BENCHMARKING CODING AGENTS FOR INTERACTIVE WORLD MODELING FROM EMBODIED VIDEOS

Anonymous authors
Paper under double-blind review

ABSTRACT

Building interactive simulators from real-world observations is a promising way to scale embodied data, but current pipelines still rely heavily on manual environment construction and calibration. We study whether frontier foundation models and coding agents can automate this process end to end. We formulate *autonomous video-to-simulation* as a software engineering task in which an agent observes an embodied video, constructs the corresponding simulated environment and robot behavior, and iteratively refines the result through execution feedback. To evaluate this capability, we introduce **Video2World**, a benchmark comprising 222 reconstruction instances derived from 189 robot and human demonstration videos. Video2World measures reconstructed worlds along geometric fidelity, dynamic fidelity, and functional correctness, capturing spatial perception, physical reasoning, and executable interaction. Evaluating 9 frontier coding-agent systems reveals a sharp improvement in Task success beginning with Claude Opus 5, rising from below 5% to over 15%, while substantial gaps to human-assisted reconstruction remain. We further find that worlds that look better could work worse: better visual fidelity does not always lead to higher task success. This echoes the broader gap between perceptual realism and factual correctness observed in generative models. Our data and code will be publicly released.

1 INTRODUCTION

Embodied AI aims to build intelligent systems that can perceive, reason, and act effectively in the physical world, enabling applications such as robotic manipulation, household assistance, and industrial automation (Duan et al., 2022). Fully acquiring these capabilities requires large amounts of diverse embodied interaction data, which is expensive, time-consuming, and difficult to scale in the real world (Khazatsky et al., 2024; Mandlekar et al., 2023). To alleviate this bottleneck, many prior works have explored interactive simulation as a scalable data engine, where agents can act, observe state transitions, and automatically collect embodied experience (Nasiriany et al., 2024; Wang et al., 2024a;b). Conventional Real-to-Sim pipelines, however, still require substantial engineering effort to reconstruct and configure simulation environments, calibrate geometry and physical properties, align coordinate systems, and integrate perception and simulation components (Torne et al., 2024; Han et al., 2026; Chen et al., 2026a).

In recent years, foundation models and agents have made rapid progress in spatial perception, visual understanding and planning, demonstrating increasingly strong capabilities in solving complex software and embodied tasks (Yang et al., 2025; Jimenez et al., 2024; Xie et al., 2024; Fu et al., 2026; You et al., 2026). These advances make it increasingly promising to automate the full process of interactive world modeling in simulators through an agent itself (Ye et al., 2025; Chen et al., 2026a). Such automation could substantially reduce the human effort, and enable the efficient conversion of real-world observations into scalable interactive environments for data generation, policy learning, and evaluation. However, existing benchmarks typically assess only one or a few isolated capabilities for interactive environment reconstruction, such as spatial understanding, physical rule following, scene reconstruction, or environment editing (Yang et al., 2025; Wang et al., 2026; Gao et al., 2026; Zhao et al., 2026; Tang et al., 2026). As shown in Table 1, they provide only a fragmented view of performance and do not reveal how effectively current frontier models and agents can integrate these capabilities to autonomously build, execute, and refine interactive worlds.

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.

Table 1: **Comparison with related benchmarks.** ✓ yes and × no denote whether the benchmark focuses on evaluating the corresponding capability: *Embodied* for embodied reasoning and planning; *Physics* for physical-world dynamics and interaction; *Spatial* for spatial understanding; *Agent* for agentic planning and iterative interaction; *Coding* for executable code generation.

Benchmark	Input	Embodied	Physics	Spatial	Agent	Coding	Evaluation Task
SWE-bench (2024)	Code Repository	×	×	×	×	✓	Code Generation
VSI-Bench (2025)	RGB Video	×	×	✓	×	×	Question Answer
RoboDojo (2026b)	Interactive Env.	✓	✓	×	×	×	Robot Manipulation
CaP-X (2026)	Interactive Env.	✓	✓	×	✓	✓	Robot Manipulation
RLE-Bench (2026)	Interactive Env.	✓	✓	✓	✓	✓	Robotics Engineering
EmbodiedSWE-Bench (2026)	Interactive Env.	✓	✓	×	✓	✓	Task Completion
SimuScene (2026)	Text	×	✓	×	×	✓	Physics Simulation
3DCodeBench (2026)	Text / Image	×	×	✓	✓	✓	3D Modeling
SceneActBench (2026)	Image / Video	×	×	✓	✓	✓	3D Scene Action
BVB (2026)	RGB Video	×	×	✓	✓	✓	Editable Env. Rec.
Video2World (Ours)	RGB Video	✓	✓	✓	✓	✓	Interactive Env. Rec.

To fill this gap and directly test whether foundation models and coding agents can autonomously construct interactive physical worlds, we formulate *autonomous video-to-simulation* as an end-to-end software engineering task (Ye et al., 2025; Chen et al., 2026a). Concretely, given an RGB video of an embodied interaction together with a target simulator, an agent must construct the environment, implement the robot behavior, and iteratively refine the simulation through execution feedback. To support systematic evaluation, we construct a dataset that pairs embodied videos with their corresponding simulator environments, including aligned scene configurations, object states, and interaction trajectories. This paired structure allows us to compare an agent-generated simulation directly against the underlying target environment and assess the full reconstruction quality. In this way, we measure not only whether an agent can understand the world, but also whether it can integrate embodied reasoning, physical understanding, spatial perception, and agentic coding abilities to recover an interactive environment that faithfully reproduces the demonstrated physical world.

We introduce **Video2World**, a benchmark for evaluating foundation models and coding agents on autonomous interactive world modeling from embodied videos. Video2World comprises 222 reconstruction instances across 39 task families, constructed from 189 video clips spanning robot executions, egocentric human recordings, and third-person human demonstrations. To evaluate reconstruction quality comprehensively, we design metrics from three complementary perspectives: *geometric fidelity*, which measures how accurately the reconstructed scene matches the spatial structure of the source environment; *dynamic fidelity*, which evaluates whether simulated object motions and state transitions reproduce the observed physical evolution; and *functional correctness*, which assesses whether the reconstructed environment supports successful task execution through physical interaction. Together, these metrics reflect an agent's ability to integrate spatial perception, physical reasoning, and executable coding for end-to-end interactive world modeling.

We evaluate 9 frontier coding-agent systems under a common benchmark interface and uncover several key findings. First, current coding agents can construct executable simulated worlds, but reliable completion of the demonstrated interactions remains challenging. Second, reconstruction fidelity and functional correctness are related but not equivalent: models with stronger geometric accuracy do not necessarily achieve higher task success. Third, human-assisted reconstruction provides a substantially stronger reference, while still falling short of perfect task completion. Finally, different coding-agent systems exhibit distinct strengths across the evaluated dimensions, suggesting that overall video-to-simulation performance emerges from their combination rather than any single ability.

2 Preliminary: Autonomous Video-to-Sim via Coding Agents

We study the autonomous video-to-sim task that asks a coding agent to implement an executable simulator program as an interactive environment from an embodied video. The agent must recover task-relevant scene structure, physical behavior, and robot control from visual observations.

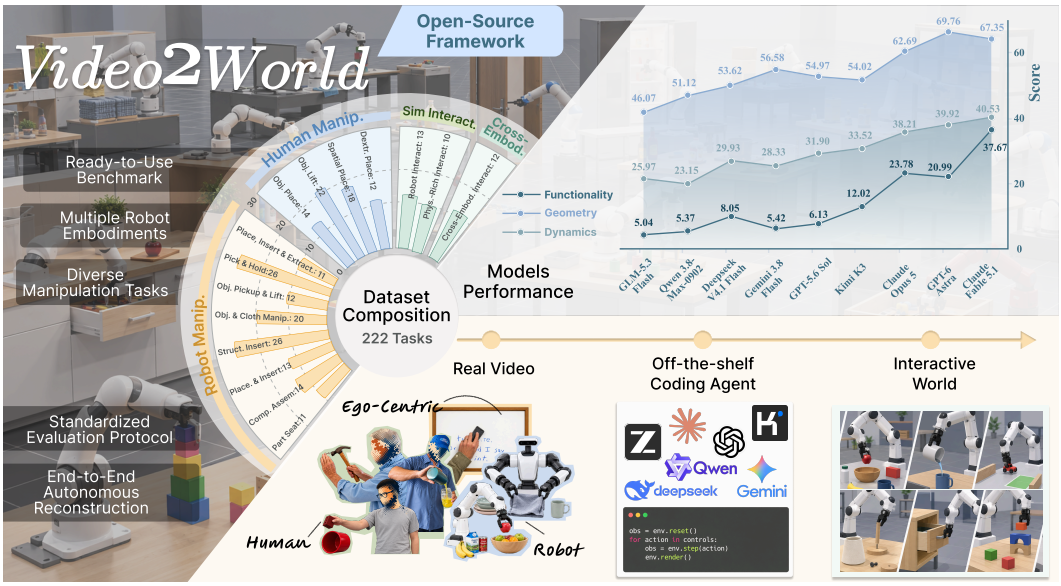

Figure 1: **Video2World** benchmarks the conversion of embodied videos into simulated worlds. Given an RGB video and simulator tools, a coding agent constructs the environment and robot behavior, then revises its implementation using execution feedback. Evaluation examines geometric fidelity, physically generated object motion, task completion.

Problem Formulation. Given an RGB embodied video v, a target simulator $\mathcal{S}$, and a set of general-purpose tools $\mathcal{T}$, a coding agent $\mathcal{A}$ produces

$$P = (\eta, \pi) \sim \mathcal{A}(\cdot \mid v, \mathcal{S}, \mathcal{T}), \tag{1}$$

where η specifies the reconstructed simulation environment and π specifies the robot behavior under the robot embodiment and control interface provided by $\mathcal{S}$. Executing P yields a rollout consisting of a sequence of simulated states and robot actions:

$$\tau_P = (s_0, a_0, s_1, \ldots, a_{T-1}, s_T), \qquad a_t = \pi(t, s_{\le t}), \qquad s_{t+1} = F_\eta(s_t, a_t), \tag{2}$$

where F_η denotes the simulator dynamics instantiated by η, while s_t and a_t denote the simulated state and robot action at step t, respectively. A reconstruction is functionally correct if the resulting rollout reproduces the task demonstrated in v through robot execution in the reconstructed environment.

Agentic Video-to-Sim. We implement agentic video-to-sim as an iterative coding process. Given the source video v, target simulator $\mathcal{S}$, and tools $\mathcal{T}$, the agent inspects visual observations and available resources, edits the simulation program, executes the current reconstruction, and uses the resulting feedback to revise both the environment η and robot behavior π. This loop allows the agent to progressively correct implementation errors and discrepancies between the simulated interaction and the source demonstration. Construction terminates upon submission or when the runtime or tool-call budget is exhausted. The resulting program $P = (\eta, \pi)$ is then fixed for evaluation.

3 Video2World: Benchmarking Coding Agent Video-to-Sim

Video2World evaluates coding agents for end-to-end reconstruction of interactive physical worlds from embodied videos. We first collect embodied videos and construct the corresponding world assets and annotations, then design a standardized agentic video-to-sim protocol, and finally define evaluation metrics for assessing the reconstructed worlds.

3.1 Property Construction

Video2World constructs object-centric world properties from embodied videos to support evaluation of reconstructed interactive environments. We focus on the objects, physical context, motion, and task states that determine how the demonstrated interaction unfolds. These properties provide common evaluation targets across different robot embodiments and control strategies.

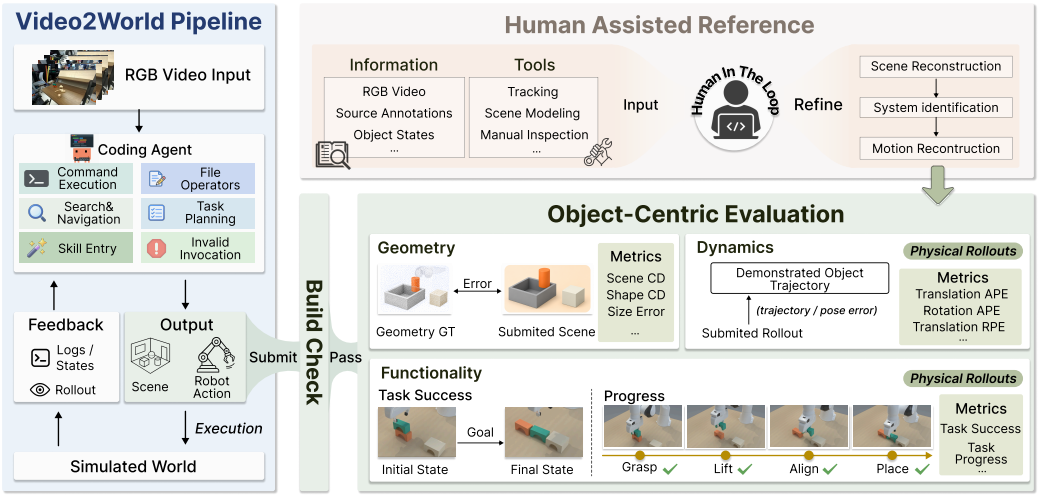

Figure 2: **Overview of Video2World.** A coding agent constructs the scene and robot behavior in a sandbox, using video inspection, simulator APIs, asset resources, and execution feedback. Submitted reconstructions are evaluated against hidden scene geometry, demonstrated object motion, and task requirements. Evaluation separately measures geometric fidelity, dynamic fidelity of object motion, and functional correctness under physical execution.

Video and Task Collection. To cover diverse scenarios and tasks, we assemble 189 video clips forming 222 reconstruction instances from FurnitureBench Heo et al. (2023), DROID Khazatsky et al. (2024), RoboDojo Chen et al. (2026b), HOI4D Liu et al. (2022), HOT3D Banerjee et al. (2025), DexYCB Chao et al. (2021), OakInk2 Zhan et al. (2024), and an in-house household dataset. The collection includes both real and simulated demonstrations spanning diverse embodiments, viewpoints, scene configurations, and interaction mechanisms. Each reconstruction instance pairs a source video with a target configuration specifying the simulator, robot embodiment, and control interface. We select clips in which the participating objects, essential interaction process, and task outcome are visually discernible. Instances are organized into fixed task families according to source-specific task categories, target embodiments, and annotation configurations (Appendices B.2).

Object-Centric World Annotations. To evaluate whether an agent reconstructs an interactive world rather than merely a visually similar scene, Video2World requires task-relevant properties that capture how the demonstrated interaction is structured and evolves. We therefore represent each video with object-centric annotations covering three aspects: scene geometry, object motion, and task requirements. Concretely, scene geometry records object shape, scale, articulation, and spatial relations; object-motion annotations describe how participating objects move and change state; and task specifications define entity roles, initial conditions, essential interaction events, terminal goals, and progress milestones. These annotations provide a common target for measuring what the agent has reconstructed and reproduced, while remaining independent of the particular simulator implementation or robot control strategy. The annotation sources, hidden specifications, and verification process are described in Appendices B.3 and B.7.

Cross-Embodiment Alignment. Video2World aligns human and robot demonstrations through task-relevant object states and motions, providing common evaluation targets across embodiments and control strategies. Concretely, we derive a reference behavior for each target robot from the hidden annotations and execute it under contact physics to verify that the demonstrated interaction can be physically reproduced. This object-centric alignment preserves shared task-level targets while allowing embodiment-specific robot trajectories. Reference construction is detailed in Appendix B.5.

As a supplementary check, we compare source and simulated effector motion on 40 human-assisted reference instances from 35 videos, selected by task success in the original evaluation. The six motion statistics show Spearman correlations of 0.74–0.99, including 0.80 for net rotation, suggesting that these reference executions also preserve key motion characteristics beyond object motion (Figure A6). The comparison protocol and metric definitions are provided in Appendix C.5.

3.2 AGENTIC PROTOCOL DESIGN

With the above collected properties, we devise a standardized protocol for testing how coding agents construct interactive worlds from videos. It controls the agent's available resources, reconstruction budget, and execution setting, while leaving the internal construction strategy fully autonomous. This design evaluates deployable coding-agent systems under a common task interface rather than enforcing a common internal scaffold.

Agent Workspace. Each reconstruction instance pairs a source video with a target configuration specifying the simulator, robot embodiment, and control interface. The coding agent operates in an isolated sandbox containing the source video, simulator APIs, documentation, available assets, and general-purpose coding tools. Within this workspace, the agent constructs the simulated environment η_i and robot behavior π_i, and may iteratively inspect simulation outputs and revise its implementation through execution feedback. Each reconstruction instance specifies the source video, target simulator, robot embodiment, and control interface. Construction follows framework-specific resource limits and stopping rules, without a fixed episode-wide tool-call or model-turn budget. More details are specified in Appendices A.1, A.4, and A.5.

Environment Execution. After submission, the reconstructed environment and robot behavior are executed by the evaluator under the assigned simulator, embodiment, and control settings. The evaluator initializes the scene, controls simulation stepping, and records object states, articulation, and contacts for evaluation. After initialization, all object changes must arise through simulator dynamics rather than direct state assignment. Each model receives one independent construction attempt per instance, with submitted code and execution traces retained for reproducibility. More details are in Appendices A.2 and A.3.

3.3 EVALUATION METRICS DESIGN

We evaluate each reconstructed world from complementary perspectives that capture whether it can be executed, whether it preserves the source scene and physical evolution, and whether it supports the demonstrated interaction. Specifically, Video2World reports executability, geometric fidelity, dynamic fidelity, and functional correctness.

Executability. We first measure whether the submitted reconstruction can be successfully executed through the prescribed simulator interface. Build Rate reports the fraction of submissions that produce a valid executable environment and robot behavior under the assigned target configuration.

Geometric Fidelity. Geometric fidelity measures how accurately the reconstructed initial scene matches the annotated source environment. We evaluate both scene-level surface correspondence and object-level reconstruction quality, including object shape, size, position, articulation, and relative spatial relations. For scene-level evaluation, let $\mathcal{E}$ denote the annotated non-support entities, $\mathcal{S}_e$ their reference surfaces, $\widehat{\mathcal{C}}$ the registered reconstructed non-support surface, and $\mathcal{B}$ the reference surface including supports. We compute an interaction-weighted geometric error

$$E_{\text{geo}} = \sum_{e \in \mathcal{E}} \bar{w}_e \, \mathcal{D}\Big(\mathcal{S}_e, \widehat{\mathcal{C}}\Big) + \mathcal{D}_\omega\Big(\widehat{\mathcal{C}}, \mathcal{B}\Big), \tag{3}$$

where $\bar{w}_e = w_e / \sum_{j \in \mathcal{E}} w_j$, and w_e assign larger weights to manipulated objects, receiving objects, and nearby structures that directly affect the demonstrated interaction. The directed distances $\mathcal{D}$ and $\mathcal{D}_\omega$ use the capped median and capped weighted-median definitions in Appendix C.1, respectively, with candidate-point weights ω inherited from the nearest reference non-support entity. Object-level metrics further report shape, scale, position, and relative-pose errors. Full surface-sampling, registration, and object-level definitions are provided in Appendix C.1.

Dynamic Fidelity. Dynamic fidelity measures how closely physical execution in the reconstructed environment reproduces the object motion observed in the source video. For object e, let $p_e(t)$ and $p_e^*(t)$ denote its simulated and demonstrated positions at time t. We measure translational motion

Table 2: **Overall performance on Video2World.** V2WScore combines Build, functionality, geometry, and dynamics into a single 0–100 score, macro-averaged across task families. Build measures executability; Functionality reports task success and stage-wise progress; Geometry measures scene- and object-level reconstruction accuracy; and Dynamics reports translation APE, rotation APE, and translation RPE. The human-assisted reference additionally uses source information and manual refinement. Best and second-best automated results are shown in bold and underlined, respectively; ties share the same formatting. Metric definitions, deformation and partial-observation terms, aggregation, and missing-data handling are detailed in Appendices C.6 and C.4.

	V2WScore	Build	Functionality		Geometry			Dynamics		
Method	Score (0–100) ↑	Rate (%) ↑	Success (%) ↑	Progress (0–1) ↑	Scene CD (cm) ↓	Shape CD (cm) ↓	Size err. (cm) ↓	T-APE (cm) ↓	R-APE (deg) ↓	T-RPE (cm) ↓
Human-assisted Ref	74.24	100.0	58.8	0.70	2.19	0.15	0.43	9.40	41.09	2.42
Claude Fable 5.1	**48.52**	**93.1**	**25.5**	**0.53**	**5.35**	1.00	2.78	**18.71**	104.61	8.25
GPT-6 Astra	43.55	**93.1**	10.7	0.33	6.23	**0.69**	**1.62**	20.78	**95.08**	**6.19**
Claude Opus5	41.56	92.8	16.0	0.34	6.01	1.14	3.13	19.37	110.85	8.42
Kimi K3	33.19	92.0	4.8	0.21	8.08	1.41	4.23	27.28	111.16	14.29
GPT-5.6 Sol	31.00	92.5	2.5	0.11	7.79	1.37	3.76	31.68	105.03	18.00
DeepSeek-V4.1 Flash	30.54	92.4	4.1	0.13	8.59	1.50	4.13	34.64	104.52	20.85
Gemini-3.8 Flash	30.11	**93.1**	2.0	0.09	8.57	1.25	3.67	37.97	114.82	28.65
Qwen-3.8 Max-0902	26.55	87.6	1.5	0.11	8.36	1.51	4.13	41.28	110.06	27.38
GLM-5.3 Flash	25.69	87.7	2.0	0.09	9.58	1.62	5.04	41.16	103.83	30.16

discrepancy relative to each trajectory's initial state:

$$E^{e}_{\mathrm{dyn}} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \rho_{c_p}(\|\Delta p_e(t) - \Delta p^*_e(t)\|_2)\,. \tag{4}$$

Here $\Delta p_e(t) = p_e(t) - p_e(0)$ and $\Delta p^*_e(t) = p^*_e(t) - p^*_e(0)$, $\mathcal{T}$ contains the comparison times after temporal correspondence, and $\rho_{c_p}(r) = \min\{r, c_p\}$ with $c_p = 1\,\mathrm{m}$, following Appendix C.2. This removes global translation offsets while preserving differences in motion magnitude and temporal evolution. We additionally evaluate rotational motion, account for known object symmetries, and use frame-level surface correspondence for deformable objects. These metrics focus on task-relevant object dynamics, allowing different control strategies to realize the same physical interaction.

