.67, while subject consistency shows similar correlations of 0.74 and 0.67. All five visual metrics have weak correlations with interaction outcomes, ranging from -0.05 to 0.27. These results suggest that visual metrics capture some aspects of generated hand appearance and motion, but provide limited evidence of whether the predicted interactions are correct.

4.3 Evaluator Reliability

4.3.1 IDM

We verify the IDM evaluator on real ground-truth videos. The recovered actions closely match the recorded ones, with an arm-position error of 1.05 cm, a geodesic arm-rotation error of 0.048 rad, and a finger-joint error of 0.020 rad. As shown in Table 3, the dedicated per-task checkpoints achieve a replay success of 0.87, where each task is evaluated over 10 episodes. The remaining gap mainly stems from three factors. First, the dexterous hand is often occluded, which makes hand-related dimensions hard to predict. Second, small per-step errors accumulate during open-loop replay without feedback correction. Third, the simulator itself introduces replay errors, and only 93.6% of ground-truth replays succeed.

4.3.2 MLLM Judge Validation

We validate the MLLM judges against ground-truth clips and human ratings. The arm-side judge correctly identifies all 50 reference clips. On the scored finger-control clips, the finger judge identifies the intended finger in 20 of 24 cases, achieves a vote agreement of 0.947, and produces no false alarms on eight static controls. For interaction outcomes, the judge separates successful from failed reference clips with an AUC of 0.906, compared with 0.964 for the human annotator. Its clip-level ratings also correlate with human ratings at a Spearman's ρ of 0.723. Further calibration results are reported in Tables 4 and 5.

5 Conclusion

We introduced DexWorld-Bench to evaluate whether dexterous world models follow fine-grained action commands, predict the resulting interactions, and produce behavior that can be executed in simulation. Its full task sequences, critical interaction clips, and controlled action comparisons provide complementary views of model performance. Across five widely-used models, interaction prediction improves with action conditioning, but finger-level responsiveness remains weak, and errors in generated dynamics limit task execution. Visual quality metrics alone also provide limited evidence of correct interaction outcomes. These findings identify fine-grained action grounding and sustained executable prediction as priorities for future dexterous world models.

AI USE STATEMENT

Generative AI is used in two distinct roles in this work. **As part of the benchmark methodology,** multimodal large language models serve as evaluators: GPT-based judges score finger control accuracy and interaction outcome consistency, and a locally deployed Qwen3-VL model judges arm-side control accuracy. These judges are calibrated against ground-truth labels and human ratings before use (Section 4.3.2 and Appendix A.5), and their scores are not treated as reliable by default. **As research and writing assistants,** we used generative AI tools to search for relevant literature, help implement evaluation and plotting scripts, check the consistency of reported numbers, and improve the readability and grammar of the manuscript. We did not use generative AI to design the benchmark dimensions, formulate the research questions, or produce experimental results. All AI-assisted code and text were reviewed and verified by the authors, who take full responsibility for the content of this paper.

REFERENCES

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

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

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

Hongzhe Bi, Zihao Zhou, Yihang Tang, Jingrui Pang, Shuhe Huang, Haitian Liu, Runqing Wang, Shuai Huang, Yichen Wang, Yiming Cheng, et al. Motus2: A self-evolving general world model for dexterous manipulation. *arXiv preprint arXiv:2608.30237*, 2026.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yinuo Liu, Yaochen Wang, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. Mllm-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark. *arXiv preprint arXiv:2402.04788*, 2024.

Peterson Co, Sicheng Hu, Chunxuan Jiao, Hongyang Cheng, Yulin Luo, Yijie Xu, Sixiang Chen, Zhongxia Zhao, Zihao Wang, DaFeng Chi, et al. Worldsimprobe: Diagnosing simulator faithfulness in action-conditioned world models for embodied manipulation. *arXiv preprint arXiv:2608.09298*, 2026.

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. *arXiv preprint arXiv:2607.04546*, 2026.

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

Raktim Gautam Goswami, Amir Bar, David Fan, Tsung-Yen Yang, Gaoyue Zhou, Prashanth Krishnamurthy, Michael Rabbat, Farshad Khorrami, and Yann LeCun. World models for learning dexterous hand-object interactions from human videos. *arXiv preprint arXiv:2512.13644*, 2025.

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

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

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, et al. Vbench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. Dreamgen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025.