Functional Correctness. Functional correctness evaluates whether the reconstructed world supports successful execution of the demonstrated task. For each task family, we define object-centric requirements covering task-defining initial conditions, essential interaction events and their ordering, and the terminal goal. We report both Task Success and graded Task Progress through intermediate milestones, allowing different embodiments, trajectories, and execution timings as long as the required interaction is physically realized. More details are in Appendix C.3.

4 Experiments

We investigate the reliability of autonomous video-to-simulation via coding agents, the performance relative to human-assisted reconstruction, the relationship between geometric fidelity and functional correctness, and the effector-motion correspondence in selected HAR executions.

4.1 Experimental Setup

Coding agents. We evaluate 9 coding-agent models on 222 instances, covering 189 video clips and 39 task families. Seven models run through OpenCode, while GPT-6 Astra and Fable-5.1 use Codex CLI and Claude Code, respectively. We evaluate each configuration as a complete agent system, including its underlying model, coding scaffold, and inference settings. Each model makes one autonomous construction attempt per instance under the benchmark interface and assigned target configuration (Section 3). All agents have access to physical simulation and rendered rollout feedback and may iteratively revise both the scene and robot behavior within a 180-minute wall-clock limit.

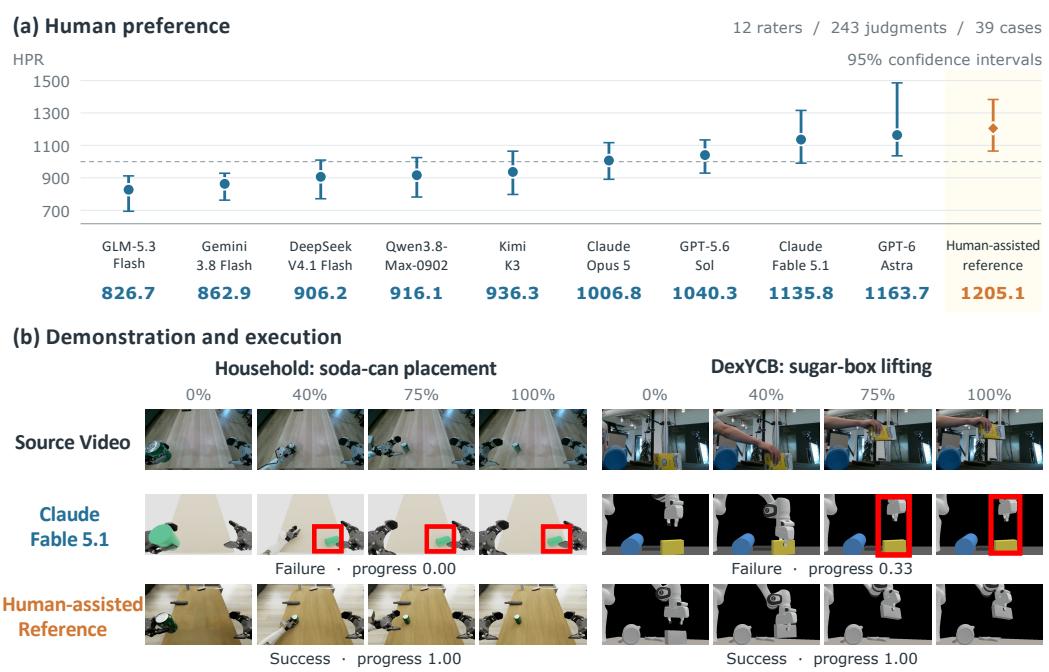

Figure 3: **Human in the loop reference and evaluation** (a) Human preference ratings of coding models and the human-assisted reference, based on 243 judgments from 12 raters across 39 cases. (b) Source demonstrations and simulated executions for two representative tasks. Each method executes its own behavior in its reconstructed environment. Outcomes and task progress are reported below each simulated sequence; red boxes highlight failed executions.

The submission retained by the prescribed stopping rule is frozen for evaluation, without human correction or selection based on hidden scores. More details are in Appendix A.

Human-assisted reference. HAR leads across the reported automatic metrics and obtains the highest Human Preference Rating (HPR) point estimate in our pairwise study of overall scene-and-interaction faithfulness (Figure 3a). The study protocol, participant information, and rating procedure are detailed in Appendix B.6. Successful HAR rollouts further demonstrate that the target tasks are feasible under the benchmark configuration, including cases where coding agents produce incomplete or unsuccessful executions (Figure 3b). However, HAR itself does not achieve uniformly reliable task completion, showing that improved scene knowledge and manual refinement do not eliminate the difficulty of reconstructing executable behavior.

Evaluation. We report V2WScore, a 0–100 aggregate of Build, functionality, geometry, and dynamics, alongside its component metrics. The score is computed independently for each method, which executes its own robot behavior in its reconstructed environment. Task judgments use the recorded rollout and configuration-specific physical-validity checks. Build Rate, Task Success, and Task Progress are aggregated over all applicable instances, with failed builds assigned zero. Continuous errors are computed on evaluable comparisons, with coverage reported. Each metric is averaged within task families and then across eligible families with equal weight. Reported 95% confidence intervals use stratified source-cluster bootstrapping, retaining related clips and target configurations together. More details and experiments are in Appendix C and Appendix D.

4.2 Main Results

Coding agents build executable worlds but remain unreliable at full task completion. Most agent-generated reconstructions can be simulated, as reflected by high Build Rates, but Task Progress consistently exceeds Task Success (Table 2), showing that agents often achieve partial interaction without completing all required conditions. Even Fable-5.1, the strongest model by V2WScore, exhibits this gap. Moreover, strong object reconstruction does not necessarily yield stronger functional

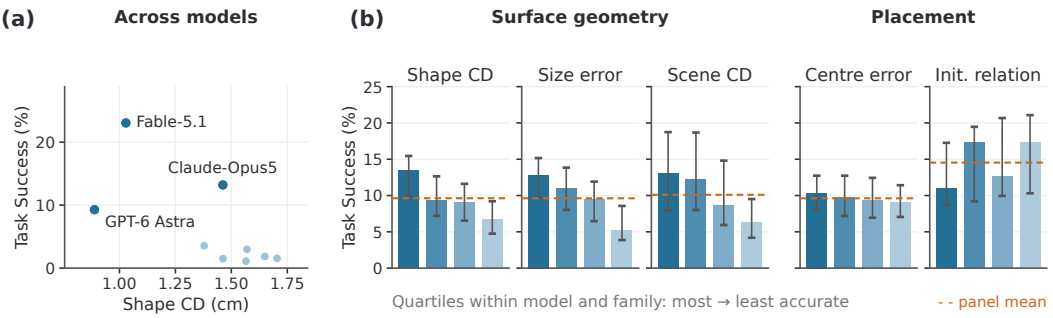

Figure 4: **Geometric fidelity versus functional correctness.** (a) System-level Shape CD against Task Success; both are macro-averaged over task families, and Task Success counts failed builds as failures. (b) Task Success across geometric-error quartiles formed within each model and task family, averaged equally across families and then models. Dashed lines mark the mean of each panel; error bars show 95% source-video bootstrap intervals. The initial relation is the object–receiver relation before execution, since the terminal relation enters the success predicate.

performance: GPT-6 Astra leads in object shape and size accuracy, while Fable-5.1 achieves better scene-level and task performance. These results highlight a gap between reconstructing individual assets and jointly recovering a scene and behavior that support successful interaction.

Human-assisted reconstruction provides a strong but empirical reference. HAR leads across the reported automatic metrics and is also preferred by human evaluators (Figure 3a), supporting its use as a high-quality reference. Successful HAR rollouts further demonstrate that the target tasks are feasible under the benchmark configuration, including cases where coding agents produce incomplete or unsuccessful executions (Figure 3b). However, HAR itself does not achieve uniformly reliable task completion, showing that improved scene knowledge and manual refinement do not eliminate the difficulty of reconstructing executable behavior.

Geometric fidelity helps but does not guarantee functional correctness. Lower shape, size, and scene errors are generally associated with higher Task Success within models and task families (Figure 4b), yet many geometrically accurate reconstructions still fail. The system-level comparison shows the same mismatch: GPT-6 Astra achieves the strongest object-shape fidelity, whereas Fable-5.1 completes more tasks (Figure 4a). Conversely, some reconstructions remain functional despite spatial disagreement with the source, since robot behavior can adapt to the reconstructed layout. Geometry and functionality should therefore be evaluated separately to distinguish faithful reconstruction from task-effective reconstruction.

Successful reconstructions also recover key motion characteristics. We compare effector-motion statistics between source demonstrations and successful simulated reference executions, including motion duration, path length, peak speed, lift height, net displacement, and net rotation. Figure A6 shows broadly similar motion characteristics across the two, providing additional evidence that successful reconstructions capture not only terminal states but also important temporal and spatial properties of the demonstrated interaction. Detailed statistics are provided in Appendix C.5.

5 Related Work

Real-to-Sim Reconstruction for Robot Manipulation. Real-to-Sim methods reconstruct interactive simulations from real-world observations for robot learning and evaluation. RialTo and Re^{3}Sim reconstruct environments for policy learning and transfer (Torne et al., 2024; Han et al., 2026), while PhysTwin recovers deformable-object models that reproduce observed interactions (Jiang et al., 2025). GenSim and RoboGen use foundation models to generate tasks, environments, and robot behaviors for scalable simulation-based learning (Wang et al., 2024a;b). More closely related, Video2Policy reconstructs manipulation tasks from internet videos and uses language models to generate and refine rewards for policy learning (Ye et al., 2025). Agentic Real-to-Sim coordinates vision-language agents

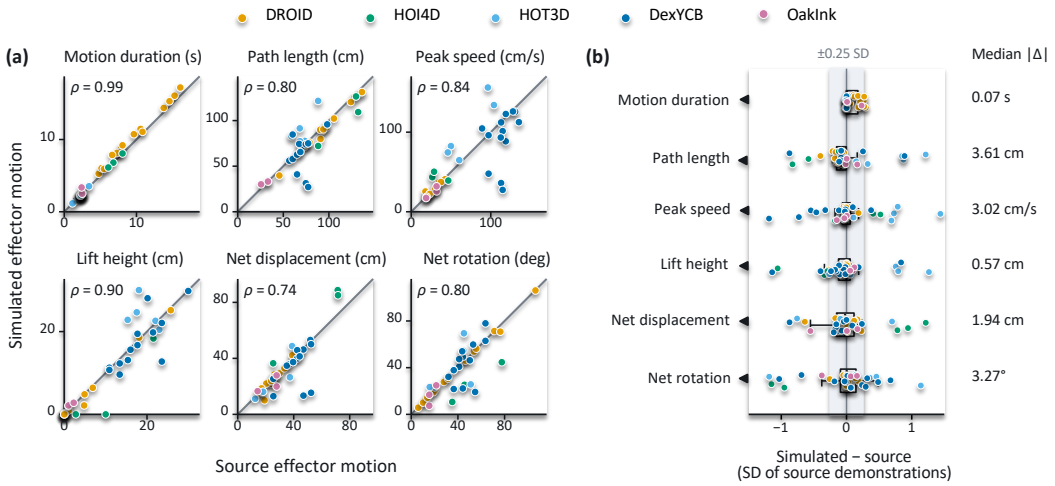

Figure 5: **Robot Effector Motion Beyond Object-Centric Evaluation.** (a) Source versus simulated effector statistics for 40 human-assisted reference instances from 35 videos, selected by task success in the original evaluation and colored by source. We compare gripper TCPs on DROID and human wrists with robot hand roots on human-video sources. Diagonals indicate equality; ρ is Spearman correlation. (b) Simulated-minus-source differences normalized by the source sample SD. Boxes show median/IQR; whiskers use $1.5\times$ IQR. Shading marks ± 0.25 SD; arrows count off-scale points; the right column gives median absolute differences. See Appendix C.5 for definitions.

with specialized perception and simulation tools to convert recorded interactions into executable episode twins (Chen et al., 2026a).

Foundation-Model Agents for Embodied Tasks. Foundation models connect semantic reasoning with embodied action through planning, code generation, and execution feedback. SayCan grounds language-based planning in robotic skill affordances (Ahn et al., 2022), while PaLM-E integrates visual and continuous sensor observations for embodied reasoning (Driess et al., 2023). Code as Policies generates executable programs that compose perception and control APIs (Liang et al., 2023), and VoxPoser constructs spatial value maps through language-model-generated code (Huang et al., 2023). Voyager further demonstrates how execution feedback enables program revision and reusable skill acquisition in Minecraft (Wang et al., 2023).

Benchmarks for Coding and Agentic Construction. Coding-agent benchmarks evaluate program execution and refinement. SWE-bench measures repository-level issue resolution (Jimenez et al., 2024); embodied benchmarks include CaP-X for robot-control programs under varied interfaces and observations (Fu et al., 2026), RLE-Bench for robot learning engineering (RLE-Bench contributors, 2026), and EmbodiedSWE-Bench for long-horizon manipulation and other physical tasks (You et al., 2026). Construction benchmarks cover procedural modeling from text and images in 3DCodeBench (Gao et al., 2026), visually conditioned scene operations against hidden geometric targets in SceneActBench (Zhao et al., 2026), and code generation for text-specified physical scenarios in SimuScene (Wang et al., 2026). Particularly related, BVB evaluates video reconstruction as animated Blender scenes through spatiotemporal facts and perceptual similarity (Tang et al., 2026).

6 Conclusion

We introduced **Video2World**, a benchmark for autonomous reconstruction of interactive physical worlds from embodied videos. Video2World formulates video-to-simulation as an end-to-end software engineering task and evaluates geometric fidelity, dynamic fidelity, and functional correctness. Our experiments show that current coding agents can often build executable simulations, yet reliable task completion remains challenging. We further find that geometric fidelity does not fully determine functional correctness, and even human-assisted reconstruction remains imperfect. These results highlight interactive world reconstruction as a joint scene–behavior problem and motivate execution-grounded evaluation beyond individual fidelity metrics.

AI USE STATEMENT

Generative AI was used only to produce synthetic data for the experiments. DeepSeek V4.1 Flash, GLM-5.3 Flash, Kimi K3, GPT-5.6 Sol, Gemini 3.8 Flash, Claude Opus 5, and Qwen3.8 Max were run as coding agents through OpenCode using OpenRouter model APIs; GPT-6 Astra and Fable-5.1 were run through Codex CLI and Claude Code, respectively. Given an embodied video and simulator tools, each agent attempted to construct an executable simulated environment and robot behavior. The resulting submissions and, when executable, their simulation rollouts formed the synthetic experimental data evaluated in this work.

ETHICS STATEMENT

Our study evaluates agent-generated robot behaviors in simulation and does not establish their safety for deployment on physical robots. The benchmark uses existing embodied demonstrations, including videos of people; access to and reuse of source media should follow the terms of the originating datasets. We report performance at the task and model level rather than making claims about individuals.

REPRODUCIBILITY STATEMENT

The paper and appendix describe the task definitions, agent interface, model configurations, simulator versions, execution protocol, evaluation metrics, and score aggregation. Submission packages, agent transcripts, and simulation records are retained with their associated configurations for auditing individual runs. Exact reruns also depend on access to the source videos, simulator assets, software versions, and model services.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can, not as i say: Grounding language in robotic affordances, 2022. URL https://arxiv.org/abs/2204.01691.

Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Shangchen Han, Fan Zhang, Linguang Zhang, Jade Fountain, Edward Miller, Selen Basol, Richard Newcombe, Robert Wang, Jakob Julian Engel, and Tomas Hodan. HOT3D: Hand and object tracking in 3D from egocentric multi-view videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7061–7071, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Banerjee_HOT3D_Hand_and_Object_Tracking_in_3D_from_Egocentric_Multi-View_CVPR_2025_paper.html.

Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S. Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, Jan Kautz, and Dieter Fox. DexYCB: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9044–9053, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Chao_DexYCB_A_Benchmark_for_Capturing_Hand_Grasping_of_Objects_CVPR_2021_paper.html.

Guanxiong Chen, Qianjun Xia, Jiawei Peng, Heng Zhang, Pengyu Jing, Bole Ma, Justin Qian, Yixian Cheng, Ziyi Jiao, Bingyang Zhou, Yiduo Qu, Luoxin Ye, Kaifeng Zhang, Kunyi Wang, Weijia Zeng, Yunuo Chen, Pengzhi Yang, Ziqiu Zeng, Siyuan Luo, Huamin Wang, Chao Liu, Alan Yuille, Fan Shi, Changxi Zheng, Yunzhu Li, Chenfanfu Jiang, and Peter Yichen Chen. Agentic real2sim: Physics-based world modeling with vision-language agents, 2026a. URL https://arxiv.org/abs/2607.19190.

Tianxing Chen, Yue Chen, Zixuan Li, Junyuan Tang, Kailun Su, Haoran Lu, Weijie Wan, Baijun Chen, Songling Liu, Haowen Yan, Honghao Su, Zhiyang Dou, Kaixuan Wang, Dandan Zhang, Yunze Liu, Yan Qin, Qiwei Liang, Qiwei Wu, Zijian Lin, Wenwei Lin, Yuran Wang, Minghua He, Tianshu Wu, Ruihai Wu, Jingquan Zhou, Kai-Chong Lei, Haibao Yu, Yuanfeng Ji, Weiyang Jin, Guanyu Lin, Xiaofan Li, Qi Xiong, Renjing Xu, Zhongyu Li, Wenhao Chai, Enze Xie, Ziwei Wang, Yao Mu, Hao Dong, Wojciech Matusik, Mingyu Ding, Wenbo Ding, Ping Luo, and Masayoshi Tomizuka. Robodojo: A unified sim-and-real benchmark for comprehensive evaluation of generalist robot manipulation policies, 2026b. URL https://arxiv.org/abs/2607.04434.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. Palm-e: An embodied multimodal language model, 2023. URL https://arxiv.org/abs/2303.03378.

Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. A survey of embodied ai: From simulators to research tasks, 2022. URL https://arxiv.org/abs/2103.04918.

Letian Fu, Justin Yu, Karim El-Refai, Ethan Kou, Haoru Xue, Huang Huang, Wenli Xiao, Guanzhi Wang, Dantong Niu, Fei-Fei Li, Guanya Shi, Jiajun Wu, Shankar Sastry, Yuke Zhu, Ken Goldberg, and Linxi "Jim" Fan. Cap-x: A framework for benchmarking and improving coding agents for robot manipulation, 2026. URL https://arxiv.org/abs/2603.22435.

Yipeng Gao, Lei Shu, Genzhi Ye, Xi Xiong, Ameesh Makadia, Meiqi Guo, Laurent Itti, and Jindong Chen. 3dcodebench: Benchmarking agentic procedural 3d modeling via code, 2026. URL https://arxiv.org/abs/2606.01057.

Xiaoshen Han, Junqiu Yu, Minghuan Liu, Yilun Chen, Xiaoyang Lyu, Yang Tian, Bolun Wang, Weinan Zhang, and Jiangmiao Pang. Re^{3}sim: Generating high-fidelity simulation data via 3d-photorealistic real-to-sim for robotic manipulation, 2026. URL https://arxiv.org/abs/2502.08645.

Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J. Lim. FurnitureBench: Reproducible real-world benchmark for long-horizon complex manipulation. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.041. URL https://roboticsproceedings.org/rss19/p041.html.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models, 2023. URL https://arxiv.org/abs/2307.05973.

Hanxiao Jiang, Hao-Yu Hsu, Kaifeng Zhang, Hsin-Ni Yu, Shenlong Wang, and Yunzhu Li. Phystwin: Physics-informed reconstruction and simulation of deformable objects from videos. *ICCV*, 2025.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues?, 2024. URL https://arxiv.org/abs/2310.06770.

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*, 2024. doi: 10.15607/RSS.2024.XX.120. URL https://www.roboticsproceedings.org/rss20/p120.html.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control, 2023. URL https://arxiv.org/abs/2209.07753.

Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. HOI4D: A 4D egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21013–21022, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Liu_HOI4D_A_4D_Egocentric_Dataset_for_Category-Level_Human-Object_Interaction_CVPR_2022_paper.html.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations, 2023. URL https://arxiv.org/abs/2310.17596.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots, 2024. URL https://arxiv.org/abs/2406.02523.

RLE-Bench contributors. Rle-bench: A qualifying exam for coding agents as robot learning engineers. https://github.com/RLE-Bench/RLE-Bench, 2026.

Yolo Y. Tang, Daiki Shimada, Jiayue Meng, Jing Bi, Pinxin Liu, Yicheng Wang, Yunzhong Xiao, Zhangyun Tan, Zeliang Zhang, Chao Huang, Susan Liang, Qianxiang Shen, Luchuan Song, Ali Vosoughi, Mingqian Feng, Melika Filvantorkaman, and Chenliang Xu. Bvb: Benchmarking agentic video understanding via programmatic reconstruction in blender, 2026. URL https://arxiv.org/abs/2609.15478.

Marcel Torne, Anthony Simeonov, Zechu Li, April Chan, Tao Chen, Abhishek Gupta, and Pulkit Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation, 2024. URL https://arxiv.org/abs/2403.03949.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models, 2023. URL https://arxiv.org/abs/2305.16291.

Lirui Wang, Yiyang Ling, Zhecheng Yuan, Mohit Shridhar, Chen Bao, Yuzhe Qin, Bailin Wang, Huazhe Xu, and Xiaolong Wang. Gensim: Generating robotic simulation tasks via large language models. In *International Conference on Learning Representations*, volume 2024, pp. 4890–4924, 2024a.

Yanan Wang, Renxi Wang, Yongxin Wang, Xuezhi Liang, Fajri Koto, Timothy Baldwin, Xiaodan Liang, and Haonan Li. Training and benchmarking code generation for physics-inspired animations, 2026. URL https://arxiv.org/abs/2602.10840.

Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation, 2024b. URL https://arxiv.org/abs/2311.01455.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments, 2024. URL https://arxiv.org/abs/2404.07972.

Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces, 2025. URL https://arxiv.org/abs/2412.14171.

Weirui Ye, Fangchen Liu, Zheng Ding, Yang Gao, Oleh Rybkin, and Pieter Abbeel. Video2policy: Scaling up manipulation tasks in simulation through internet videos, 2025. URL https://arxiv.org/abs/2502.09886.

Haoxiang You, Zeyu Shen, Yilang Liu, Zhicheng Zheng, Lihan Zha, Kashu Yamazaki, Mingtong Zhang, Suning Huang, Jiankai Sun, Qianzhong Chen, Lucy He, Kaiyuan Liu, Haoran Chang, Katerina Fragkiadaki, Dhruv Shah, Mac Schwager, Peter Henderson, Ian Abraham, and Canwen Xu. Embodiedswe: Coding agents for long horizon dexterous robotics, 2026. URL https://arxiv.org/abs/2609.27308.

Xinyu Zhan, Lixin Yang, Yifei Zhao, Kangrui Mao, Hanlin Xu, Zenan Lin, Kailin Li, and Cewu Lu. OAKINK2: A dataset of bimanual hands-object manipulation in complex task completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 445–456, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Zhan_OAKINK2_A_Dataset_of_Bimanual_Hands-Object_Manipulation_in_Complex_Task_CVPR_2024_paper.html.

Yifei Zhao, Xiangxin Zhou, Wenhao Yang, Jiaqi Tang, Pu Jian, Huanjin Yao, Jiarui Yao, Haowei Lin, Chunchao Guo, Zhuo Chen, Wenkai Lyu, Jianzhu Ma, Xueqian Wang, and Wenxi Zhu. Sceneactbench: Can agents act on the 3d scenes they see?, 2026. URL https://arxiv.org/abs/2607.22393.

A ENVIRONMENT AND AGENT INTERFACE

This section specifies the interface through which a coding agent constructs and submits a reconstruction to Video2World. A benchmark instance fixes an RGB video, a target embodiment, an execution backend, and a public task brief. The agent estimates the scene and supplies robot controls; the benchmark reconstructs the submitted environment and measures the resulting physical interaction. The interface described here is the benchmark configuration frozen on 24 September 2026. It is implemented using the scene, control, and validation facilities of VIDEO2SIM.

Instantiation of the reconstruction problem. The environment description η contains the reconstructed geometry, collision shapes, material and inertial properties, passive mechanisms, and initial object configuration, together with the estimated camera and robot placement required by the assigned interface. The target simulator $\mathcal{S}$ fixes the embodiment, admissible controls, and execution rules. The initial physical state s_0 is obtained from the submitted configuration using the initialization procedure specified below. Camera parameters determine rendered observations; they do not prescribe object motion. Thus, $P = (\eta, \pi)$ is an executable scene-and-control package interpreted by the benchmark's simulator, with the agent estimating the scene and behavior within a fixed execution contract.

Representation of robot behavior. In the implemented interface, the agent specifies a finite sequence of control references $U = (u_0, \ldots, u_{N-1})$ and a control interval Δt. Each reference is a Cartesian target or a vector of joint and gripper commands, as detailed in Table A2. The behavior π combines these references with the fixed controller for $\mathcal{S}$, which performs the applicable interpolation, inverse kinematics, actuator control, and contact-dependent waits. The general state-dependent notation $\pi(t, s_{\leq t})$ therefore includes feedback supplied by this controller; the submitted behavior is a control sequence, rather than an arbitrary policy callback. The coding agent may revise U after inspecting a construction rollout, but does not revise the frozen sequence during final evaluation.

State and execution time. The simulated state comprises robot and object configurations and velocities, together with the simulator state required to advance their dynamics. Robot actions a_t in the rollout describe physical execution and are distinct from the coding agent's tool operations during construction. The transition F_η is understood with the assigned simulator and controller fixed; a recorded transition can span multiple physics integration steps. Source-video frames, submitted control samples, and recorded simulation states consequently need not have equal counts or timestamps. Interpolation, initialization, contact-dependent waits, and terminal settling determine the executed clock. The declared control interval and recorded state timestamps, rather than the video frame index, determine the timing of the reconstruction.

A.1 PUBLIC INPUTS AND INFORMATION BOUNDARY

Episode inputs. The agent receives a local copy of the source video, its image dimensions and frame count, and the instructions for the assigned embodiment. Video inspection, frame extraction, geometric estimation, numerical optimization, and local simulation are available through ordinary coding tools. No particular perception or planning pipeline is prescribed. The agent must infer object geometry, metric scale, initial poses, camera parameters, and robot behavior from the permitted observations. For an egocentric recording, later measurements are expressed in the first camera frame after accounting for camera motion using the static scene.

Public simulator and robot information. Public resources specify the target simulator, robot geometry, joint limits, control conventions, and submission requirements. Agents infer the task objective, participating objects, semantic roles, and interaction sequence from the source video. No additional task descriptions, semantic role annotations, or task-specific success criteria are supplied. Source calibration, reference geometry, measured object trajectories, reference controls, and benchmark scores remain hidden.

Workspace and observations. The standard construction driver stages the video separately from the reference data and allocates episode-specific output and temporary directories. Instructions restrict episode-specific reads to the staged inputs and prohibit obtaining the source dataset's ground truth or calibration. Construction transcripts are retained and audited for accesses to protected sample

Table A1: **Agent-visible resources and withheld information.** Public resources are selected by the assigned configuration. Candidate observations concern the agent's own reconstruction.

Resource	Interface specification
Source observation	RGB video; resolution, frame count, and the sampling information stated in the task brief. No depth or measured source calibration.
Public specification	Target simulator and robot; simulator documentation, robot assets, joint limits, control conventions, and submission requirements.
Workspace tools	Shell execution; reading, writing, and editing files; filename and text search. Image-capable file reads expose inspected source frames or candidate renders.
Construction feedback	Program output and errors; package-validation errors; candidate physical rollouts, rendered observations, and available execution diagnostics.
Withheld information	Reference scene and object meshes, measured source trajectories, source robot logs, hidden camera calibration, reference submissions, and benchmark scores.

paths. These are workspace restrictions and an access audit; the audit alone does not establish operating-system isolation. A tool observation consists of the content actually returned to the agent: file contents, search results, execution output, errors, or inspected images. A generated image or video becomes a visual observation only when it is opened through an image-capable interface. Hidden benchmark scores are never included in the standard validation-repair loop.

Initialization of a benchmark episode. A reconstruction trial starts from the selected video and embodiment in the experiment manifest. Before launching the agent, the benchmark verifies the input video and public-resource versions, creates a fresh workspace and conversation, and supplies the fully instantiated task brief. The source video and released resources are staged separately from evaluation annotations; previous candidates and conversations are not reused. Thus, reproducing a trial fixes both the video bytes and the assigned configuration, together with the prompt revision, public assets, agent invocation, and evaluation version. The supported configurations in Table A2 describe the available interface; the selected experiment manifest determines which instances are run.

Execution environment for the model comparison. The seven OpenCode configurations in Table A7 use a benchmark-managed executor that additionally enforces filesystem and network restrictions. Public inputs are read-only, and each agent receives one CPU, 4 GiB of memory, and no GPU device access. Network connections are restricted to the model API proxy and the submission service. Web access, external plugins, human clarification, and delegated agents are disabled. These resource limits describe the coding process. In the current protocol, every configuration also provides physical rollout feedback during construction, as specified in Section A.5; lack of direct GPU access by the coding process does not remove this requirement. Final evaluation is performed separately with access to the withheld annotations. Astra and Fable use externally launched coding-agent sessions, whose configuration and delivery procedure are specified below.

Visual observations and media limits. The public frame sampler defaults to 16 evenly spaced frame indices, including both endpoints, with original timestamps retained. Images are resized to a maximum side length of 1,024 pixels and encoded as JPEG at quality 85. An extraction may contain 2–48 frames and must write to a new output directory. For the OpenCode configurations, each model request admits at most 16 images, 4 MB per image, and 20 MB of aggregate image data before base64 encoding; the serialized request is limited to 30 MB. Context management preferentially retains recent images and the first image of each source within this budget. Removed images are represented by visible placeholders and may be inspected again. Frame extraction and image observation are separate operations: only images actually returned by a read or attachment enter the model's visual context.

A.2 Simulation Configurations and Control Conventions

Table A2 summarizes the simulation configurations and control interfaces for the 222 benchmark instances. The two human-video configurations use the same 33 source clips with different target embodiments. All configurations are included in the main evaluation.

Table A2: **Registered candidate interfaces.** N is the number of reconstruction instances; the total is 222. Frequencies refer to submitted controls, not physics integration. For each floating hand, the seventh palm value records the open/closed convention; its twenty finger angles determine the actual posture.

Configuration	N	Simulator / robot	Submitted control
Furniture assembly	64	SAPIEN / Panda	TCP position, orientation, and gripper; 7 values at 5 Hz.
DROID tasks	20	SAPIEN / Panda with Robotiq 2F-85	TCP position, orientation, and gripper; 7 values at 5 Hz.
Human video to arm	33	SAPIEN / Panda	TCP position, orientation, and gripper; 7 values at 5 Hz.
Human video to hands	33	SAPIEN / paired Wuji hands	Two palm streams of 7 values and two finger streams of 20 angles; synchronized timestep.
Planar pushing	3	SAPIEN / xArm7 with pusher	TCP position and orientation at 5 Hz; seventh value unused.
RoboDojo	25	Isaac Sim / dual X5 or dual xArm7	14 or 16 joint/gripper values at 25 Hz.
Household manipulation	37	Isaac Sim / G1 with Omnipicker grippers	20 joint/gripper values at 20 Hz.
Rope and soft toy	7	MuJoCo / xArm7	7 arm angles and 1 gripper angle; declared control timestep.

Units, poses, and cameras. Distances are in metres, time in seconds, and joint and orientation angles in radians. Rigid poses use a translation and a scalar-first unit quaternion, $(x, y, z, q_w, q_x, q_y, q_z)$. Camera coordinates follow the OpenCV convention: $+x$ points right, $+y$ down, and $+z$ forward. A transform ${}^{A}T_{B}$ maps coordinates from frame B to frame A. The agent estimates the camera parameters requested by the configuration at the delivered image resolution. The rigid camera-frame profiles estimate focal lengths and principal point; RoboDojo assumes square pixels and a centred principal point and asks for the focal length. Resizing and letterbox padding must be accounted for when projecting onto the video.

For the camera-frame rigid and human-video interfaces, the declared camera-to-delivery transform is the identity. The evaluator applies the withheld registration between the source camera and the evaluation frame; this registration is not returned to the agent. Candidate self-simulation instead uses the agent's own estimate of the robot base pose. Native RoboDojo and household submissions explicitly estimate the camera-to-public-world transform, and evaluation uses that estimate rather than substituting hidden calibration. Joint commands remain in the named robot joint frames.

Single-arm Cartesian control. A submitted row is

$$u_k = (x_k, y_k, z_k, \alpha_k, \beta_k, \gamma_k, g_k), \qquad R_k = R_z(\gamma_k) R_y(\beta_k) R_x(\alpha_k). \tag{5}$$

The position denotes the tool centre point (TCP); for the parallel-gripper interfaces this is the midpoint between the fingertip pads, with the approach direction along local $+z$ and the opening direction along local $+y$. A gripper command of one means open and zero means closed. The first row specifies the initial tool configuration. The public brief recommends successive translations no larger than 3 cm and rotations no larger than 0.3 rad, and a stationary terminal segment. These are motion-planning recommendations, distinct from the runtime's numerical validity checks.

For FurnitureBench, rigid DROID, and planar pushing, the benchmark registers the camera-frame submission using the withheld source calibration and executes the controls with the assigned robot at its evaluation placement. Inverse kinematics produces joint targets, with four interpolated joint-control updates between consecutive submitted poses. For the prescribed 5 Hz input, joint control is 20 Hz and physics integration is 100 Hz. Reset uses seed zero. An initially open gripper is followed by 0.5 s of settling. An initially closed stream omits this settling and performs grasp initialization, with closing preparation of up to 2 s for the Panda or 1.5 s for the Robotiq. Closing transitions can insert additional waits: up to 1.5 s for the Panda and 1.2 s for the Robotiq, with earlier termination when the respective contact condition is met. The final pose is measured after a 1 s settling period. These waits belong to the executed clock and are retained in the rollout diagnostics. The human-video arm configuration uses its separate 300 Hz contact executor and also includes 1 s of terminal settling.

The rigid benchmark uses frictional holding with configuration-specific contact approximations. For an initially held object, it permits a surface-alignment correction of at most 3 cm before closure. Applicable furniture insertion tasks use an evaluator-owned retention approximation: a released part satisfying the predefined seating geometry is fixed at its attained pose to represent a threaded or tight fit. Ordinary DROID placement disables this retention rule. These operations are fixed properties of the evaluator, not additional actions available to the agent. In DROID, the gripper closes kinematically before first contact and then uses force-limited control; its closing rate is derived evaluator-side from the source gripper trace, with a minimum increment of 0.02 rad per control update. This trace is not exposed during reconstruction.

The public local replay tool supports candidate-only inspection using the agent's estimated robot and camera geometry. It is distinct from the registered benchmark evaluator: its 5 cm inverse-kinematics rejection threshold and 180 s action limit are local replay checks, not universal benchmark admission thresholds. Local replay success therefore does not certify the benchmark outcome.

Synchronized dexterous control. Each Wuji hand has six wrist degrees of freedom and twenty finger joints. The submission provides a palm-pose stream and a finger-angle stream for each side, with equal lengths and a shared timestep that is a positive integer multiple of $1/300$ s; the task brief gives $1/30$ s as an example. An unused hand still receives a stationary stream. Finger angles are ordered by thumb, index, middle, ring, and little finger, with four joints per digit, and must satisfy the public robot limits. Palm $+z$ points along the extended fingers and $+x$ toward the palm's curling direction; the thumb lies on $+y$ for the right hand and $-y$ for the left.

Both hands evolve under force-limited dynamics at 300 Hz. Wrist forces are capped at 200 N per translational axis, and wrist torques at 100 N,m per rotational axis; finger effort and velocity limits follow the robot description. The default initialization uses an open posture and a palm position displaced backwards by 8 cm, followed by a shared 1.2 s approach. A submitted initialization offset may have magnitude at most 25 cm. Both hands then follow the same action clock, without independent adaptive waits, and the rollout ends with 1 s of settling. Objects are held through contact and friction. Commanded palm motion is not exact object motion, and contact from either hand is considered when determining release. The hand validity checks allow at most 6 mm of robot penetration, 15 mm of object displacement during initialization, 4 mm of nonadjacent self-overlap diameter, and 0.001 rad of finger-joint limit excess. Joint-speed exceedances are recorded as diagnostics; they do not independently invalidate the rollout.

Native joint control. RoboDojo accepts the left-arm angles, left gripper, right-arm angles, and right gripper, in that order. Dual X5 has six arm joints per side (14 values total); dual xArm7 has seven (16 values total). The interval is 0.04 s and grippers use zero for closed and one for open. The household interface instead orders seven left-arm angles, seven right-arm angles, waist lift, waist pitch, head yaw, head pitch, and two gripper commands (20 values total), at an interval of 0.05 s. Waist lift is a distance; the other articulated coordinates are angles. Here grippers use zero for open and one for closed. Only the outer gripper joints are driven; mechanical follower joints retain their native coupling. RoboDojo integrates physics at 250 Hz (ten steps per command) with reset seed seven. The 35 rigid household cases integrate at 60 Hz (three steps per command), while the two wallet cases use 120 Hz (six steps per command). The household executors append a task-specific stationary terminal tail of either two or forty command intervals (0.1 s or 2 s); the wallet cases use 2 s. One initialization physics step precedes the recorded household trajectory.

In the current household brief, the first-frame camera pose must be consistent with the public head-camera mount and the agent's inferred initial waist and head configuration. If ${}^{W}T_{H}$ is the resulting USD camera transform expressed using column vectors, then

$$ {}^{W}T_{C} = {}^{W}T_{H}\ \mathrm{diag}(1, -1, -1, 1). \tag{6} $$

USD row-vector matrices are transposed before this conversion. The scene remains in the first camera frame even when the head subsequently moves. This requirement entered the public brief on 23 September 2026; it is not a claim about the prompts used for earlier submissions. Structural validation checks the submitted rigid transform and joint limits; it does not independently certify this forward-kinematics consistency.

Articulated-gripper MuJoCo control. The rope and soft-toy configurations use the supplied xArm7 collision model, including six mechanically linked gripper joints, five coupling equalities, and four exclusions between adjacent links. The agent estimates the robot base placement but preserves these internal mechanics. The TCP is 0.172 m from the final arm link. Each action contains seven arm angles and the gripper drive angle: zero is open and 0.85 rad is closed. The drive command changes at no more than 2 rad/s. The public brief specifies a physics timestep of $50\,\mu$s and a coupling constraint time constant of 0.1 ms with damping ratio one. The runtime rejects a coupling residual above 0.001 rad at any physics step. The declared control interval must lie between 1 ms and 1 s; submissions contain at most 20,000 rows and at most 180 s of controls. Executed states are sampled at 20 Hz. Manipulated objects are passive rigid bodies, articulated chains, or deformable elements; controls actuate only the supplied robot.

Runtime versions. The inspected rigid execution environment uses SAPIEN 3.0.3 and ManiSkill 3.0.1 with Python 3.11.7. The isolated rope and soft-toy environment uses MuJoCo 3.11.0 with Python 3.11.7. The native robot environment uses Isaac Sim 5.1.0.0. These are backend-specific environments, rather than a single shared installation. Robot assets, controller implementations, native runtime dependencies, and the selected evaluation version are bound to the run's reproduction record; library dependency lower bounds do not identify a reproducible runtime.

A.3 Submission, Validation, and Execution

Submission contents. A submission is a self-contained scene-and-control package. It declares the assigned embodiment, object identities and roles, geometry and collision geometry, physical properties, initial poses, camera model, control convention, and timestep. All referenced candidate geometry must resolve to local package assets. A reconstruction report states the observations supporting metric estimates, modeling approximations, and remaining ambiguities. Annotated source frames provide evidence for estimates such as object dimensions and contact timing. Table A3 gives the concrete interchange files; these filenames are serialization conventions, not a prescribed reconstruction method.

Table A3: **Submission artifacts.** T denotes source-video frames and N submitted control rows; these lengths need not agree.

Artifact	Required content
`protocol.json`	Version 3.0 manifest: target configuration, scene or scene-file reference, camera, robot, action-file references, timestep, and provenance.
`actions.npy`	Finite control matrix with the embodiment-specific width in Table A2; Cartesian streams are stored as 64-bit floating-point arrays.
Additional hand arrays	For paired hands, a second $N \times 7$ palm stream and two $N \times 20$ finger-angle streams, with four distinct file references.
`scene.json` / `scene.xml`	Native Isaac submissions use a separate scene description; MuJoCo submissions use an MJCF scene with the supplied robot and explicit object, robot, and target roles.
`assets/`	Locally referenced visual and collision geometry. Native mesh submissions provide vertices and face indices, with metric scale incorporated into the vertices.
`expected/` `obj_poses.npy`	Camera-frame $T \times 7$ object-pose estimates for the camera-frame rigid and human-video profiles. These are declarations, not commands; native joint and MuJoCo interfaces do not require them.
`source/`, `report.md`	Input video, annotated evidence frames, and reconstruction rationale with uncertainty and approximation disclosures.

Scene semantics. Manipulated objects are dynamic; receiving objects and supports are represented explicitly. Static and dynamic geometry are distinguished, and a receptacle's collision model must preserve its cavity. The shared rigid schema specifies primitive half-extents, whereas the native Isaac scene format specifies full extents; the conversion must not introduce a factor-of-two scale error. Object names identify semantic roles in the rigid interfaces. Native tasks use the supplied semantic identifiers, while MuJoCo submissions identify disjoint robot, manipulated-object, and target subtrees.

Doors and drawers require passive revolute or prismatic joints, respectively, with the joint axis and anchor expressed in the parent object's local frame. Initial joint displacement is zero; limits are in radians for revolute joints and metres for prismatic joints.