Feng Jiang, Yang Chen, Kyle Xu, Yuchen Liu, Haifeng Wang, Zhenhao Shen, Jasper Lu, Shengze Huang, Yuanfei Wang, Chen Xie, et al. Robowm-bench: A benchmark for evaluating world models in robotic manipulation. *arXiv preprint arXiv:2604.19092*, 2026.

Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolo, 2023. URL `https://github.com/ultralytics/ultralytics`.

Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labeling real videos. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–10. IEEE, 2025.

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. IEEE, 2021.

Byungjun Kim, Taeksoo Kim, Junyoung Lee, and Hanbyul Joo. Dexterous world models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 29663–29673, 2026.

Zeyu Liu, Zhangzhe Zhu, Yang Zhang, Chenyou Fan, Chenjia Bai, and Xuelong Li. Kinebench: Benchmarking embodied world models via idm-free kinematic grounding. *arXiv preprint arXiv:2607.19876*, 2026.

Dingyi Rong, Yue Shi, Chaofan Ma, Jiezhang Cao, Zongrui Wang, Zeyu Zhang, Yao Mu, Guangtao Zhai, and Ning Liu. H2r-bench: Benchmarking human-to-robot manipulation video generation in world models. *arXiv preprint arXiv:2608.13049*, 2026.

Yu Shang, Zhuohang Li, Yiding Ma, Weikang Su, Xin Jin, Ziyou Wang, Lei Jin, Xin Zhang, Yinzhou Tang, Haisheng Su, et al. Worldarena: A unified benchmark for evaluating perception and functional utility of embodied world models. *AI Open*, 7:208–226, 2026.

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

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

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

Hanwen Wang, Weizhi Zhao, Xiangyu Wang, Siyuan Huang, He Lin, Boyuan Zheng, Rongtao Xu, Gang Wang, Yao Mu, He Wang, et al. Dexjoco: A benchmark and toolkit for task-oriented dexterous manipulation on mujoco. *arXiv preprint arXiv:2605.16257*, 2026.

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

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026a.

Zizhao Yuan, Zhengtu Liang, Taowen Wang, Qiwei Liang, Yichi Wang, Yunheng Wang, Yuetong Fang, Lusong Li, Zecui Zeng, and Renjing Xu. Not all actions are equal: Rethinking conditioning for dexterous world model. *arXiv preprint arXiv:2606.27325*, 2026b.

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

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Lulu Gu, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, et al. Vbench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

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

A Appendix

A.1 Hand Geometry Accuracy Details

Hand geometry accuracy measures whether the dexterous hand in a generated video can be explained by a kinematically valid Allegro hand. It requires no reference video, so applying the identical pipeline to ground-truth videos yields the measurement floor against which models are compared. For each video, we sample a contiguous window of $T{=}20$ frames starting at 30% of its length (ground-truth videos are strided to the models' 5 fps) and resize all frames to 640×640. A YOLOv8-pose detector (Jocher et al., 2023), fine-tuned on simulator-rendered frames with automatically projected labels, localizes 20 keypoints per hand (five link origins on each of the four fingers), yielding image coordinates $\mathbf{u}_{t,k}$ and confidences $w_{t,k}$, the latter set to zero below 0.3. A pose fitter then optimizes the 16 joint angles $\mathbf{q}_t$ and the palm pose $\mathbf{T}_t$ so that the forward kinematics FK of the simulator's Allegro model (MJCF), projected by the known camera intrinsics π, matches the detections:

$$\min_{\{\mathbf{q}_t,\mathbf{T}_t\}} \sum_{t=1}^{T} \frac{\sum_k w_{t,k}\, \rho_\delta\big(\pi(\mathrm{FK}_k(\mathbf{q}_t;\mathbf{T}_t)) - \mathbf{u}_{t,k}\big)}{\sum_k w_{t,k}} + \mathcal{R}_{\mathrm{temp}}, \qquad \mathbf{q}_t = \mathbf{q}^{\mathrm{lo}} + \big(\mathbf{q}^{\mathrm{hi}} - \mathbf{q}^{\mathrm{lo}}\big) \odot \sigma(\boldsymbol{\phi}_t), \tag{1}$$

where ρ_δ is the Huber loss and $\mathcal{R}_{\mathrm{temp}}$ penalizes changes between adjacent frames. Because the sigmoid parameterization keeps every joint within its limits by construction, an unreachable pose manifests as joints pressed against their limits rather than as a limit violation.