Structural validation. The validator checks that required files exist and that geometry, poses, and action arrays satisfy the assigned contract. Configuration-specific checks include object and receiver presence, finite arrays, synchronized hand streams, dimensions and command rates, admissible joint limits, local asset references, and valid rigid transforms. Native scene checks additionally verify positive dimensions and mass, normalized quaternions, collision participation, and consistency between declared sizes and mesh bounds. A structurally valid package may still fail to initialize, contain unreachable controls, or fail the task; validation is not a success oracle.

Physical execution. Evaluation initializes the scene from the submitted package and executes its own controls through the registered robot interface. The benchmark owns simulation stepping and state recording. Candidate object-pose predictions do not drive bodies, and candidate instructions cannot replace the measured rollout with a declared trajectory. Direct object animation, object actuators, artificial grasp attachments, and post-initialization object-state overrides are outside the candidate control interface. Passive mechanisms required by an object's physical structure, such as a door hinge or rope joints, are part of the scene model. Native executors record actual robot and object states and, where supported, contacts, constraint residuals, and physical-validity diagnostics. Candidate-only inspection uses the declared geometry and camera where supported; evaluation renders follow the configuration-specific camera and registration policy described above, including native mounted cameras.

Build and outcome separation. A missing or invalid candidate package is a method failure. Successful parsing, a completed simulator process, and the agent's declared completion status do not imply Task Success. Task judgments use the executed states and the benchmark's evaluation specification. Conversely, an execution that completes and produces valid records can fail the task while remaining evaluable. Runtime or dependency failures attributable to evaluation infrastructure are recorded separately and require a valid rerun before a task conclusion is drawn. The evaluator's recorded failure category distinguishes these cases; no hidden score is used to choose a better intermediate submission.

A.4 Agent Tools and Public Operations

Coding tools provide the agent's interaction with the construction environment. They support video inspection, numerical analysis, geometric modeling, and writing scene and control artifacts. Table A4 summarizes these capabilities and the feedback available to the agent. The reconstruction procedure is unrestricted within the task's observation and resource constraints. OpenCode selects the editing interface for the configured model. Fable exposes its native shell, read, write, and edit tools, while Astra uses the coding tools of its pinned Codex CLI. Exact invocation schemas accompany the reproducibility material.

The shell tool defaults to a 120 s timeout, which may be overridden for individual commands. Text output is ordinarily limited to 2,000 lines or 51,200 bytes; oversized shell output is retained in a local file that the agent can subsequently read. Text reads default to 2,000 lines, with a 51,200-byte limit and a 2,000-character limit per line. Filename and text searches return at most 100 results. These observation limits are distinct from the episode's total construction allowance.

The benchmark provides operations for observing the video, inspecting physical execution, and managing candidate reconstructions (Table A5). Frame extraction exposes visual evidence; physical replay provides feedback on the reconstructed interaction; validation checks structural requirements; checkpointing preserves a candidate during refinement; and submission requests final delivery. None of these operations returns reference annotations or benchmark scores. A checkpoint or submission is validated on a protected copy, so subsequent edits to the working candidate cannot change a previously accepted reconstruction.

The agent is informed whether preservation or submission succeeded, whether an error arose from the candidate or the validation service, and how many submission opportunities remain. Only a complete, internally consistent copy can become an accepted candidate. Validation establishes compliance

Table A4: **Coding capabilities available to the agent.** All operations are restricted to the permitted workspace and public resources. Model-specific tool definitions are supplied by the pinned agent framework.

Capability	Agent-specified input	Agent-visible feedback
Shell execution	Program or command; optional execution limit	Standard output, execution errors, exit status, and timeout information.
File and image reading	File or directory; optional text range	File contents, directory entries, or an image observation.
Filename search	Filename pattern and search location	Matching file paths.
Text search	Text pattern and search scope	Matching passages and file locations.
File creation	File location and contents	Confirmation, diagnostics, or a write error.
Text replacement	File and requested replacement	Modification result or an error.
Patch editing	Additions, changes, or deletions	Results of the requested file modifications.
Task bookkeeping	Task descriptions and completion states	Updated local task list.

Table A5: **Benchmark operations during construction.** The interface separates visual and physical observations, structural feedback, and candidate delivery. An accepted submission terminates generation; successful validation alone does not.

Operation	Input	Agent-visible feedback	Effect on the episode
Frame extraction	Source video and sampling specification	Frame references, timestamps, and image metadata	Makes frames available for inspection; the candidate is unchanged.
Physical replay	Candidate scene and robot controls	Rendered rollout and available execution diagnostics	Supports inspection and revision of the reconstruction; does not submit or score it.
Validation	Candidate reconstruction and task configuration	Missing artifacts and violations of the public submission requirements	Construction continues; no candidate is submitted.
Checkpointing	Candidate reconstruction	Validation feedback and confirmation of preservation	A valid candidate is retained as a fallback during the permitted refinement.
Submission	Candidate reconstruction	Acceptance or validation errors	Acceptance freezes the final candidate and ends generation; an invalid submission consumes one opportunity.

with the public submission requirements; it neither determines physical task success nor changes the candidate selected for evaluation.

A.5 Construction Budget and Feedback

Physical feedback during construction. All benchmark configurations provide access to physical rollout feedback, including rendered observations and execution diagnostics. Agents may inspect their reconstructed interactions alongside the source video and revise both the environment and robot behavior within the construction budget. Feedback is restricted to the agent's own reconstruction; reference annotations and benchmark scores remain unavailable. Final evaluation executes the selected, frozen reconstruction without further agent revision.

Construction feedback is restricted to the candidate's own scene, controls, rendered observations, and available execution diagnostics. It does not expose reference geometry, source calibration, reference controls, or hidden evaluation scores. A rollout becomes visual feedback when its rendered observations are supplied to or opened by the agent; merely creating a video file is insufficient. The agent compares these observations with the permitted source video. This construction loop is distinct from final evaluation, which executes the selected, frozen package and computes the benchmark outcomes without further agent revision. Candidate replay and final evaluation retain their respective camera and registration conventions described in Section A.2.

Budget semantics. Construction is bounded by the configured time and, where applicable, API-cost allowance. There is no fixed episode-wide tool-call or model-turn budget. Coding, validation, and rollout inspection take place within construction; the separate evaluation allowance is used after the candidate is frozen. Submission opportunities constrain delivery attempts and do not limit the number of analysis, editing, or inspection operations. Likewise, a per-command timeout or per-response output limit is not an episode-wide tool-call budget. The model-specific stopping and candidate-selection rules below determine which package is retained when construction ends.

Scope of the construction protocols. The following three paragraphs document the legacy reference two-round authoring driver. The multi-model experiments additionally use a managed executor with explicit submissions and protected checkpoints, and externally launched native coding agents. Their respective stopping and delivery rules are specified below; the two-round default is not a model-turn limit for those experiments.

The reference authoring driver allows at most two construction rounds: the initial task brief and, if necessary, one repair round containing the concrete validation errors. Both rounds belong to one attempt; the default adapter resumes the same agent session for the second round. Validation is performed after each round, and construction stops at the first validation-passing submission or after the second round. The package left by this process is handed to evaluation, including an invalid final package if repair fails. A run producing no package is retained as a failed build. A fresh experimental attempt must use a new run location, because the outer runner treats an existing submission as completed work.

The configured outer limits are 10,800 s for construction and 5,400 s for evaluation, with timeout status retained. These limits bound the respective runner processes; there is no additional fixed token budget or tool-call count in the standard interface. The default author adapter requests Claude Opus 5 and exposes the the tool capabilities in Table A4; the benchmark also permits an explicitly configured external author command. The requested model, adapter, command, and effective limits therefore belong to each run record. These defaults describe the current interface and do not retroactively establish the settings of every historical result.

Between rounds, feedback concerns the submitted package's validity. During a round, the agent can use available candidate-only simulation and inspection tools to refine its reconstruction. Such outputs describe the consequences of its own scene and controls. Reference-based geometric errors, trajectory errors, Task Progress, and Task Success are computed after construction and are not returned as repair feedback.

Managed construction protocol. The multi-model executor uses explicit submission rather than the two-round return-and-validate procedure above. Each episode receives one generation attempt,

a cumulative construction allowance of 10,800 s, and a $60 observed API-cost threshold. Delayed usage reporting may cause the billed cost to exceed the threshold. There is no fixed total number of model turns or tool calls. A normal agent return without delivery can resume the same conversation and workspace; three cumulative empty turns, defined as turns with neither a tool call nor a candidate, terminate the attempt. Resource exhaustion and repeated interaction without progress may also stop construction. A terminal method failure does not automatically trigger a fresh generation attempt.

Table A6: **Construction schedule for benchmark-managed episodes.** Times are cumulative within one attempt. The first milestone is a prompt instruction; the generation and publication deadlines are enforced by the executor.

Elapsed time	Construction and delivery policy
About 30 min	Complete an initial evidence-based candidate; validate it and repair missing or malformed artifacts.
120 min	Restrict remaining work to completing and repairing delivery.
150 min	Close the optional refinement allowance.
165 min	Stop generation and new delivery requests; select the latest valid protected checkpoint if no final submission has been accepted.
180 min	Complete process termination, integrity verification, and publication of the selected candidate.

Submission allowance and candidate selection. After the first successful public validation, the agent may submit immediately or save a checkpoint before one bounded refinement. Up to three submission attempts are permitted. An invalid package or invalid submission path consumes an attempt; a failure of the executor's validation service does not. Saving a checkpoint neither consumes nor replenishes submission opportunities. Once a valid checkpoint exists, only one optional refinement checkpoint is permitted, before the 150-minute cutoff.

A successful explicit submission ends generation and determines the package evaluated, without requiring a final natural-language response. If generation reaches the 165-minute deadline, the latest valid executor-saved checkpoint is selected. An invalid revision cannot replace that checkpoint. Unsaved workspace files and claims in an agent response are not eligible substitutes. The selected package is frozen and verified before evaluation, and hidden scores never determine which candidate is chosen. An episode without a valid delivery, or with an explicit failure or infeasibility declaration, is retained as a method failure. Process-cleanup and package-integrity failures are classified separately as executor faults.

Freezing the reconstruction. The selected candidate fixes both η and π: scene assets, physical parameters, initial configuration, camera estimates, controls, and their timing are preserved together. Subsequent workspace edits and construction rollouts cannot alter that candidate. Evaluation and any infrastructure rerun use the same package under the specified evaluator. A timeout without an eligible submission or protected checkpoint yields a build failure; it does not create a candidate from unfinished workspace files. Candidate selection uses the delivery protocol, without comparing hidden scores across intermediate reconstructions.

Timeouts and infrastructure failures. Model-service connection failures receive up to three request retries. Persistent API failures enter a recovery queue with a 600 s cooldown and at most three additional retests; the conversation, workspace, and cumulative construction budgets are preserved. Episodes that exhaust service recovery are held for inspection rather than assigned a task outcome. A validator infrastructure failure does not consume a submission or refinement opportunity.

Evaluation has a separate cumulative allowance of 5,400 s for reference execution, candidate simulation, and metric computation. A pre-evaluation GPU probe has a separate 90 s timeout. A failed probe requeues the unchanged package without consuming its evaluation allowance and removes that GPU from admission; consecutive failures on two distinct GPUs pause the evaluation pool. Evaluator timeouts, unavailable dependencies, missing outputs, and corrupt execution records are recorded as infrastructure failures. They require a valid evaluation of the same frozen candidate before a task outcome is assigned. An infrastructure rerun does not regenerate the reconstruction or select a

different workspace version. Candidate-caused invalidity and a physically executed failure remain method outcomes.

Reproduction record. Reproducing an interface run requires the source-video identity, assigned configuration, fully substituted prompt and public handout, model and adapter invocation, effective budgets, submitted package, construction transcript, and physical execution record. The construction record also identifies the candidate versions replayed, the rendered observations actually supplied to the agent, and the feedback preceding each revision. Versioned source and asset hashes identify the robot, simulator adapters, validation rules, and evaluation snapshot; recorded timesteps, initialization, seeds, and state timestamps identify the executed dynamics. The prompt supplement records the source snapshot used to prepare this section. Because simulator state, rendering, and solver behavior depend on the runtime, matching an interface specification alone is not evidence of identical trajectories across unverified installations.

A.6 Agent Initialization and Model Configurations

The experiment configuration identifies the model, agent framework, provider route, construction budget, public interface specification, and evaluation version. Seven model configurations use OpenCode 1.18.29 under the managed construction protocol. GPT-6 Astra uses Codex CLI 0.154.0-alpha.6.2, and Fable-5.1 uses Claude Code 2.1.276. Table A7 gives the exact model identifiers and inference settings. Provider fallback is disabled for the OpenCode configurations, and their per-response output limit is 32,000 tokens including reasoning. The native coding agents retain the output-length behavior of their pinned CLIs.

Table A7: **Model configurations.** The first seven rows use OpenCode and the model-identifier prefix `openrouter/`. A dash denotes an unspecified reasoning-effort setting.

Model	Model identifier	Provider	Effort
DeepSeek V4.1 Flash	`deepseek/deepseek-v4.1-flash`	`novita/fp8`	–
GLM-5.3 Flash	`z-ai/glm-5.3-flash`	`novita/fp8`	–
Kimi K3	`moonshotai/kimi-k3`	`moonshotai/mxfp4`	–
GPT-5.6	`openai/gpt-5.6-sol`	`openai`	–
Gemini 3.8 Flash	`google/gemini-3.8-flash`	`google-vertex/global`	High
Claude Opus 5	`anthropic/claude-opus-5`	`anthropic`	–
Qwen3.8 Max	`qwen/qwen3.8-max-0902`	`alibaba`	–
GPT-6 Astra	`gpt-6-astra`	Codex	Ultra
Fable-5.1	`claude-fable-5-1`	Claude Code	Max

Initialization of a single-video trial. A trial is instantiated from the selected video, target embodiment, public interface specification, and model configuration. Before the first model call, the benchmark stages the permitted inputs, initializes empty analysis and output directories, and supplies the fully substituted task brief to a fresh agent session. The same conversation and workspace persist throughout construction. The selected candidate is subsequently transferred to the separate evaluation process, which has access to the hidden annotations. Reproductions use the same input and configuration versions and begin from a new session. Executable launch procedures and machine-specific setup are provided with the reproducibility material.

Native coding-agent sessions. Astra and Fable operate on exported public task packages containing the video, released resources, and instantiated brief. Each task package binds the source video, target embodiment, released resources, and prompt revision. Their external supervisors provide filesystem restrictions, API access, time accounting, and a local submission service. The native model is launched with the settings in Table A7; framework-specific runtime settings accompany the experimental configurations. The completed package, submission receipt, session record, and transcript are imported into the benchmark evaluation queue.

These sessions use a 180-minute hard limit. The selected result is the last successful submission, or a preserved checkpoint on timeout. Their local submission service does not by itself enforce the managed executor's complete staged-deadline and refinement policy. Accordingly, the 165-minute generation cutoff and executor-managed counters in Table A6 apply to the OpenCode runs. Agent-framework differences and Astra's per-instance reasoning settings remain part of the reported experimental configuration. The model configurations are associated with their own instance manifests and prompt revisions; the current interface inventory is not substituted for those experimental manifests.

A.7 Task Prompts

The task brief combines configuration-specific instructions, the applicable submission contract, and episode-specific information, including the video location, image dimensions, frame count, output location, and public resources. The panels below summarize these instructions in publication notation, using mathematical symbols and placeholders for episode-specific values. The submission schema is specified in Table A3. Panels labeled verbatim preserve the recorded wording apart from the indicated substitutions.

Construction feedback and session control. All evaluated model configurations provide access to candidate-only physical replay, rendered observations, and execution diagnostics. The shared construction instruction below directs agents to inspect their simulated interactions alongside the source video and revise the reconstruction as needed within the construction budget. Reference annotations and benchmark scores remain unavailable during construction.

For benchmark-managed OpenCode runs, candidate delivery requires explicit submission or selection of a protected checkpoint under the prescribed stopping rules; returning from an agent turn does not itself deliver a candidate. The validation-repair prompt applies to the two-round reference authoring driver. The managed-session continuation messages and executor-status template apply to the benchmark-managed executor.

Shared construction instruction: physical rollout feedback

Execute the reconstructed scene with your proposed robot controls and inspect the rendered physical rollout alongside the source video. Use the observed interaction and execution diagnostics to revise the scene and controls as needed within the construction budget. If execution fails, use the reported error to repair the candidate and replay it. Package validation alone does not replace physical execution and visual inspection. Feedback concerns your own reconstruction; reference annotations and benchmark scores are unavailable. Submit the completed candidate through the assigned delivery interface. The selected package is fixed for subsequent evaluation.

Reconstruction brief: rigid scene and Cartesian control

You are given one RGB video V with resolution $W \times H$ and T frames, and the public specification of the assigned robot. Reconstruct the manipulated object, the receiving object or interacting surface, and the surrounding geometry needed to reproduce the observed task. Infer the task from the video. Source robot logs, depth, camera calibration, reference object geometry, and reference trajectories are not available.
Estimate object shape, size, initial pose, and camera intrinsics from the images. Preserve hollows and openings that participate in contact. Represent static objects and supports separately from manipulated objects. Express geometry and motion in the prescribed camera frame; for a moving camera, use the first frame as the fixed coordinate system.
Supply your own absolute TCP poses and gripper commands at 5 Hz. Positions are in metres, rotations follow $R_z R_y R_x$, and one denotes open while zero denotes closed. The first row specifies the initial TCP pose. Re-plan the motion for the assigned embodiment while preserving the demonstrated object interaction and outcome. Keep successive displacements within approximately 3 cm and rotations within 0.3 rad, and finish at rest.
Deliver the scene, camera estimate, controls, object-motion estimates at the video timestamps, local geometry assets, annotated evidence frames, and a report of observations, assumptions, and uncertainty. Check that the grasp and release commands agree with the reconstructed object and goal locations. Run the assigned package validator and repair any reported errors. Write only in the designated output and temporary directories. Do not obtain the source dataset's ground truth or calibration.

Embodiment supplement: human video to paired hands

Reconstruct the manipulated object, visible support, and relevant receiving objects, and perform the interaction with a left and a right Wuji hand. Both hands are available even if only one human hand is visible. An unused hand must receive stationary commands.
Supply synchronized palm poses and twenty finger angles for each hand. Both sides share one timestep, a positive multiple of $1/300$ s. Use the public palm axes, finger ordering, joint limits, and force-limited control specification. Finger angles determine posture; the palm stream's gripper indicator is informational. Express all poses and object-motion estimates in the first camera frame.
Model a drawer with a passive prismatic joint and a door with a passive revolute joint. Preserve collision clearances and cavities. Do not attach objects to robot links or prescribe object motion during simulation. Include both hands in release reasoning and report uncertainty in the proposed task completion.

Reconstruction brief: native dual-arm environments

You are given an RGB video and the public specifications of the assigned simulator and robot. Infer the demonstrated task, participating objects, and their roles from the video. Reconstruct the visible geometry, supports, scale, and initial poses, and provide your own joint commands. Make manipulated objects dynamic and preserve physical collisions. Use only the staged inputs, public resources, and feedback from your own reconstruction.
Express scene poses in the OpenCV camera frame and provide a rigid transform from that frame to the public robot world. Joint commands are expressed in the robot's named joint frames. For RoboDojo, estimate the full optical pose and focal length; supply the assigned 14- or 16-value controls at 25 Hz, with zero meaning closed and one open. For the household robot, supply the 20-value arm, waist, head, and gripper controls at 20 Hz, with zero meaning open and one closed. Derive the first-frame camera transform from your estimated initial waist/head configuration and the public camera mount, as in Eq. 6.
Deliver the manifest, scene description, finite action matrix, local geometry assets, and reconstruction report. Object motion must result from robot actuation and simulation. An expected object trajectory is neither required nor executed. Validate the package against the assigned submission requirements. Do not access hidden source layouts, source camera calibration, reference controls, or reference execution records.

Reconstruction brief: rope and soft-toy environments

Reconstruct the video as a native MuJoCo scene using the supplied xArm7 robot and articulated gripper. Read the public robot description and preserve its internal body geometry, actuators, mechanical couplings, and adjacent-link collision exclusions. Estimate the base placement from the first camera frame.
Model the manipulated object with passive bodies, an articulated chain, or deformable elements, and model the receiver with physical colliding surfaces. Set gravity consistently with the first camera frame. Supply eight joint controls in radians, with gripper angle zero for open and 0.85 for closed. Use the specified $50\,\mu$s physics timestep and preserve the gripper coupling and command-rate limits.
Identify the robot, manipulated object, and receiving structure as disjoint physical roles. For rope routing, preserve the connected rope and the open channel between the receiver walls. Use no object actuators, artificial attachments, motion-capture drivers, or object-state overrides. Include the video, scene, controls, assets, and a report of uncertainties. Run structural validation and physical replay. A completed replay is not a benchmark task judgment. Read no hidden sample metadata, calibration, or reference trajectory.

Validation-repair prompt (verbatim apart from substitutions)

Your package at *[output location]* does not validate:
– *[validation errors]*
Fix it (same rules apply) and re-run the validate command.

Managed-session continuation: no tool call or candidate (verbatim)

The previous turn ended without a tool call or delivery. Continue the task from the existing workspace and conversation; inspect existing artifacts, do not repeat completed steps; when a package passes public validate, call `python public/v2_tools.py submit <pkg_dir>`.

Managed-session continuation: candidate or checkpoint present (verbatim)

No explicit submission received. Call python public/v2_tools.py submit <pkg_dir> to deliver within your remaining allowance.

The executor also appends a status message to the latest tool observation, or supplies it as a system prefix before the first tool result. The message reports elapsed time, remaining generation time, construction phase, the existence of a valid checkpoint, remaining submission and refinement opportunities, and media usage. The following template preserves the field order and fixed wording; angle-bracket fields are populated from the episode state.

```
[executor-status/1]
{"revision":"executor-status/1","elapsed_s":<seconds>,"remaining_s":<
    seconds>,"stage":<phase>,"valid_checkpoint":<boolean>,"
    submit_remaining":<count>,"refinement_remaining":<count>,"
    media_remaining":<media-status>,"removed_images":<count>}
Removed images are visible placeholders. Re-read an image to attach it
    again within the remaining media budget.
[/executor-status/1]
```

Because agent frameworks differ in their internal tool use, context management, and candidate-selection behavior, comparisons in this paper should be interpreted as comparisons among complete coding-agent configurations rather than controlled estimates of isolated backbone-model capability.

B Benchmark Construction and Reference Annotations

Benchmark construction establishes the correspondence between a source demonstration and the quantities evaluated in simulation: task-relevant geometry, object motion, and interaction outcomes. Each video is paired with a target robot and simulator, together with hidden annotations defining these quantities. Annotation provenance distinguishes source measurements, estimates fitted to observations, and reconstructed physical targets. The human-assisted reference (HAR) is constructed using these resources and evaluated against the same applicable targets as coding-agent submissions.

B.1 Data Sources and Video Selection

We collect videos from FurnitureBench, DROID, RoboDojo, HOI4D, HOT3D, DexYCB, OakInk2, an in-house household collection, and reconstructed digital-twin demonstrations of planar pushing, rope routing, and soft-toy packing. These sources vary in demonstrator embodiment, viewpoint, object properties, and interaction structure. RoboDojo supplies simulation-generated demonstrations; the other groups include recorded real-world interactions. The digital-twin group contains three planar-pushing, five rope-routing, and two soft-toy clips, using real videos with reconstructed physical targets.

The benchmark contains 189 distinct source video clips and 222 reconstruction instances across 39 task families (Table A8). Each instance pairs a source video with a target simulator, robot embodiment, and control interface. The 33 human-demonstration clips each have an arm and a bimanual-hand configuration. A clip is a selected temporal window, not necessarily an independent acquisition episode: the 64 furniture clips come from 46 original episodes, and some human clips share a longer recording. All 222 instances are included in the main evaluation.

Selection and coverage. Selection retains the initial configuration, the task-defining interaction, and the available terminal context. A visible task label alone is insufficient: the clip is reviewed to establish what interaction actually occurs, including whether it begins with an object already held or ends before release. Furniture windows are capped at eight per assembly step and two per leg-insertion step, covering 25 assembly steps. The retained-window ordering uses reference execution outcome, then agreement with the official terminal assembly pose, and finally a deterministic sample identifier. This construction choice favors reconstructable windows within the cap and is part of the selection policy. Unsuccessful selected references remain in the inventory. Human-video and

Table A8: **Composition of the construction inventory.** Clips count distinct staged video windows; instances count clip–configuration pairs. Embodiments include the end effector. Shared recordings and reuse across embodiments are retained in the provenance.

Video source	Clips	Instances	Target embodiment
FurnitureBench	64	64	Panda
DROID	20	20	Panda/Robotiq (18); cloth gripper (2)
RoboDojo	25	25	Dual X5 or dual xArm7
HOI4D	9	18	Panda; paired Wuji hands
HOT3D	7	14	Panda; paired Wuji hands
DexYCB	11	22	Panda; paired Wuji hands
OakInk2	6	12	Panda; paired Wuji hands
In-house household	37	37	G1 with Omnipicker grippers
Reconstructed twins	10	10	xArm7 with pusher or articulated gripper
Total	189	222	Source-specific configurations above

Table A9: **Evidence used to construct the hidden annotations.** The middle column identifies available source information; the last identifies the processing and modeling needed for benchmark evaluation. Reconstructed twins supply modeled targets.

Source	Available source evidence	Benchmark annotation work
FurnitureBench	CAD, tags, calibrated cameras, and robot logs.	Register part poses; fit grasp offsets to bridge occlusions during verified holding.
DROID	Calibrated RGB-D, multiple views, and robot kinematics.	Fit object geometry and poses; reconstruct supports and cavities; track visible cloth surfaces.
RoboDojo	Native assets and synchronized simulation states.	Export geometry, cameras, and trajectories from the same execution as the video.
HOI4D	Instance CAD, object boxes, object poses, and camera trajectories.	Compose object and camera poses in a gravity-aligned frame; model supporting surfaces.
HOT3D	Object CAD and tracked object/camera poses.	Register trajectories; identify supports and static/dynamic roles.
DexYCB	YCB meshes, calibrated views, and object/hand poses.	Convert pose units and frames; estimate support height from initial mesh bottoms.
OakInk2	Task programs, part scans, motion capture, and camera calibration.	Select primitive segments; infer articulation from relative part motion; construct collision approximations.
Household	Registered assets, RGB-D observations, and source recordings.	Estimate geometry and poses or visible surfaces; retain view dependence and uncertain intervals.
Reconstructed twins	Reconstructed geometry and rigid or particle trajectories.	Freeze the modeled target and its provenance; adapt the simulation to the target robot.

household admissions follow documented clip-level review; clip boundaries, source identifiers, and exclusions are retained.

Source-specific annotation procedures. Auxiliary source information supports hidden annotations and is unavailable as episode evidence to the evaluated coding agent. Table A9 distinguishes information supplied by each source from the annotation work needed to make it usable for evaluation. Robot assets remain public as described in Section A.1. Geometric and trajectory coverage is source-dependent: a partial visible surface or an estimated orientation axis is not treated as a complete object model or a fully observed rigid pose.

B.2 Task Organization and Dataset Composition

The benchmark comprises 39 task families used as fixed evaluation groups. Family assignments reflect source-specific task categories, target embodiments, and annotation configurations, as listed in

Figure A1. A reconstruction instance pairs a video with a target simulator, robot embodiment, and control interface; family assignments provide the grouping for task-level reporting. The manifest links each instance to its source sequence, clip boundaries, participating entities, and annotation version, including shared recordings across windows and embodiments. Family-specific conditions are given in Section C.3.

Family inventory. Figure A1 lists every family and its number of reconstruction instances. The labels describe the interaction and retain the embodiment or annotation variant used in the fixed family assignments. F01–F09 cover furniture seating and assembly; F10–F11 cover DROID rigid manipulation and cloth folding; F12–F19 cover human-demonstrated placement or lifting under the two target embodiments. F20–F21 contain the native RoboDojo interactions for dual X5 and dual xArm7. These include pickup, tube and charger insertion, block and bowl stacking, tower construction, digit arrangement, category and size sorting, object exchange, and ordered covering/uncovering. Tower construction has one X5 instance; the other twelve native tasks have one instance per embodiment.

F22–F36 cover household placement, pickup with retained holding, initially held insertion, and basket extraction. Their labels distinguish the retained pose, object-origin, visible-surface, and wallet annotation variants. F37–F39 cover planar pushing, rope routing, and soft-toy packing. The accompanying family catalogue records the objective and instance membership for each label, preserving the manifest's assignments without merging families that share a broad verb such as "place". Metric-specific eligibility is specified in Section C.4.

B.3 Hidden Evaluation Specifications

Each instance has a source-derived or explicitly reconstructed evaluation specification. The specification links persistent object identities to geometry, demonstrated motion, and task requirements. Provenance, uncertainty, and visibility masks identify the quantities and intervals that support evaluation. Human-assisted reference outputs are experimental submissions and do not replace source observations.

Entities and scene geometry. Entities include manipulated objects, receivers, supports, articulated structures, and relevant obstacles. Persistent identifiers link each entity's geometry, trajectory, and task annotations. The scene records metric dimensions, initial poses, and passive articulation where present. Evaluation geometry and simulator geometry are distinguished when collision decomposition, primitive fitting, or an articulation conversion changes the representation. A support or cavity inferred from incomplete observations retains that status; a visually plausible reconstruction is not relabeled as a direct measurement.

Demonstrated object motion. Rigid annotations contain positions and orientations; articulated annotations additionally record the relevant joint coordinate when available. Rope and soft-toy targets retain time-indexed particles or surfaces. Cloth uses camera-visible surface points from stereo reconstruction with object masks and robot-occlusion removal. Household point and wallet annotations retain their view and visibility conditions. Such surface records do not automatically establish material-point correspondence or a unique rigid rotation. Coordinate transforms, metres, seconds, scalar-first quaternions, source frame indices, and validity masks are recorded explicitly.

Interpolation and occlusion handling follow the source procedure rather than a universal smoother. Furniture tracking combines tag observations with an object–tool transform and tag-based residual corrections during verified held intervals; offset residuals beyond 3 cm or 20° from the segment median are excluded before temporal smoothing. Gaps in released static segments require compatible bracketing observations (within 1.5 cm and 10°), an admissible gripper state, and the configured gap limit. These are annotation-construction checks, not task-success tolerances. DROID combines visible pose fits with measured tool motion during carrying. Released human-dataset poses undergo the recorded coordinate conversions; interpolated and estimated intervals remain identifiable. Household visibility masks preserve unsupported intervals instead of treating them as observations. Validity masks are shared across methods and are not selected from candidate errors.

Illustrative annotation. Figure A2 follows a HOI4D bottle transfer from the initial support to the receiving table. The clip contains 87 frames at 15 Hz (source frames 42–128), with released bottle

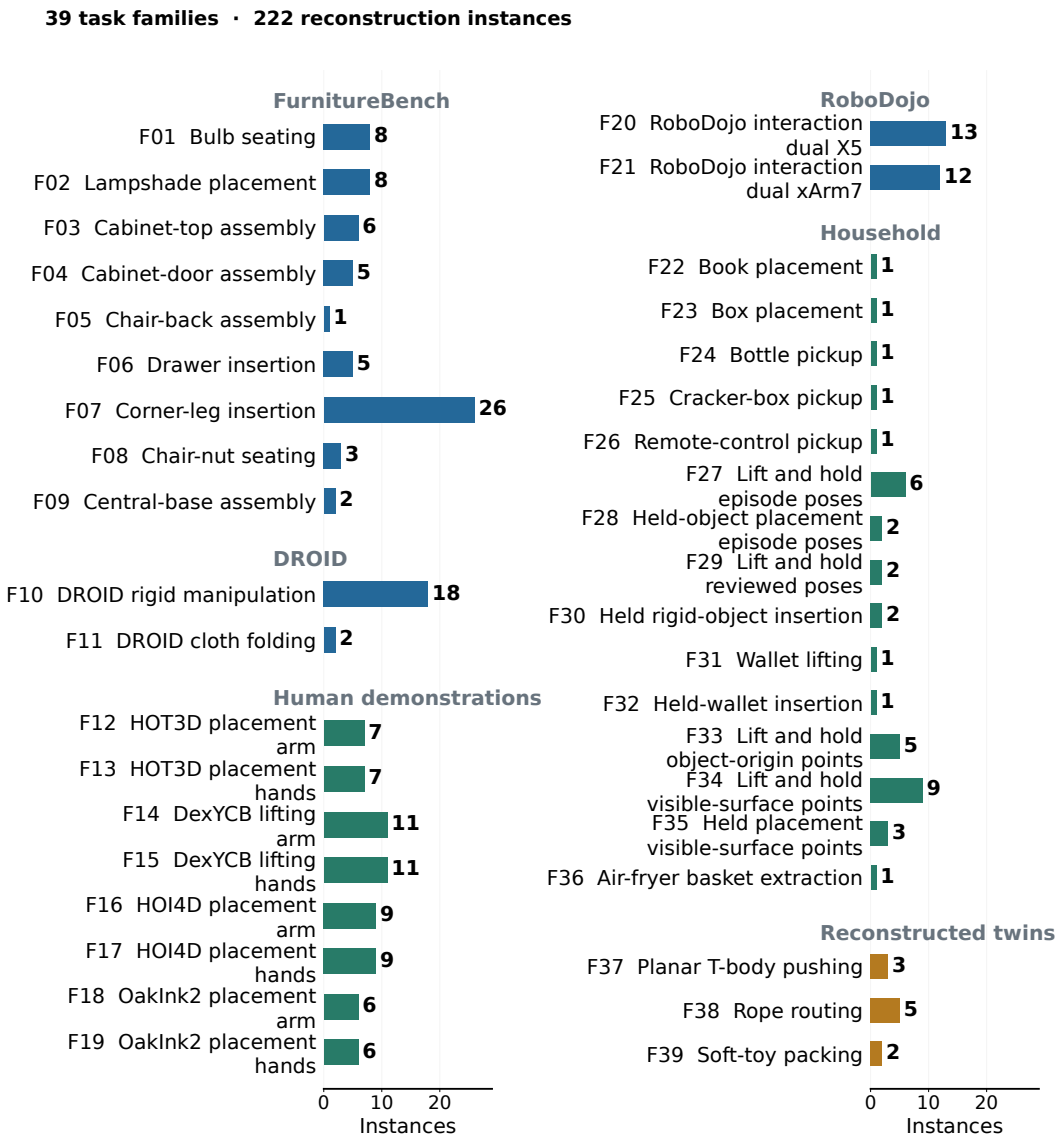

Figure A1: **Complete task-family inventory.** Each bar is one family in the fixed construction manifest, and its label gives the number of video–configuration pairs. The 39 counts sum to 222 instances. Arm and hand variants count separately; their underlying source clips are shared. Family identifiers F01–F39 index the accompanying catalogue. Counts describe the construction inventory rather than a model's successful builds or the metric-specific evaluable subset.

geometry of approximately $7.07 \times 7.00 \times 22.13$ cm. Object and camera poses are composed in a fixed annotation frame, associating the same object with its motion and the receiving surface despite camera movement. The bottle rises by a maximum of 29.2 cm and ends 13.6 cm above its initial height. The annotation specifies pickup and release at clip indices 15 and 80, while the terminal predicate additionally requires the goal pose, support, release, and rest conditions shown in the figure. The receiving support's lateral extent is estimated from the available observations.

Task requirements and initial conditions. The task specification identifies the initial object relations, the events that define the observed interaction, and the terminal goal. A clip beginning with an object already held requires the subsequent transport and, where demonstrated, release; it does not require an unobserved pickup. A clip beginning with an object supported requires the transfer

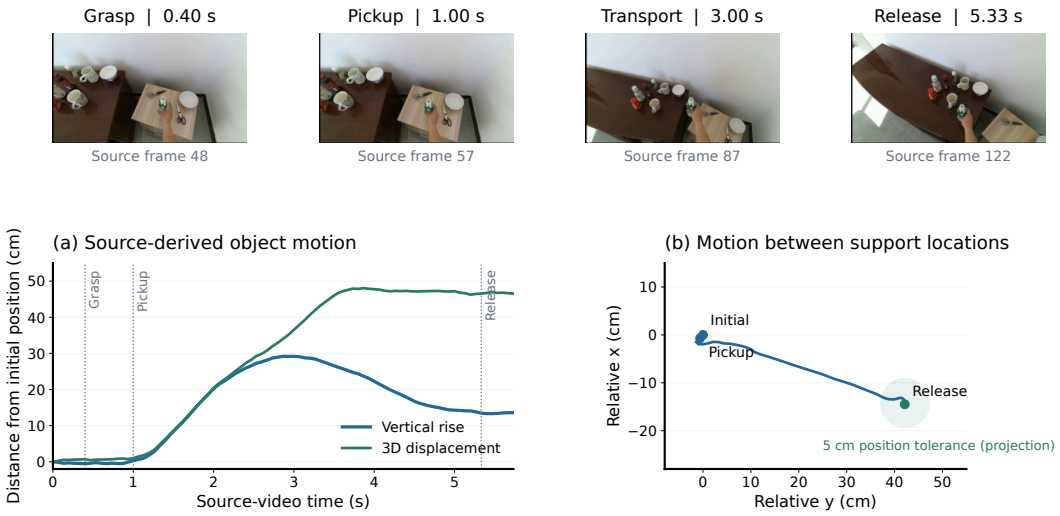

Figure A2: **From a video demonstration to an object-level evaluation specification.** The manually drawn outlines identify the same bottle in frames A–C; they are visual guides, not segmentation annotations. Corresponding points on its annotated height trajectory identify the initial support, transport, and receiving support; dashed lines mark the annotated pickup and release. The conditions at right define acceptable terminal placement, supplementing the required transfer. The curve is computed from source object poses, while the conditions are the evaluation targets; neither is a robot-execution result.

appropriate to that starting relation. These conditions constrain the interaction without requiring exact agreement with every source pose.

For placement goals, a favorable initial arrangement alone earns no completion. If the submitted object initially satisfies the goal relation, it must leave that relation and complete a new placement within the same manipulation attempt, including any required release and stability conditions. An initially satisfied goal is therefore not an automatic rejection of the entire reconstruction, but it cannot receive credit for remaining unchanged. The initial relation is evaluated before the first action; a post-action state cannot substitute for it. For a moving receiver, object–receiver relations use both poses at the same timestamp.

Required events are associated with the manipulated object and a single attempt. Release and stability require recorded evidence when the task calls for them. A release of another object or a nominal settling delay does not establish the required event. Incidental grasp choice and exact source robot motion are generally unrestricted. Native task rules can additionally require robot return to the home configuration; these explicit completion conditions are retained. These requirements are recorded with the task annotation and instantiated by the evaluation rules in Section C.3.

From observations to task conditions. We identify the object relation that changes in the source clip and the events needed to establish that change. A placement annotation links transport to the receiving surface and, when demonstrated, release; removal requires the object to leave its original container; assembly identifies the part and its intended interface. Clip boundaries determine whether acquisition or release is observed and therefore required. The resulting condition is checked against the source frames and the annotated geometry before being used to assess an execution.

Physical trials and inspection of alternative solutions then test whether the condition captures the intended interaction. For example, support alone does not establish removal from a container, and an apparently seated furniture part may still correspond to the wrong interface. The family-specific predicates and numerical tolerances are collected in Section C.3. Here we document their source evidence and the review process through which ambiguities and errors are resolved.

B.4 Worked Example: From Video Input to Scored Execution

We trace one DROID upright-bottle instance through the complete interface, using the saved GPT-6 Astra submission from the visual-refinement experiment. Figure A3 shows its input and physical execution. The example uses the unchanged submitted package and recorded rollout, scored under the current definitions in Section C; no new agent construction or simulation was performed. It is an illustrative instance, not an aggregate model result.

Video, target configuration, and visible resources. The input is an 83-frame, 640×640 RGB clip at 7.5 Hz from a fixed external camera, covering original episode indices 64–228 at stride two. Its first-to-last observation interval is 10.93 s. The visible interaction is to pick up the smaller bottle lying on the table, turn it upright, and release it on the support. The assigned target is a Panda arm with a Robotiq 2F-85 gripper in SAPIEN, controlled by absolute TCP pose and gripper commands at 5 Hz.

The agent receives the video, a contact sheet extracted from it, the public robot dimensions and TCP conventions, and the submission specification. The refinement round additionally supplies its first-pass package, a candidate-only initial render, a montage of that package's executed actions, and diagnostics reporting execution completion, inverse-kinematics error, and rest. The controller produces these simulator observations between rounds. The agent may inspect source frames, edit the package, and check its geometry and schema; this experiment does not give it direct simulator execution. In this worked example, the controller executes the candidate between rounds and supplies rendered observations and execution diagnostics to the agent for refinement. Hidden calibration, source object geometry and motion, task annotations, and scores are withheld. The archived full briefs and delivered feedback distinguish the original construction inputs from the subsequent refinement inputs.

Entity correspondence and hidden annotations. The manipulated entity is the smaller bottle, matched to the submission's moving green capped bottle; the receiving support is matched to its tabletop. The stationary bottle, glass, bowl, and cooker are contextual candidate geometry, not additional motion targets. The reference bottle is a human-reviewed fit to calibrated source observations, with full oriented-bounding-box dimensions of approximately $7.79 \times 7.79 \times 14.36$ cm. Its 83 annotated poses, the support model, timestamps, and withheld camera registration establish geometry and motion targets in the common evaluation frame. The reference bottle begins nearly horizontal, with its axis approximately 87.53° from vertical.

The active task annotation requires an upright bottle supported by the submitted tabletop, followed by release and rest. The bottle's neck determines the directed upright axis when the geometry resolves it. The terminal tilt must be at most 30° from the support normal; support uses a 1 cm clearance tolerance and 5 mm footprint margin. Initial satisfaction alone cannot establish a new placement. The saved brief and original annotation predate the revised bottle rule and retain an absolute-endpoint condition; the current semantic predicate above governs this example's score. Absolute placement discrepancy remains visible in the geometric and motion diagnostics.

Submission and recorded execution. The final package contains the reconstructed bottle mesh, tabletop and contextual geometry, estimated camera and robot placement, and a 97×7 control array with interval 0.2 s. It also includes an 83×7 estimated object-pose sequence, which is a declaration rather than an animation command. For scoring, the evaluator converts the camera-frame delivery into the evaluation frame using the withheld registration. Physical execution applies the submitted controls and records the bottle's states, gripper state, grasp/release events, and final rest condition.

The saved rollout contains 97 object states. On its declared observation clock, grasp is detected at 0.4 s, transport at 2.6 s, release at 12.8 s, and terminal completion at 19.2 s. These times identify the scoring record; the exported video's playback rate is not a physics clock. Motion fidelity uses the 83 demonstration timestamps, with nearest-state correspondence and full recording coverage. Forty-one execution states lie beyond that comparison window. They remain available for the functional decision, which includes the later release and terminal state. Thus, a longer successful interaction can still incur temporal motion error.