Four metrics are computed per case from the fit. Let $d_t^h \in \{0,1\}$ indicate whether hand h is detected in frame t. The *hand detection rate* is $D = \min_h \frac{1}{T}\sum_t d_t^h$, taking the worse hand in bimanual tasks. The first detected frame is used to select among candidate boxes and would otherwise bias the residual downward, so it is excluded from scoring. The other three metrics are evaluated on the scored frames $\mathcal{S}^h$, i.e., the remaining detected frames, and again take the worse hand:

$$E = \max_h \frac{1}{|\mathcal{S}^h|} \sum_{t\in\mathcal{S}^h} \frac{1}{H} \frac{\sum_k w_{t,k} \left\|\hat{\mathbf{u}}_{t,k} - \mathbf{u}_{t,k}\right\|_2}{\sum_k w_{t,k}}, \tag{2}$$

$$S = \max_h \frac{1}{|\mathcal{S}^h|\,|\mathcal{J}|} \sum_{t\in\mathcal{S}^h} \sum_{j\in\mathcal{J}} \mathbb{1}\left[\bar{q}_{t,j} \le \epsilon \;\vee\; \bar{q}_{t,j} \ge 1-\epsilon\right], \tag{3}$$

$$C = \max_h \frac{1}{|\mathcal{S}^h|} \sum_{t\in\mathcal{S}^h} \max_{(a,b)\in\mathcal{P}} \mathrm{pen}_{ab}(\mathbf{q}_t), \tag{4}$$

where E is the *reprojection residual*, with $\hat{\mathbf{u}}_{t,k}$ the fitted projection and H the frame height, so that models with different output resolutions are compared on a common scale; S is the *joint-limit saturation*, with $\bar{q}_{t,j} = (q_{t,j} - q_j^{\mathrm{lo}})/(q_j^{\mathrm{hi}} - q_j^{\mathrm{lo}})$, $\epsilon = 0.02$, and $\mathcal{J}$ the 12 joints observable from the keypoints (the four distal joints move no keypoint and are excluded); and C is the *self-collision depth*, with pen_{ab} the penetration depth between the simulator's collision boxes of links a and b, computed exactly by the separating-axis test, and $\mathcal{P}$ the non-adjacent pairs among the 13 links determined by those joints. Since Eq. equation 1 contains no collision term, C is independent of E and S. A pose with legal joint angles and a small residual can still place two fingers in the same space.

When a case's detection rate falls below 0.8, E, S, and C are marked as undecidable and excluded from the averages, which avoids the survivorship bias of scoring only a few clear frames. The detection rate itself is not gated. Likewise, a metric is marked as undecidable on a task whose ground-truth floor already exceeds a preset tolerance. For the benchmark score, each term is normalized by its value on ground-truth videos, using model/GT for the detection rate and GT/model for the three error terms. The four ratios are then averaged with equal weight:

$$\mathrm{HGA}_s = \frac{1}{4}\left(\frac{\bar{D}_s}{\bar{D}_s^{\mathrm{GT}}} + \frac{\bar{E}_s^{\mathrm{GT}}}{\bar{E}_s} + \frac{\bar{S}_s^{\mathrm{GT}}}{\bar{S}_s} + \frac{\bar{C}_s^{\mathrm{GT}}}{\bar{C}_s}\right), \tag{5}$$

where bars denote means over the decidable cases of evaluation subset s. Ground-truth videos score 1 by construction, and the final score is the macro-average of HGA_s over the training-set and held-out subsets.

A.2 Object Trajectory Metric Details

This section defines the Object Trajectory Score (OTS) and the Trajectory Consistency Horizon (TCH) in full. Both share the same point-tracking evidence and the same frozen anchors derived from ground-truth videos.

Point tracking. For each manipulated object o, we sample 24 mesh vertices, pose them with the recorded simulator trajectory, and project them through the recorded camera to obtain reference positions $\mathbf{p}_{t,i}$. The same points are queried at frame 0 and tracked through the generated video with CoTracker3 (Karaev et al., 2025), yielding positions $\hat{\mathbf{p}}_{t,i}$ and visibility flags. All distances are normalized by the side length of the tracking grid. Let $\mathcal{V}_o$ denote the valid point–frame pairs of o, i.e., pairs with $t \geq 1$ whose reference point lies in view and is reported visible by the tracker. Frame 0 is the conditioning frame and is therefore excluded. On these pairs we define the tracking error and the reference displacement as

$$e_o = \operatorname*{med}_{(t,i)\in\mathcal{V}_o} \left\|\hat{\mathbf{p}}_{t,i} - \mathbf{p}_{t,i}\right\|_2, \qquad d_o = \operatorname*{med}_{(t,i)\in\mathcal{V}_o} \left\|\mathbf{p}_{t,i} - \mathbf{p}_{0,i}\right\|_2, \tag{6}$$

where d_o equals the error that a frozen video repeating frame 0 would obtain.

Anchors. All anchors are computed from ground-truth videos only and then frozen, so adding new models never changes existing scores. e_o^{GT} and $\bar{d}_o$ average e_o and d_o over the ground-truth cases of the task, and the noise floor ϵ_τ is the tracking error of CoTracker3 on the ground-truth videos of task τ. An object is *moving* if $\bar{d}_o > 3\epsilon_\tau$ and *stationary* otherwise. A moving object whose dynamic range $\bar{d}_o - e_o^{\mathrm{GT}}$ falls below ϵ_τ has a degenerate denominator and is scored as stationary.

Object Trajectory Score. Each object is scored according to its class:

$$S_o = \begin{cases} \dfrac{\bar{d}_o - e_o}{\bar{d}_o - e_o^{\mathrm{GT}}}, & o \text{ moving}, \\[2ex] \dfrac{10\epsilon_\tau - e_o}{10\epsilon_\tau - e_o^{\mathrm{GT}}}, & o \text{ stationary}. \end{cases} \tag{7}$$

Thus $S_o = 1$ matches ground truth, and $S_o = 0$ corresponds to a stationary object that drifts by $10\epsilon_\tau$, or to a moving object that never moves in a case whose displacement equals the task average. Since $\bar{d}_o$ is a task-level anchor, the score of a frozen video on an individual case deviates from 0 with that case's actual displacement. Negative values are worse than either, and scores are not clipped. Because a model can lower e_o simply by letting the object vanish, we penalize lost objects. With the survival s_o defined as the fraction of in-view points that the tracker still holds and $\rho_o = s_o / s_o^{\mathrm{GT}}$, we set

$$S_o \leftarrow S_o - \big(1 - \gamma(\rho_o)\big)\,|S_o|, \qquad \gamma(\rho) = \begin{cases} \rho, & \rho < 0.5, \\ 1, & \text{otherwise}. \end{cases} \tag{8}$$

The penalty scales a positive score by ρ_o and pushes a negative score further below zero, so losing an object never raises the score, whatever its sign. We apply it as a gate rather than a continuous discount because moderate survival shortfalls are common under hand occlusion even on ground-truth videos, so only losing more than half of the achievable points is attributed to the model. If an object has no valid pair at all, i.e., every point is lost, e_o is undefined and the object is scored as a frozen video, $S_o = 0$. The OTS of a case is the mean of S_o over its objects.

Trajectory Consistency Horizon. For each moving object o, the per-frame error $m_o(t)$ is the median tracking error over the valid points of frame t, and the divergence onset is the first frame at which this error exceeds three times the noise floor for two consecutive frames:

$$t_o^* = \min\big\{\, t \geq 1 :\ m_o(t) > 3\epsilon_\tau \ \wedge\ m_o(t{+}1) > 3\epsilon_\tau \,\big\}. \tag{9}$$

The horizon of a case is set by its earliest-diverging moving object:

$$h = \min_{o\in\mathcal{M}} \frac{t_o^*}{T-1}, \qquad \mathrm{TCH} = \gamma(\bar{\rho}_{\mathcal{M}})\,\frac{h}{\bar{h}_\tau^{\mathrm{GT}}}, \tag{10}$$

where $\mathcal{M}$ is the set of moving objects and T the number of frames. The term $t_o^*/(T-1)$ is taken as 1 if o never diverges and as 0 if all of its points are lost, i.e., it counts as diverging immediately. $\bar{\rho}_{\mathcal{M}}$ is

the ratio between the mean survival of the moving objects and its ground-truth counterpart. Since h is non-negative, applying the penalty multiplicatively here is already monotone. Even ground-truth videos diverge under this rule because of tracker noise and occlusion, so we normalize by the mean ground-truth horizon $\bar{h}_\tau^{\mathrm{GT}}$ of task τ. Ground-truth videos therefore score close to 1 on average, and the score is not clipped. Tasks without moving objects, or whose ground-truth horizon is below 0.05, contribute no TCH.