Inputs to the decision procedure. The scoring procedure receives four linked records: the validated scene and control package; the build receipt and entity correspondence; the executed object poses,

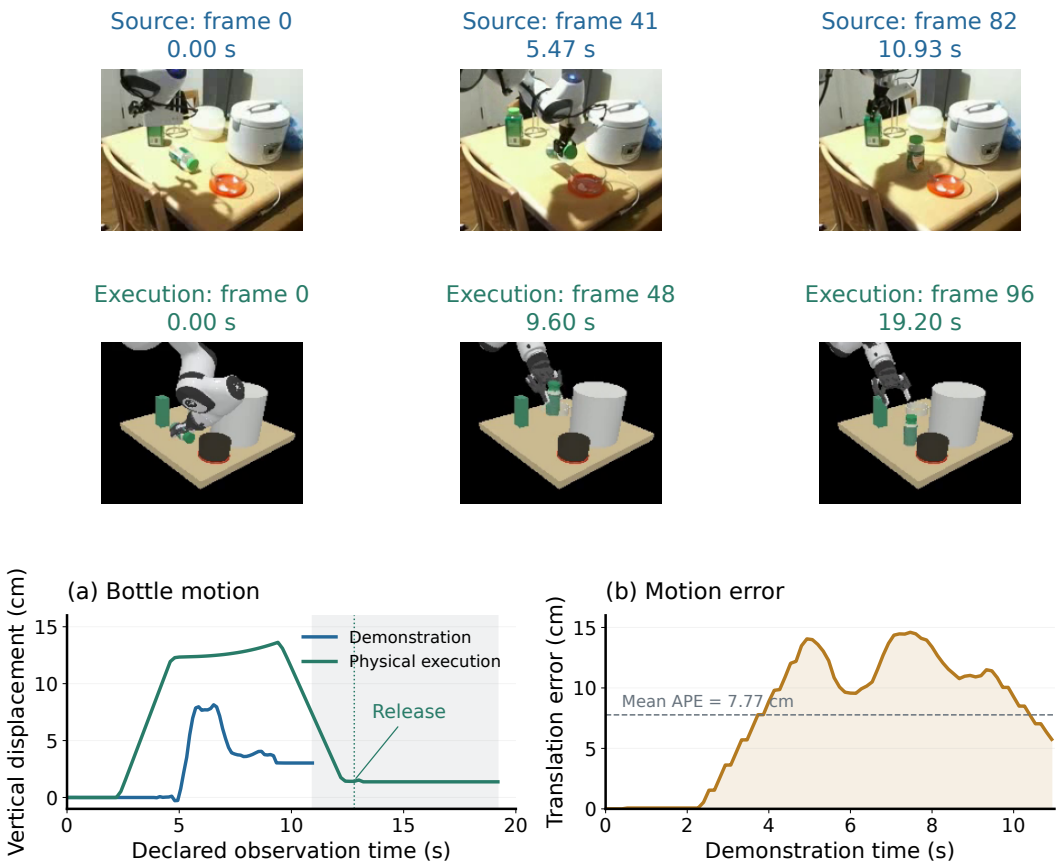

Figure A3: **A complete evaluated reconstruction.** Top: three frames from the agent's RGB input. Middle: three frames of its submitted controls executed in its reconstructed scene; the displayed times use the recorded observation clock. Bottom left: annotated and executed bottle-height changes after the object-frame correspondence used for scoring. The shaded tail lies beyond the demonstration's motion-comparison window, while functional evaluation still uses its release and terminal state. Bottom right: per-frame translation discrepancy on the demonstration clock, whose mean is 7.77 cm. The recorded interaction succeeds despite temporal and geometric discrepancies. Full videos, annotations, controls, execution records, and scoring inputs accompany this example.

terminal poses, grasp/release events, gripper state, and rest flag; and the hidden scene, source poses, camera registration, and active task specification. Geometry is evaluated from the initialized package. Motion is evaluated from the recorded execution, not the submitted pose estimates. The task decision checks the same moving bottle throughout: the initial upright goal is false, the terminal bottle is released and at rest, its support projection is admissible, and its directed axis differs from the tabletop normal by approximately 0.009°. The recorded support gap has magnitude below 0.01 mm, within the prescribed tolerance. These conditions yield success and all four progress stages under the assigned rigid profile. They do not constitute a separate certification of every possible contact or collision.

Final scores and retained evidence. Build and Task Success are both one, and Task Progress is 100%. The continuous scores are

$$\begin{aligned}(\text{Scene CD}, \text{Shape CD}, \text{Size}, \text{Centre}) &= (2.51,\ 3.58,\ 5.76,\ 24.80)\ \text{cm},\\ (\text{translation APE}, \text{translation RPE}) &= (7.77,\ 0.68)\ \text{cm},\\ (\text{rotation APE}, \text{terminal axis error}) &= (9.27^\circ,\ 1.78^\circ).\end{aligned} \tag{7}$$

Terminal displacement error is 4.57 cm. It is measured relative to each trajectory's initial object position, whereas the 24.80 cm centre error retains the initialized placement discrepancy. The low

registered Scene CD therefore does not imply accurate absolute placement, and successful upright placement does not imply accurate bottle dimensions. No composite score is formed. The example supplement retains the input and rollout videos, full public prompts and feedback, both submitted and evaluation-frame packages, hidden annotations, active scoring rules, per-frame arrays, full-precision scores, and file hashes. A record-scoring entry point reproduces the measurements from these saved inputs without rerunning the agent or simulator.

B.5 HUMAN-ASSISTED REFERENCE

The HAR is developed separately for each instance through scene construction, robot-control design, and iterative physical testing. Source geometry, motion annotations, robot logs, and existing simulation assets provide the initial reconstruction; inspection of grasping, transport, insertion, and release guides subsequent revisions. The selected reference comprises both the reconstructed environment and the robot behavior that was executed in it.

This workflow permits access to source annotations, simulator states, task predicates, and evaluation scores. Consequently, comparison with coding agents measures the performance of reconstruction with additional information and human engineering, with selected execution identities and the available construction histories retained.

Construction by source. Furniture references use source geometry and robot/object motion to initialize grasps and transport, followed by physical approach, contact verification, alignment, and release. The current 64 selected furniture executions use PD arm and gripper control with frictional holding; their execution conditions are detailed below.

DROID references combine estimated object geometry with recorded robot motion or object-guided control fitting, retaining the derivation used by the selected package. Human-video references use released hand observations where available, object trajectories, and geometric contact selection to initialize target robot poses. In a carried segment, an initial grasp transform maps object motion into tool targets; physical execution determines whether the grasp actually holds. Contact search and optimization can revise wrist/palm poses and finger controls using object-motion, hand-pose, contact, and validity feedback. RoboDojo uses native task assets and recorded joint-control executions. Household references retain the selected source simulation or documented native reconstruction, with its own state and video identity. Pushing, rope, and toy references adapt the frozen reconstructed twins to the target robot and native execution mechanics.

Additional information by source. Table A10 identifies the resources beyond the agent's RGB input that support each reference workflow. These resources can supply geometry, calibration, motion, or an existing physical execution, rather than merely assisting visual interpretation. Their reuse is part of the HAR method. In the furniture feedback controller, privileged simulator object state also guides execution. Accordingly, the reference measures source-assisted reconstruction and robot engineering; it does not estimate unaided human reconstruction from RGB.

Human effort and recorded iterations. The retained records do not provide systematically measured person-hours or a complete count of manual interventions by source. Human inspection, asset preparation, controller development, and automated search were interleaved. Process runtimes, file timestamps, and optimizer evaluations therefore cannot be converted into human labor. We report the coverage of identifiable trial ledgers and leave unrecorded human-time totals unspecified, rather than imputing a uniform budget.

Figure A4(b) summarizes two documented campaigns. The furniture campaign covers 64 instances with 128 attempt records, including two pilots: 127 contain complete physical records and one records a rendering-initialization interruption. Attempts per instance range from one to six, with median two. A separately audited human-video contact-search batch covers 20 of the 66 arm/hand configurations: 177 physical trial records for six HOI4D configurations, 118 for four HOT3D configurations, 177 for six DexYCB configurations, and 114 for four OakInk2 configurations. Each covered configuration has 26–31 records, including baseline, seeded, and optimized trials. These counts describe saved simulator trials within those campaigns, not manual editing iterations or the total historical effort for each source. DROID, RoboDojo, household, and reconstructed-twin histories lack a comparably consolidated attempt count in this audit; their missing totals are not zero.

Table A10: **Information available to human-assisted reconstruction beyond RGB.** Inputs vary by source and instance. The final column describes their use in constructing the selected reference; it does not claim that all assets were reconstructed from video. The evaluated coding agents do not receive these source-specific annotations or reference executions.

Source	Additional annotations and assets	Use in reference construction
FurnitureBench	Part CAD, calibrated part/robot motion, assembly targets, simulator state.	Reuse part and collision assets; initialize grasps and transport; refine contact, alignment, and release.
DROID	Fitted object/support geometry, calibration, robot kinematics, object motion; visible cloth surfaces.	Initialize scene and controls from source records; fit object-guided motion and refine execution. The two cloth references retain their declared assisted grasp mechanism.
RoboDojo	Native scenes, robot and object assets, task rules, recorded controls and states.	Adapt the recorded native task execution to the benchmark interface.
HOI4D	Instance CAD, object and camera trajectories, fitted support, available hand observations.	Retarget object transport to the arm or hands; optimize contact using annotated motion.
HOT3D	Object CAD, tracked object/camera motion, available hand poses, support estimates.	Initialize grasp and transport, then refine wrist/palm and finger controls.
DexYCB	YCB meshes, calibrated views, object/hand poses, estimated support height.	Reuse geometry and pose observations to construct and refine lifting behavior.
OakInk2	Part scans, calibrated motion capture, task segments, hand/wrist motion.	Prepare geometry and object/hand-guided control targets; refine contact execution.
Household	Source assets, accepted RGB-D pose or surface estimates, paired source simulation or recovered execution records.	Retain the selected recorded simulation or documented native reconstruction, preserving its state/video identity.
Reconstructed twins	Frozen reconstructed geometry and rigid/particle motion; robot and native simulation assets.	Adapt the reconstructed target to xArm7 pushing, rope routing, or toy packing.

Evaluation feedback and unsuccessful references. Reference development permits access to task predicates, reference-based errors, physical diagnostics, and success/progress judgments. The furniture campaign uses scored retries and observed contact failures; the human-video search records use annotated object/hand errors and contact/validity feedback. These workflows are consequently informed by evaluation targets. For reused native or source executions, the retained evidence establishes the selected run and its outcome but does not provide a complete history of how often a person consulted scores. We make no claim of evaluation-blind development for any source.

Both successful and unsuccessful selected references are retained in the benchmark. All 222 instances have a selected reference execution and are included in functional evaluation. The selected execution records the retained reconstruction, not first-attempt success. Earlier failures remain in the available construction histories but do not become additional test instances. Figure A4(a) gives the source-level counts. All 222 instances have a selected execution receipt; this records the retained reconstruction, not first-attempt success. Earlier failures are retained in the available construction histories but do not become additional test instances. The furniture source-selection cap described in Section B.1 remains a separate, outcome-informed admission choice.

Interpretation of the reference score. HAR achieves 58.8% Task Success, macro-averaged with equal weight across the 39 task families covering all 222 benchmark instances, consistent with Table 2. The score characterizes the selected outcomes of source-assisted reconstruction and human engineering. It does not isolate human visual-reconstruction ability or performance under the coding agents' information and construction budgets. Together, the resource disclosure, trial coverage, feedback policy, and retained failures specify the scope of this comparison.

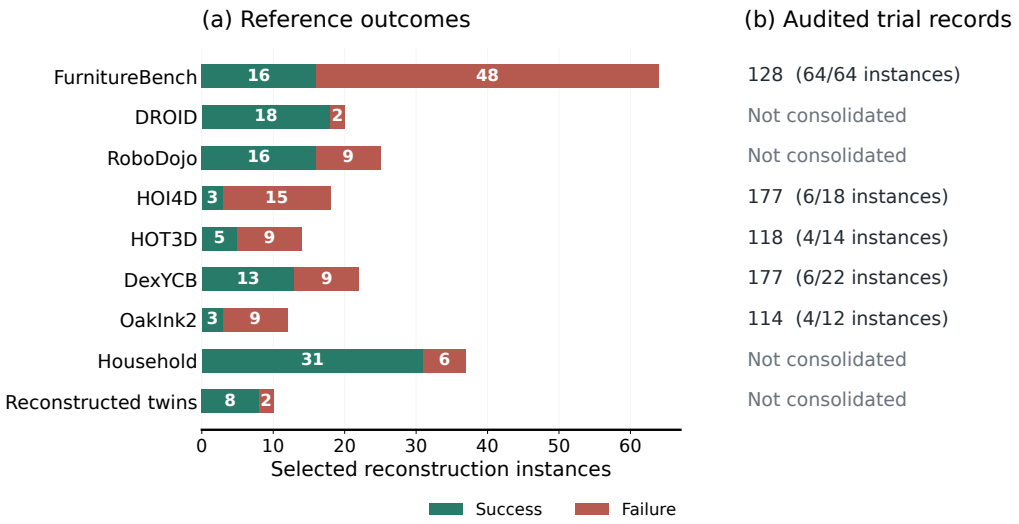

Figure A4: **Source-level reference outcomes and recorded construction effort.** Left: outcomes of the selected references under the current evaluation snapshot, including failures and suspended task targets. These are instance counts; the headline score instead gives equal weight to task families. Right: trial counts from the two audited campaigns, with the number of covered configurations relative to that source's inventory. Unconsolidated histories are explicitly marked. Trial records do not measure human person-hours or complete manual-iteration counts, and coverage differs across sources.

For instance i, the reference and its execution are

$$P_i^h = (\eta_i^h, \pi_i^h), \qquad \tau_i^h = \mathcal{R}_{c_i^h}(\eta_i^h, \pi_i^h). \tag{8}$$

Here c_i^h records the simulator, embodiment, controller, initialization, and observation settings used for the selected HAR execution. The HAR scene η_i^h is evaluated against the annotated reference geometry η_i^*, although the two may share source assets. Shared assets and conversions are documented. When HAR directly reuses evaluation geometry, its geometric score measures correspondence to that geometry and does not independently establish reconstruction ability from RGB. Demonstrated object trajectories provide the motion-evaluation targets, while τ_i^h records the execution of the selected HAR program.

Execution conditions and comparison. HAR and coding-agent reconstructions are evaluated against the same applicable geometry, motion, and task-completion targets using common metric definitions. Each method executes its own robot behavior in its own reconstructed scene. Execution settings are recorded for each method.

For the furniture configurations, both HAR and the coding-agent executor use 100 Hz physics integration and 20 Hz joint control. HAR observes the terminal state for 10 s, while the coding-agent executor uses 1 s of terminal settling. HAR uses PD arm and gripper control, a friction coefficient of 1, and a 70 N finger-force limit, without grasp attachments, pose snapping, or assembly latches. The coding-agent executor additionally permits the bounded initial grasp-alignment correction and, for applicable insertion tasks, the seating-retention approximation described in Appendix A.2. These differences can affect contact outcomes and terminal stability.

The reported comparison therefore evaluates the complete reconstruction workflows, including differences in information access, human refinement, and recorded execution conditions. Isolating the effect of the construction method alone would require matched execution conditions.

Human trials and alternative solutions. Construction involved author-directed attempts, inspection of alternative solutions, and explicit review of failed interactions. The furniture campaign covers 64 instances from 46 original episodes. It records 126 launches for the main campaign: 125

completed physical records and one rendering-initialization interruption, plus two earlier pilot records. Completion of a record includes unsuccessful or early-aborted task execution. Revisions included sustained bilateral-contact confirmation, gripper-width feedback, lift verification, measured in-hand pose, grasp-axis search, slower release, and stopping after loss of hold.

A subsequent two-task study tested two alternatives each for a desk leg and a cabinet door. All four completed physical trials failed; four initialization diagnostics and two early stops were retained separately. The leg trials compared height-aligned insertion with additional small contact searches. Door trials examined 47 grasp candidates and an expanded set of 576 candidates, with geometric and kinematic rejection reasons retained. These attempts exposed a missing door-axis/interface annotation: the annotated endpoint itself was rejected by the old target rule. The response was to repair the task specification and re-evaluate saved execution, rather than interpreting every failed attempt as evidence that the controller or embodiment could not solve the task.

Stopping, feedback, and selection. The comparison does not assume a common construction budget. Available campaign records retain search parameters, attempt limits, and selection decisions, but they do not establish complete human-effort totals for every reference. In the current furniture campaign, the selected execution is the latest successful trial if one exists, otherwise the latest complete trial. Earlier reference campaigns used their declared variant ordering and retained a successful selected execution or the last actual execution when all attempts failed. Subsequent scoring revisions preserve the selected execution and recompute its scores; they do not select a new winner from history. Failed builds, physical failures, and planning attempts that never reached execution remain distinguishable.

Reference success is neither an annotation-acceptance condition nor a guaranteed upper bound. The main comparison executes each method's own behavior in its own scene, without transferring controls between reference and coding-agent scenes. Differences reflect the complete construction workflows, including privileged information, engineering effort, and any recorded execution-condition differences.

B.6 Human Evaluation

Evaluation target and stimuli. We assess overall scene-and-interaction faithfulness through pairwise comparisons of the nine coding-agent systems and HAR. Each comparison presents the source demonstration, denoted R, alongside two reconstructions of the same case, denoted A and B. A case identifies a source demonstration and target configuration; different embodiments of the same source video can form distinct cases. Each coding-agent video shows its own generated behavior executing in its own reconstructed scene. HAR participates as an ordinary candidate, rather than as the source video or a guaranteed correct answer.

The rating question was: "Which result more faithfully reconstructs the scene and interaction in the real video?" Raters were instructed to consider objects and layout, task semantics, motion, and visible physical plausibility. They were also instructed that geometric simplification or a different robot embodiment does not automatically imply failure, and that visual detail alone should not determine preference. This question measures holistic faithfulness rather than a separate task-success judgment.

Videos were displayed without audio at 640×480 resolution using aspect-preserving resizing and padding. We retained the complete recorded interaction, including prelude, waiting, and terminal settling, at its original duration and $1\times$ playback speed. Videos could be played independently or together; neither camera viewpoints nor trajectory timing were forced to match. A preference submission required at least 90% playback coverage of each playable video. "Unable to judge/playback issue" could be submitted without meeting that threshold. All retained preference judgments met the playback requirement, which checks playback rather than sustained attention.

Case sampling and comparison coverage. The study used a frozen pool of 195 cases from 30 task families, preceding the final 222-instance benchmark inventory. We sampled 54 cases using task-family stratification and seed 20260924. Each family received at least one case; the remaining slots were allocated by the largest deficit relative to the family's proportional share of 54, subject to availability. Cases were then sampled within families using the fixed seed.

We planned two agent–agent comparisons per sampled case, giving 108 comparisons, with each of the 36 agent-pair types assigned to three cases. We then added three HAR comparisons per agent across 27 distinct families, retaining the original sampled cases and agent comparisons. Thus, the complete plan contained 135 case–candidate-pair comparisons, with each of the 45 system-pair types represented three times. Pair allocation used identifiers, family membership, and assignment counts, without consulting automatic scores, task-success labels, or human preferences.

Media and provenance checks admitted 83 comparisons from 41 cases: 70 agent–agent and 13 agent–HAR comparisons. The remaining 52 comparisons were deferred, without replacement by higher-quality or successful examples. Admission required verifiable source correspondence and complete execution footage, or a confirmed agent-caused build failure represented by an anonymous failure panel. Missing media or infrastructure failures were deferred rather than treated as candidate losses. The released comparisons covered 43 of the 45 system-pair types; the effective comparison graph connected all ten systems, and HAR was compared with every agent. Actual coverage was therefore less balanced than the planned design. Playable executions were retained regardless of task success.

Participants, masking, and assignment. Twelve invited internal raters, comprising paper authors and company employees, participated voluntarily with informed consent and received no additional compensation. They were not informed of the specific experimental purpose. Each released comparison was assigned to three distinct raters using a seeded allocation that balanced workload. Nine raters completed 21 primary comparisons each and three completed 20 each. Raters were instructed to answer independently.

The interface, media addresses, and personal exports concealed system names, HAR identity, automatic scores, and other raters' answers. The organizers retained the identity mapping. A/B positions were randomized independently for each assignment, and question order was shuffled within each rater. R remained the fixed source reference; the interface did not prescribe a randomized playback sequence. Masking candidate labels does not establish that internally recruited raters had no prior familiarity with the outputs.

Responses and quality checks. The five responses were A better, B better, equivalent, both poor, and unable to judge/playback issue. The 83 comparisons yielded 249 primary judgments: 79 A-better, 82 B-better, 14 equivalent, 68 both-poor, and six unable-to-judge responses. The six excluded responses came from two comparisons for which all three raters selected unable to judge; both candidates in each comparison were confirmed build failures. The remaining 243 judgments covered 81 comparisons across 39 cases and 19 task families, with three valid judgments from distinct raters per comparison. Seven cases contributed three judgments each, 22 contributed six, and ten contributed nine. Raters could overlap across different comparisons of the same case. Valid preferences covered seven of the nine source categories in the sampling pool; no Poway or RoboDojo comparison passed admission in this study.