Aggregation. Both metrics are computed only on cases that pass the frame-0 reproduction check. Case scores are averaged within each task, macro-averaged over tasks, and finally macro-averaged over the training-set and held-out subsets.

A.3 Inverse Dynamics Model Configuration

Per-task IDMs substantially outperform the shared multi-task IDM (0.87 vs. 0.53 mean replay success in Table 3). We attribute this to the heterogeneity of the task distribution. The tasks differ in visual appearance and kinematics, and a single shared model must jointly fit several distinct action distributions, which is harder to optimize than fitting one consistent distribution per task. Moreover, since the benchmark task set is fixed and per-task training is inexpensive, maintaining one checkpoint per task incurs little overhead. We therefore adopt per-task IDMs throughout the benchmark.

Even per-task IDMs do not achieve perfect replay, and the failures are deterministic across repeated runs. Inspection of the failing episodes shows that the simulator's success criteria depend on precise contact establishment and stable state retention, both of which are sensitive to small action errors under open-loop replay.

Table 3: Replay success on the training-set full-task sequences for actions recovered from real ground-truth videos, with a shared multi-task IDM versus per-task IDMs.

Action source	click_mouse	pick_bucket	water_plant	Mean
Recorded GT actions	1.00	1.00	1.00	1.00
Shared multi-task IDM	0.50	0.50	0.60	0.53
Per-task IDMs	0.90	0.90	0.80	0.87

A.4 Validation of the control-accuracy judges

The arm-side and finger-level judges are calibrated on ground-truth clips using three-vote accuracy, vote agreement, abstention rate, and false alarms on static controls. Judge-task pairs that fail calibration are excluded from scoring. The arm-side judge correctly identifies all 50 reference clips. The finger judge identifies the intended finger in 20 of 24 scored clips and produces no false alarms on eight static controls. Its errors are concentrated on adjacent fingers, motivating per-task reporting of finger-level accuracy.

Fine-grained finger judgments depend on the judge model. Under the same prompt and voting protocol, accuracy ranges from 0.125 to 0.833 across the three judges in Table 4. We therefore use gpt-6-astra, which achieves the highest accuracy and no false alarms on the controls.

Table 4: Judge sensitivity on the finger control-accuracy GT set. All judges use the same 24 scored clips, the same probe, and $K=3$ votes, and only the judge model varies. Specificity is measured on 8 unperturbed control clips.

Judge	GT accuracy (scored tasks)	Vote agreement	Abstain rate	Specificity (8 controls)
gpt-5.4	3/24 = 0.125	0.719	5/24	5/8 false alarms
gpt-5.6-sol	12/24 = 0.500	0.702	2/24	5/8 false alarms
gpt-6-astra	**20/24 = 0.833**	**0.947**	**0/24**	**0/8 false alarms**

Table 5: Agreement between outcome-consistency judges and human annotators.

Judge	n	ρ	κ_q	GT AUC	Binary agreement	Fidelity Δ
gpt-5.4	182	0.389	0.323	0.677	0.797	$+0.006$
gpt-5.6-sol	183	0.654	0.623	0.840	0.817	-0.019
gpt-6-sol	177	0.723	0.668	0.906	0.860	-0.055
Human annotator	—	—	—	0.964	—	—

A.5 Human validation of the outcome-consistency judge

Human annotators rated a subset of reference and generated clips using the same 1–5 scale and task-specific anchors as the outcome-consistency judges. Table 5 compares each judge with the human on clips rated by both, with n denoting the number of shared clips. Spearman's ρ measures whether their scores order clips similarly, while quadratic-weighted κ_q measures agreement on the ordinal ratings. GT AUC measures how well scores separate successful from failed reference clips across all eight tasks. For binary agreement, human and judge ratings are each converted to success or failure using the threshold $s \geq 4$, and the fraction of matching judgments is reported. To compute fidelity, each rater separately compares the generated clip with its reference clip using their own ratings. Fidelity Δ is the judge's aggregate fidelity minus the human's aggregate fidelity.

Among the three judges, gpt-6-sol achieves the highest agreement on clip ratings and the strongest success–failure separation. Its aggregate fidelity is 0.055 below the human-scored value on the annotated subset.