Each rater also received one unannounced repeat comparison, separated from its original presentation by at least eight primary questions and with independently randomized A/B positions. These 12 repeats were excluded from ratings and confidence intervals. After aligning candidate identities, repeat-answer consistency was $8/12$; no rater was excluded on this basis. Across the 81 valid comparisons, all three raters agreed in preference direction on 31, counting equivalent and both-poor responses as ties. These are descriptive consistency measures.

Human Preference Rating. We fit a Bradley–Terry model to all 243 valid judgments, rather than to majority votes. For judgment t comparing candidates i_t and j_t, let $y_t = 1$ when i_t is preferred, $y_t = 0$ when j_t is preferred, and $y_t = 0.5$ for either equivalent or both-poor responses. Unable-to-judge responses are excluded. With $d_t = \beta_{i_t} - \beta_{j_t}$, we minimize

$$\mathcal{L}(\boldsymbol{\beta}) = \sum_t \left[\log(1 + \exp(d_t)) - y_t d_t\right] + \frac{0.001}{2} \sum_{m=1}^{10} \beta_m^2. \tag{9}$$

Each valid judgment has unit weight; we do not additionally equalize case or family weights. Equivalent and both-poor responses each contribute half a win and half a loss, while their counts remain separately reported. Both-poor responses account for $68/243 = 28.0\%$ of valid judgments. No separate tie-probability parameter is fitted.

We optimize using L-BFGS-B from zero initialization, with `ftol=1e-12`, `gtol=1e-8`, and at most 1,000 iterations. After checking convergence, we center the ten fitted parameters and convert them to the Elo scale:

$$\mathrm{HPR}_m = 1000 + \frac{400}{\ln 10}(\beta_m - \bar{\beta}), \qquad \bar{\beta} = \frac{1}{10}\sum_{m=1}^{10} \beta_m. \tag{10}$$

The mean Human Preference Rating (HPR) is therefore 1000. HPR is a batch estimate of relative preference, not an online Elo update or a task-success percentage.

Confidence intervals. We use 2,000 source-video-cluster bootstrap replicates with seed 20260924. Clusters are defined by the SHA-256 hash of the original demonstration, keeping all associated embodiments, comparisons, and raters together. The 41 cases with primary responses form 39 source-video clusters. Two clusters contain only unable-to-judge responses and remain in the resampling pool; the 243 valid judgments originate from 37 nonempty clusters. Each replicate samples 39 clusters with replacement, carries their valid judgments with the corresponding multiplicities, and refits all ten ratings with the same regularization and centering. Sampling an empty cluster adds no judgments. All 2,000 fits were valid. The 95% intervals in Figure 3(a) are the 2.5th and 97.5th percentiles of each system's bootstrap ratings.

Interpretation and scope. The intervals quantify uncertainty under source-video resampling for this rater group; raters are not independently resampled. We do not infer significant pairwise differences from the point-estimate ranking. The study measures relative holistic faithfulness under its observed case coverage and internal participant pool. HAR uses privileged information and human refinement, and its videos came from an independently versioned reference release; participation in the same preference study does not imply matched construction budgets or identical automatic-evaluation snapshots. Automatic scores do not enter the preference model or determine its outcomes. The study compares complete reconstruction workflows and complements automatic evaluation with an assessment of their visible outputs.

B.7 Annotation Verification and Evaluator Refinement

Verification examines whether the annotations represent the demonstrated interaction and whether the evaluator recognizes valid executions of that interaction. Source videos, geometric overlays, object trajectories, and task semantics provide the evidence used to define and adjudicate evaluation criteria. Reference and agent-generated executions are used only to expose ambiguous or incomplete predicates; they do not determine the intended task outcome or method-specific scoring rules. Any identified ambiguity is resolved against the source demonstration and task specification, after which the revised evaluator is applied uniformly to all submitted methods. A different grasp or transport path is therefore admissible when it preserves the task-defining events and outcome.

Author review and adjudication. Authors inspect the source video, annotated object motion, reference execution, and task predicates together. An accepted annotation can accompany an unsuccessful HAR execution, and a geometric modeling concern need not invalidate an otherwise successful interaction. Accordingly, these counts describe review coverage, not agreement with binary task-success labels.

Review identified two erroneous success decisions, in bulb insertion and drawer insertion, and three additional progress disagreements, in cabinet-top and drawer assembly. In these cases, the scoring criteria were corrected by referring back to the demonstrated interaction and its task-defining events. Other review notes concern interpenetration, inaccurate geometry, or uncertain endpoints. Corrections were incorporated into the shared evaluator and the affected submissions were rescored under the same revised criteria.

All results reported in the paper are computed using the same final evaluation snapshot; no reported method is scored under a method-specific or earlier predicate version.

Human adjudication of scoring errors. Saved executions also document two complementary scoring failures and their corrections. First, an endpoint-only rule accepted a lampshade placement

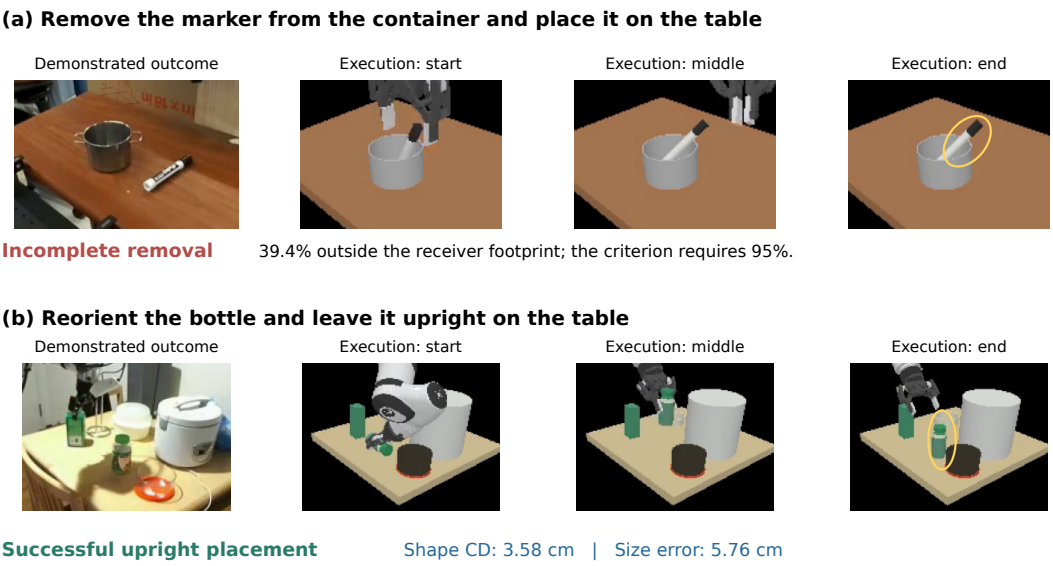

Figure A5: **Evaluating the requested interaction separately from geometric fidelity.** Each row shows the demonstrated outcome and three frames from the scored Astra visual-feedback execution, with manually drawn endpoint outlines identifying the manipulated object. (a) The marker remains partly in the container; robot withdrawal does not establish object extraction. (b) The bottle is reoriented and placed upright, satisfying the task predicate despite the reported shape and size errors. Execution frames are 0, 36, and 72 for (a), and 0, 48, and 96 for (b). Judgments use the complete saved rollout; frames are cropped for visibility and are not temporally aligned with the source endpoint.

whose initial state already satisfied the broad goal condition. In the saved 36-execution control study spanning 12 source clips, the original action sequence and two modified sequences for this instance all received full success and progress. Requiring departure from the initial goal and a new placement changed all three judgments to failure, with progress reduced to 0.25; the other 33 judgments were unchanged. These are erroneous acceptances of an interaction that was not completed.

Second, two Astra bottle reconstructions were rejected despite an upright, released, supported endpoint. The earlier comparison paired opposite ends of the reference bottle axis and also imposed proximity to an unnecessary absolute source endpoint. Adjudication instead required the candidate bottle's directed axis to point upward, together with support, release, rest, and an actual reorientation. Both reconstructions then satisfied the revised predicate in a review covering six executions of this source clip. Figure A5(b) shows the visual-feedback reconstruction. These erroneous rejections concern task interpretation; geometric discrepancies remain measurable under the separate reconstruction metrics. The documented cases are examples of evaluator refinement, rather than an estimate of error prevalence across the benchmark.

Counterexamples and admissible alternatives. Figure A5 contrasts incomplete extraction with successful reorientation. In the extraction case, the marker remains partly inside its receiver after the robot withdraws: only 39.4% of its vertices lie outside the receiver footprint, below the 95% removal criterion. In the reorientation case, the bottle is placed upright, but its Shape CD and Size error remain 3.58 cm and 5.76 cm. Thus, successful interaction and geometric reconstruction accuracy require separate measurements. A further cup-placement reconstruction satisfies its relative-placement goal with Scene CD 3.64 cm, Shape CD 3.05 cm, and Size error 6.23 cm.

Counterexamples also test whether the rules reject an already solved initial arrangement without excluding legitimate alternative solutions. Native solved-start probes require a new object-level completion, rather than robot return home alone. Conversely, two shortened Push-T executions retain success because the object reaches the goal before truncation: their position/orientation errors are 1.09 cm/5.90° and 2.27 cm/7.80°, within the 4 cm/15° tolerance. A modified or shorter action sequence is therefore evaluated by its physical outcome, not assigned a failure label by construction.

Revision and consistent rescoring. The recorded revisions address initial-goal shortcuts, object-specific release and stability, central versus corner leg interfaces, door-axis annotations, and directed bottle orientation. Container and support rules are defined on the receiver's free cavity and bearing surface. A corrected condition applies to every method under the same task contract. Changes confined to scoring are evaluated on the retained executions; changes to physical execution require a new rollout. Annotation, evaluator, and execution versions preserve the association between a reported judgment and the evidence on which it depends.

C Evaluation Details

Video2World evaluates geometric fidelity, fidelity of executed object motion, and functional correctness separately. Geometry describes the initialized reconstruction; motion metrics compare its physical execution with the demonstration; task metrics assess whether the requested interaction is completed. These quantities characterize a complete reconstruction, including both its environment and its robot behavior. Robot trajectories provide execution evidence but are not motion-matching targets.

For method m and instance i, the evaluated reconstruction and rollout are

$$P_i^m = (\eta_i^m, \pi_i^m), \qquad \tau_i^m = \mathcal{R}_{c_i^m}(\eta_i^m, \pi_i^m). \tag{11}$$

The record c_i^m identifies the assigned simulator and embodiment and the controller, initialization, sampling, and settling conditions actually used. The HAR execution settings are specified in Section B.5. Each method executes its own behavior in its own scene; fixed-reference-action replays, where retained, are separate diagnostics. Superscript $*$ denotes the evaluation annotations described in Section B.3, including explicitly modeled targets for reconstructed twins. The method superscript is suppressed below. Metric applicability follows the available annotation and the assigned evaluation profile.

C.1 Geometric Fidelity

Rigid-scene evaluation compares the initialized geometry in the prescribed common frame. Scene Chamfer Distance measures visible scene correspondence after bounded registration. Object-level measurements separately retain physical size and initial placement. Distances in the definitions below are in metres and are reported in centimetres.

Surface extraction and visibility. Task-relevant entities are associated through the instance's semantic roles. Where an object is delivered as several parts, the registered assembly procedure combines the corresponding physical parts before surface sampling. Supports are distinguished from manipulated and receiving entities. Mesh surfaces are sampled uniformly by area at 10^4 points/m^2, with 300–40,000 points per entity and random seed zero. Canonical triangle ordering makes the scene samples independent of vertex indices, face ordering, and winding. The dense reference surfaces used as distance targets use 64 times this density and upper sample limit. Candidate samples are deduplicated into 4 mm voxel centres.

Visibility is determined with the per-instance evaluation camera after the configuration's coordinate conversion. Samples lie within a 0.9 m horizontal radius of the robot base, in front of the camera by more than 5 cm, and within the calibrated image's valid content rows, excluding letterbox padding. A rasterized depth buffer removes occluded samples using a 1 cm tolerance plus the depth range in a 3×3 pixel neighbourhood. The candidate's identified table is excluded from the depth buffer used for its object surfaces, so a misplaced support cannot hide those objects from comparison. A required reference manipulated or receiving entity with no visible samples retains its unoccluded surface within the workspace. Profiles without a calibrated evaluation camera use the workspace surface directly.

Interaction weights. Let $\mathcal{E}_i$ contain the reference non-support entities with surface samples. A manipulated object or its designated receiver receives weight two. For another entity e, let $d_{i,e}^{\mathrm{ctx}}$ be its minimum distance to the demonstrated TCP path and the manipulated object's swept geometry. Its

weight is

$$w_{i,e} = \mathrm{clip}_{[0,1]}\left(\frac{0.25\,\mathrm{m} - d_{i,e}^{\mathrm{ctx}}}{0.20\,\mathrm{m}}\right). \tag{12}$$

Thus, nearby context receives unit weight within 5 cm and zero weight beyond 25 cm. The sweep uses at most 300 distinct object vertices at regularly spaced source frames, with stride $\max(1, \lfloor T/15 \rfloor)$ for a T-frame trajectory; context entities use at most 1,500 vertices. Vertex subsampling uses seed zero, and weights are rounded to three decimal places. Nonfinite source TCP observations are excluded. An entity whose context distance cannot be determined receives unit weight; if no demonstrated motion is available, all context entities receive unit weight. These weights depend on the evaluation annotations rather than the candidate.

Interaction-weighted surface distance. Write $\mathcal{S}_{i,e}$ for an entity's reference samples, $\mathcal{B}_{i,e}$ for its dense surface, and $\mathcal{B}_i^{\mathrm{tab}}$ for the dense reference support. Let $\mathcal{C}_i$ be the candidate non-support voxel centres and $\widehat{\mathcal{C}}_i = g_i(\mathcal{C}_i)$ their registered positions. With

$$d(p, \mathcal{Q}) = \min_{q \in \mathcal{Q}} \|p - q\|_2, \qquad \mathcal{B}_i = \mathcal{B}_i^{\mathrm{tab}} \cup \bigcup_e \mathcal{B}_{i,e}, \tag{13}$$

the forward entity residual is

$$a_{i,e} = \mathrm{med}_{p \in \mathcal{S}_{i,e}}\, d(p, \widehat{\mathcal{C}}_i), \qquad \delta_{i,e} = \min\{a_{i,e}, c_{\mathrm{sc}}\}, \qquad c_{\mathrm{sc}} = 0.10\,\mathrm{m}. \tag{14}$$

This residual compares each annotated entity with the complete candidate non-support surface. Semantic correspondence determines roles and assembly membership; it does not impose a separate binary presence test on this nearest-surface distance. For a candidate point p, let $e_i(p)$ be the entity containing its nearest dense reference non-support sample, and set $\omega_i(p) = w_{i,e_i(p)}$. Scene Chamfer Distance is

$$d_i^{\mathrm{sc}} = \frac{\sum_e w_{i,e}\delta_{i,e}}{\sum_e w_{i,e}} + \mathrm{wmed}_{p \in \widehat{\mathcal{C}}_i;\, \omega_i} \min\{d(p, \mathcal{B}_i), c_{\mathrm{sc}}\}. \tag{15}$$

The weighted median is the first sorted residual whose cumulative weight reaches half the total weight. If all weights in either term are zero, that term uses its unweighted counterpart. Support is included as a reverse-distance target, although candidate support samples are scored separately. If the candidate has no non-support samples, both terms receive the 10 cm cap, giving a 20 cm Scene CD. Missing measurable reference surfaces constitute an evaluation error.

Bounded registration. Scene registration uses

$$g(p) = sRp + t, \qquad s \in [0.95, 1.05], \qquad \theta(R) \le 15^\circ, \tag{16}$$

with free translation. A deterministic finite search initializes translation from the identity, scene centroids, coordinate-wise medians, bounding-box centres, and available corresponding-entity centroids. Each initialization undergoes at most 30 similarity iterative closest point (ICP) iterations, retaining the closest 70% of point correspondences. Scale is bounded at each update; a rotation beyond 15° is clamped about its estimated axis and translation is refitted. The two best hypotheses undergo additional translation-only Powell refinement within ± 10 cm of their fitted translations, using at most 180 objective evaluations and position/objective tolerances $10^{-5}/10^{-6}$. The final hypothesis minimizes Eq. 15 among the fitted candidates and the identity transform. This finite search does not assert a global optimum.

Supports do not anchor object registration. Their classification is checked again in the registered frame, with refitting if the classification changes. Native multi-object scenes additionally use the declared initial-object anchor to establish their common frame before surface comparison. This coordinate preparation and the bounded scene fit are recorded separately. The fitted scene transform is not applied to object-placement or trajectory errors.

Missing geometry, extra geometry, and supports. With threshold $\rho = 0.05\,\mathrm{m}$, the coverage diagnostics are

$$f_i^{\mathrm{miss}} = \frac{\sum_e w_{i,e}\mathbf{1}[a_{i,e} > \rho]}{\sum_e w_{i,e}}, \qquad f_i^{\mathrm{extra}} = \frac{\sum_{p \in \widehat{\mathcal{C}}_i} \mathbf{1}[d(p, \mathcal{B}_i) > \rho]}{|\widehat{\mathcal{C}}_i|}. \tag{17}$$

Missing geometry is therefore a weighted fraction of entities, whereas extra geometry is a fraction of candidate voxels. The former uses an unweighted mean if all entity weights are zero. Empty candidate geometry gives missing fraction one and extra fraction zero, with the empty denominator recorded. An entity classified as missing may have a residual between 5 and 10 cm; its distance contribution is still the capped measured residual.

Explicit supports and identified table parts are evaluated separately. Otherwise, a candidate part is table-like when its horizontal span is at least 45 cm, its median distance to the reference table plane or surface is at most 10 cm, and fewer than half its samples lie within 3 cm of a reference non-support entity. Declared physical assemblies retain their object role. Table error is the median distance from the registered candidate support samples to the annotated plane or dense support surface, capped at 10 cm. A missing candidate support receives this cap; absent reference support makes the measure inapplicable. For instances with source-annotated partial tabletop boundaries, an additional image-space diagnostic measures distance to the projected candidate boundary using the delivered camera and no fitted scene transform. This diagnostic is reported in pixels and retains the annotation's visible extent.

Object shape, size, and placement. For point sets $\mathcal{P}$ and $\mathcal{P}^*$, define the untruncated bidirectional distance

$$D(\mathcal{P}, \mathcal{P}^*) = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} d(p, \mathcal{P}^*) + \frac{1}{|\mathcal{P}^*|} \sum_{q \in \mathcal{P}^*} d(q, \mathcal{P}). \tag{18}$$

Shape CD evaluates $D(h\mathcal{P}, \mathcal{P}^*)$ after rigid alignment without scaling. Each mesh contributes 4,000 area-uniform samples in its local frame, using seed zero. Symmetric ICP starts from the 24 proper axis-aligned rotations about centred point sets, runs 12 iterations per start, and refines the best three for 40 further iterations. The smallest bidirectional distance is retained. Sampling in local mesh frames makes this shape measure independent of the submitted world pose.

Size error is

$$e_i^{\text{size}} = \| \operatorname{sort}(\ell_i) - \operatorname{sort}(\ell_i^*) \|_2, \tag{19}$$

where ℓ_i and ℓ_i^* are the full edge lengths of independently fitted oriented bounding boxes (OBBs). Sorting removes axis naming conventions; it does not establish semantic width, depth, and height correspondences. Centre error compares surface-sample centroids at the initialized poses in the common frame. Mesh-origin error is retained separately. Where a receiver is specified, the initial relation $T_{r \to m} = T_r^{-1} T_m$ provides a further placement diagnostic. Object-coordinate correspondences are fixed from the reconstructed and annotated geometry and retained throughout execution; they are distinct from fitting a trajectory to its target.

C.2 Dynamic Fidelity

Dynamic fidelity measures object motion generated by the submitted behavior in the submitted environment. It reflects both scene and control quality. Pose, visible-point, and deformable-surface annotations support different comparisons, as specified below.

Temporal correspondence. The shared rigid-pose evaluator uses the demonstration timestamps and each execution's declared observation clock. For regularly sampled records, the first recorded state defines time zero; demonstration time $k\Delta t^*$ selects execution index $\operatorname{round}(k\Delta t^*/\Delta t)$, clipped to the recorded range. Half-index ties use round-to-even. Household pose profiles with timestamped RGB-D observations instead use linear position interpolation and quaternion SLERP at the source timestamps; profiles with explicit observation-to-execution indices use those indices. The source time origin and any initialization offset belong to the profile's execution record. Observation intervals are distinct from physics integration timesteps.

No temporal warping is applied to these primary motion scores. For a completed execution shorter than the demonstration, the scorer extends the last recorded state over the remaining comparison times. This is an endpoint-hold convention for scoring, not additional simulated motion. The regular pose profile reports the number of demonstration times beyond the last execution time plus half a sample interval, and coverage is one minus this count divided by the number of demonstration frames. Timestamped profiles report coverage on their recorded time interval. Execution states beyond the demonstration window are excluded from motion comparison but remain available for terminal task

evaluation. Construction failures, incomplete records, and infrastructure errors follow the outcome rules in Section C.4; an interrupted trace cannot be substituted for a completed execution. Duration-normalized motion scores, when retained, are explicitly separate diagnostics with the time-scale factor recorded.

Translation APE. For object e, let $p_{i,e}(t)$ and $p^*_{i,e}(t)$ denote executed and demonstrated positions after temporal correspondence. Each trajectory is translated by its own initial position:

$$\widetilde{p}_{i,e}(t) = p_{i,e}(t) - p_{i,e}(0), \qquad \widetilde{p}^*_{i,e}(t) = p^*_{i,e}(t) - p^*_{i,e}(0). \tag{20}$$

Translation absolute pose error (APE) is the capped mean

$$e^p_{i,e} = \frac{1}{|\mathcal{T}_i|} \sum_{t \in \mathcal{T}_i} \min\{\|\widetilde{p}_{i,e}(t) - \widetilde{p}^*_{i,e}(t)\|_2, c_p\}, \qquad c_p = 1\,\mathrm{m}. \tag{21}$$

This preserves motion direction, magnitude, and timing while removing constant initial translation. No scene-registration rotation or scale is fitted to the trajectories. Raw per-frame errors, the uncapped 90th percentile and maximum, and the proportion supported by execution records accompany the capped mean. Initial placement remains a separate geometric measurement.

Orientation and symmetry. Define the rotation distance

$$d_R(R_1, R_2) = \arccos\left(\mathrm{clip}_{[-1,1]} \frac{\mathrm{tr}(R_1^\top R_2) - 1}{2} \right). \tag{22}$$

For the declared symmetry group $\mathcal{H}_{i,e}$, rotation APE is

$$e^R_{i,e} = \frac{1}{|\mathcal{T}_i|} \sum_{t \in \mathcal{T}_i} \min_{Q \in \mathcal{H}_{i,e}} d_R(R_{i,e}(t), R^*_{i,e}(t)Q). \tag{23}$$

Asymmetric objects use the identity; declared cube symmetry enumerates 24 proper rotations. Continuous axial symmetry compares the corresponding directed axes, removing spin about the axis. Registered bottle axes distinguish a resolved neck direction from geometrically indistinguishable ends: their angular errors are respectively $\arccos(a^\top a^*)$ and $\arccos|a^\top a^*|$, with dot products clipped to their admissible range. Orientations are expressed in corresponding object frames without initial-orientation subtraction. Angular results are reported in degrees. Incomplete orientation observations use the separate measures below.

Translation RPE. Translation relative pose error (RPE) measures displacement increments in the common fixed coordinate axes. For each start time t_k, choose the subsequent recorded comparison time t_j nearest to $t_k + 0.2\,\mathrm{s}$, breaking ties toward the earlier time. Starts whose requested endpoint exceeds the comparison window are excluded. With $\mathcal{P}_i$ denoting the resulting pairs,

$$r^p_{i,e} = \frac{1}{|\mathcal{P}_i|} \sum_{(k,j) \in \mathcal{P}_i} \min\left\{ \|[p_{i,e}(t_j) - p_{i,e}(t_k)] - [p^*_{i,e}(t_j) - p^*_{i,e}(t_k)]\|_2, c_p \right\}. \tag{24}$$

Actual intervals are recorded to expose discretization of the 0.2 s target. Where a source-validity mask applies, every frame in the interval must be valid. A sequence too short for the interval has no RPE value. This translation measure is independent of the choice of object orientation axes. A supplementary rotation RPE compares $R(t_k)^\top R(t_j)$ with its demonstrated counterpart using the declared symmetry-aware angular distance. It applies symmetry independently to each interval and does not perform a sequence-wide selection of continuous pose representatives.

Deformable and partially observed motion. Rope and soft-toy trajectories are compared as point sets in their frozen reference frame. At each source timestamp, the nearest actual execution state is selected, with earlier-state tie breaking and endpoint holding outside the recorded interval; any declared initialization prefix is removed from the execution time origin. Their trajectory error is

$$e^{\mathrm{surf}}_i = \frac{1}{|\mathcal{T}_i|} \sum_{t \in \mathcal{T}_i} D(\mathcal{P}_i(t), \mathcal{P}^*_i(t)), \tag{25}$$

using Eq. 18 without rigid fitting, initial-centroid subtraction, or per-frame recentering. The representation uses the retained executed object surface samples and annotated particles; node-centreline distance is an additional rope endpoint diagnostic. These distances are untruncated, and recording coverage is reported. Initial and terminal surface errors use the same distance, while their size diagnostic uses sorted full world-axis bounding-box extents, distinct from rigid-object OBB size.

Cloth is compared on visible surfaces: simulated geometry is filtered by the source cameras and robot/self-occlusion conditions and voxelized like the observed surface. The surface trajectory uses Eq. 25. A separate centroid trajectory uses the translation APE/RPE definitions above, rebased at its initial visible-surface centroid. No rigid orientation error is assigned to cloth. Articulated objects use the registered task-relevant parts and relations; a universal joint-angle tracking score is not assumed for annotations lacking joint correspondences.

Household point and axis observations. The point-observation profiles compare the physical target surface along fixed source observation rays. The evaluator intersects those rays with the moving candidate object, retains depths between 0.2 and 2.5 m, and removes depth outliers beyond the larger of 15 mm and 3×1.4826 median absolute deviations. A frame requires at least 40 surviving intersections. The coordinate-wise median defines its observed point, and the surviving-ray fraction is ρ_t. For raw position error d_t, the scored error is

$$\ell_t = \rho_t \min\{d_t, 0.10\,\mathrm{m}\} + (1-\rho_t)\,0.10\,\mathrm{m}. \tag{26}$$

An unavailable point receives 10 cm, and the mean uses the complete source timeline. Displacement error uses a four-frame interval and the minimum coverage over its five frames, with the same cap and missing-observation penalty. These observation-based distances do not perform spatial trajectory fitting or candidate-dependent masking.

For source annotations supporting only geometric directions, long-axis or normal error is the unsigned axis angle. When a full geometric triad is observable, frame error minimizes angular distance over its four proper sign choices, without axis permutation. Source-validity masks are fixed from the observations; source and prediction coverage accompany the errors. Candidate frames are interpolated with SLERP at source timestamp offsets and held at endpoints outside the execution interval. These measures describe observable geometry and are reported separately from semantic full-orientation APE.

Within-instance object aggregation. The evaluation profile fixes the task-relevant object set. Single-object profiles report the manipulated object's motion. RoboDojo averages the per-object metrics equally over its named rigid task objects, excluding unlabeled clutter; its scene score additionally averages the specified interaction-role comparisons. Per-object values remain available, and averages of object-wise maxima or percentiles are identified as such. A missing required comparison is not silently removed from the object set. A defined missing-object penalty is applied where specified; otherwise the metric remains unresolved until the required evidence is available.

C.3 Functional Correctness

Functional evaluation checks the interaction defined by the source clip rather than prescribing the demonstrator's exact motion. Its conditions concern the initial object relations, essential events and their ordering, and the terminal goal under the assigned execution checks.

Task success. For a completed, evaluable rollout $\tau_i = (x_0, u_0, \ldots, x_H)$, let α_i denote admissible initial conditions, β_i the required interaction events, and ψ_i the terminal relation, including its required persistence. With v_i indicating passage of the profile's applicable execution-validity checks,

$$\varphi_i(\tau_i) = v_i\,\alpha_i(x_0)\,\beta_i(\tau_i)\,\psi_i(\tau_i), \qquad s_i = \begin{cases} \varphi_i(\tau_i), & b_i = 1, \\ 0, & b_i = 0, \end{cases} \tag{27}$$

where b_i is Build success. The rollout argument of ψ_i accommodates stability windows and native task state machines in addition to a final pose. A factor with no additional requirement is one. In the shared rigid placement profile, a reconstruction initially satisfying its goal must leave that relation and complete a new placement within the same attempt; initial satisfaction alone neither earns success nor necessarily violates α_i. An initially held clip need not demonstrate a preceding pickup.

These factors are a semantic decomposition of the task conditions, not a claim that every backend exposes identical event detectors. Release, support, holding, and stability use the configuration-specific evidence below. Missing essential evidence cannot certify success. The scope of v_i is the assigned checks: for example, RoboDojo validates recorded execution and native mechanism constraints without certifying every contact or collision. Infrastructure and record-integrity errors remain unresolved evaluation outcomes rather than being relabeled as method failures.

Progress for rigid manipulation. The shared rigid evaluator defines an ordered sequence of K_i stages ending in completion. Grasp and release events delimit manipulation attempts; initially held tasks begin with a held-state stage. Let $h_{i,a}$ be the longest ordered prefix of intermediate stages attained within attempt a. For this profile,

$$q_i = \begin{cases} 0, & \text{confirmed build failure or profile-invalid execution,} \\ 1, & s_i = 1, \\ \dfrac{\max(\{0\} \cup \{h_{i,a}\}_a)}{K_i}, & \text{otherwise, for an evaluable rollout.} \end{cases} \tag{28}$$

Success establishes completion even if an intermediate detector did not fire. For unsuccessful executions, evidence cannot be joined across separate attempts; ties in prefix length select the later attempt. A rejected initially solved endpoint can retain credit for observed intermediate stages. Invalidity gates may instead set progress to zero, as in the hand profiles.

Lifting uses acquisition or initial holding, lift, and terminal elevated completion. Placement and insertion use acquisition or initial holding, transport or approach, release when required, and terminal placement or seating. Default intermediate thresholds are 1 cm displacement, 10 cm proximity to a specified horizontal goal, and lift by $\min(3\,\mathrm{cm}, h_i/2)$ for task lift threshold h_i. Persistence defaults to a single observation unless overridden by the instance. Planar pushing uses 2 cm object movement, proximity within twice the goal radius, and completion. These are diagnostic progress thresholds, distinct from the success tolerances below. The insertion profile's coarse approach/release stages do not separately measure alignment, insertion depth, or screw rotation.

Native and deformable progress. RoboDojo retains the task's native graded stage score, taking its maximum over execution and dividing by 100; tasks without a graded score use their binary outcome. For example, a released two-block stack earns 0.15 in the corresponding stacking profile. Robot return home and the final arrangement can remain additional success requirements, so full native progress is not a universal certificate of Task Success. An initially solved arrangement without a new completion receives zero progress. Household profiles use their declared contact, carry/lift, release, and supported or retained-hold stages, with the initial-holding exception and task-specific persistence conditions.

Rope and soft-toy progress is the prefix of three indicators: movement, proximity to the channel or lifting, and terminal success. Movement requires 2 cm for rope and 5 cm for the toy; the second indicator requires a rope node within 5 cm of the channel centre or toy-centroid elevation by 5 cm. The first two indicators search the recorded trajectory and do not independently enforce temporal precedence. Cloth uses capture, a 3 cm lift while held, subsequent release, and the folding predicate; terminal success receives full progress. Thus, stage definitions and weighting belong to the task profile rather than a single universal equal-stage formula.

Terminal position and orientation. Endpoint diagnostics use the final executed state, including the prescribed terminal settling, independently of the demonstration-clock window. If the task names a receiver, the candidate relation $T_r^{-1}T_m$ is compared with the annotated terminal relation. Otherwise, region-based tasks use the difference between candidate and demonstrated displacements from their respective initial positions. Translation error is capped at 100 cm. A verified missing candidate receiver receives this penalty; an unresolved receiver pose is an evaluation error. Tasks without a declared terminal-position anchor have no such metric.

Terminal rotation uses corresponding object orientations or registered axes with the symmetry treatment above. Receiver-relative rotation remains a separate diagnostic where that relation is defined. The reported terminal-orientation measure also supports tasks with no receiver-relative rotational target, including upright-bottle reorientation; these angular quantities are not interchangeable. A

declared missing candidate object receives a 180° terminal-rotation penalty where the reference orientation is evaluable. Native multi-object endpoint measures retain their per-object relations and equal-weight aggregation. None of these endpoint errors replaces the complete success predicate.

C.3.1 Task Conditions and Tolerances

The following conditions instantiate the initial-state, event, terminal-state, and physical-validity factors above. Shared requirements are stated once; source-specific values apply to the named configurations. Release is required only when it is part of the annotated interaction. Human-video clips may end while an object remains held.

Initial relations and completion events. The initial relation is evaluated before the first action. An initially held clip requires the subsequent interaction without an unobserved pickup. For shared rigid placement and the specified native tasks, an initially satisfied goal requires execution to leave that relation and complete a new placement within the same attempt, including required release and stability conditions. This transition belongs to β_i; initial geometric satisfaction alone neither establishes success nor necessarily violates α_i. Object–receiver relations use simultaneous poses when the receiver moves. Partial insertion may continue from its observed initial state, whereas a fully contained start must first exit before receiving credit for a new insertion.

Lifting and retained holding. Lifting requires upward displacement from the annotated initial support relation. DexYCB uses a rise of at least 10 cm and terminal rest, together with the arm- or hand-specific physical checks. Its current height/rest condition does not independently certify sustained terminal holding. RoboDojo pickup also requires at least 10 cm elevation. Household pickup and holding instead use the instance's required elevation or displacement and retained terminal hold, supported by sustained contact or the specified fingertip-distance/relative-motion proxy. Wallet proximity is such a proxy, rather than a force measurement.

Supported, directional, and upright placement. Placement requires the transfer and the designated receiving relation, followed by object-specific release and rest where specified. DROID support is evaluated on the actual bearing surface, with a 1 cm contact tolerance and a 5 mm footprint margin; receiver rim height alone does not establish support. Directional placement requires at least 3 cm of horizontal progress in the demonstrated direction. Side-placement uses the declared receiver-relative relation without imposing an unrelated absolute endpoint. Upright-bottle placement requires the resolved axis to lie within 30° of the support normal; a directed neck points upward, while an unsigned axis is permitted only for geometrically indistinguishable ends. HOI4D placement uses the instance's goal region, orientation, and support-height conditions. HOT3D and OakInk2 use the annotated region, tray, or container relation, retaining any allowed held endpoint. For human-video configurations, the assigned arm or hand validity checks also apply.

Containment and extraction. DROID container placement requires at least 5% of 3,000 deterministic object-surface samples to occupy free cavity space at least 2 mm below the receiver rim, together with required release and rest. Vessel material and handle holes are excluded, and task-required lid geometry is retained. Long objects may protrude: this rule is not full containment. Removal to a support requires a supported final state with at least 95% of the object vertices outside the original container's horizontal footprint, followed by release and rest. Household initially held insertion requires outside-to-inside evidence, cavity containment, and a released stable endpoint. Air-fryer basket extraction requires the entire inner basket to clear the opening and rest after release; partial withdrawal fails.

Furniture placement and assembly. Lampshade placement distinguishes capping a lamp from setting the shade aside. Capping checks that the lamp top lies inside the hood. Set-aside checks support, separation, and the annotated upright or lying condition. Both require the relevant completion transition, release, and rest. General part seating, including bulb placement, uses a relative translation tolerance of 3 cm and orientation tolerance of 30°, with the declared symmetry. Corner-leg insertion additionally requires the demonstrated corner, tilt within 30°, distance within 7 cm of that corner, and a foot height from 5 cm below to 1.5 cm above the receiver top. Central pedestals use the annotated centre and its specified lateral tolerance. Drawer seating checks tilt, footprint, vertical

extent, and height within 3 cm of the demonstrated slot for a multi-slot receiver. Door assembly uses the demonstrated axis, normalized location on the receiving body, and interface proximity. These tests use the candidate receiver geometry and the executor's recorded retention convention. Geometric seating does not certify screw engagement, a newly formed hinge, or load-bearing strength.

Multi-object arrangements and stacks. RoboDojo conditions retain task-specific logical combinations and require robot return home where specified. All designated objects must attain their semantic assignments, and an initially solved arrangement must leave the object-level goal and be completed again; robot motion alone is insufficient. Block stacking uses a horizontal tolerance of 1.5–2 cm, according to the task, and language-conditioned variants preserve the named order. Bowl stacking uses 4 cm horizontal tolerance, all bowls upright within 45°, and the base within 7°. Tower construction checks the prescribed base, middle, and top arrangement and orientations. Digit arrangement requires descending order on the designated mats, proximity within 2 cm, and digit-specific orientation within 45°. Category sorting requires the assigned objects to lie in their category baskets below the rim; size sorting requires the prescribed doll order with a lateral spread bound of 3.5 cm. These spatial tests do not measure inter-object support force, and intermediate reward thresholds are not substituted for success conditions.

Insertion, exchange, and ordered cover operations. RoboDojo tube insertion requires all three tubes to reach 4.5 cm insertion depth with their axes within 30°; charger insertion requires 1.5 cm and 10°. Both retain the robot-home requirement and the initially-solved transition guard. Exchange targets are fixed from the two T-shaped objects' initial states; the swapped positions must be within 2 cm and orientations within 3°, followed by robot return home. Cover-and-uncover requires covering left, middle, then right, followed by uncovering red, green, then blue. All six stages and robot return home are required. Cups must be inverted within 10°, and displacement of a protected block by 5 cm triggers failure. The recorded event sequence is necessary to establish this task; a terminal image cannot do so.

Planar pushing, rope routing, and soft-toy packing. Planar pushing requires the T-shaped body's centre within 4 cm of the goal and its planar axis within 15°, with terminal rest. Gripper release is inapplicable; an initially satisfied goal still requires a new completion. Rope and soft-toy tasks require the terminal predicate to be false initially. Rope routing requires a connected rope path to cross both physical channel faces for at least 0.5 s, with colliding channel geometry and the robot/gripper checks below. Connectivity does not measure retention force. Soft-toy packing requires at least 85% of surface points to project inside the cavity using the 1.05 lateral factor, the centroid between floor and rim, and persistence for 0.5 s. Among surface points within that expanded horizontal footprint, none may lie more than 2 mm below the physical floor. This containment predicate does not certify the entire object against floor penetration.

Cloth folding. The initial visible surface must not already satisfy the target; execution must produce the required surface and footprint change. The median distance from observed to simulated surface points must be at most 3 cm, and the footprint-change ratio must be within $\pm 25\%$ of the observed change. Evaluation uses camera-visible surfaces and their validity masks. These conditions do not independently establish flatness, crease quality, or absence of crumpling.

Physical validity and evidence scope. All conditions consume actual execution records under the robot-actuation and passive-object restrictions of Section A.3. Bimanual-hand checks allow robot penetration at most 6 mm, initialization object displacement at most 15 mm, nonadjacent self-overlap diameter at most 4 mm, and finger-joint limit excess at most 0.001 rad. Joint-speed exceedances are retained as diagnostics. MuJoCo execution preserves the supplied articulated-gripper mechanics and rejects coupling residuals above 0.001 rad. Native configurations apply the checks exposed by their assigned executors. RoboDojo records execution integrity and mechanism constraints without a complete contact/collision validity certificate; the source-specific task evidence above determines release, holding, support, and containment. Applicable validity failures preclude task success. A completed simulator process alone does not establish physical validity. The geometry-based proxies identified above should be interpreted within their stated scope.

A grasp event alone does not establish sustained force closure, and coarse transport or release stages do not establish fine insertion or screw rotation. Release, stability, containment, and physical validity

therefore contribute distinct evidence to the task decision. An admissible execution can complete a task in a geometrically inaccurate scene, while a favorable endpoint can fail because a required event or validity condition is absent.

C.4 Aggregation and Reporting

Evaluation population. The main evaluation uses a fixed set of 222 reconstruction instances across 39 task families for all coding-agent methods and HAR. All instances have applicable task specifications and contribute to Build Rate, Task Success, and Task Progress. Confirmed build failures receive zero for these three metrics. Repeated construction or evaluator-repair runs do not create additional test instances: each reported result uses the selected frozen package and its resolved execution record. Applicability of individual geometric and motion metrics depends on the available reference annotations, with coverage reported separately.

Family macro-averages. For metric μ, let $\mathcal{I}_{f,\mu}$ contain the instances in semantic family f for which the reference annotation supports that metric. For Build Rate, this is the full family manifest. Let $\mathcal{F}_\mu$ be the families with nonempty $\mathcal{I}_{f,\mu}$. For $z_i \in \{b_i, s_i, q_i\}$, the reported percentage is

$$\overline{z} = \frac{100}{|\mathcal{F}_\mu|} \sum_{f \in \mathcal{F}_\mu} \frac{1}{|\mathcal{I}_{f,\mu}|} \sum_{i \in \mathcal{I}_{f,\mu}} z_i. \tag{29}$$

Task Progress is thus averaged over the same task-applicable population as Task Success, with build failures retained. Its per-instance range is $[0, 1]$, converted to percent for reporting.

Continuous errors condition on successful builds and applicable reference measurements. Let $\mathcal{J}_{f,\mu} \subseteq \mathcal{I}_{f,\mu}$ be that eligible subset after evaluation records have been resolved, and let $\mathcal{F}_\mu^+ = \{f : |\mathcal{J}_{f,\mu}| > 0\}$. The continuous-error macro-average is

$$\overline{e}_\mu = \frac{1}{|\mathcal{F}_\mu^+|} \sum_{f \in \mathcal{F}_\mu^+} \frac{1}{|\mathcal{J}_{f,\mu}|} \sum_{i \in \mathcal{J}_{f,\mu}} e_{i,\mu}. \tag{30}$$

A family with no eligible continuous comparison has an unavailable family mean and is omitted from this conditional macro-average. Consequently, the continuous-error population can differ across methods. Each such aggregate is accompanied by its eligible instance and family counts and coverage relative to the applicable manifest; comparative analyses identify the jointly evaluable subset. If no family is eligible, the aggregate is unavailable. Explicit metric penalties, including those for missing candidate geometry, remain numerical observations in these averages.

Failure and missing-data policy. Structural inapplicability, reconstruction failure, and unresolved evaluation are separate outcomes. An inapplicable metric is not replaced by zero. A confirmed build failure contributes zero to Build, Success, and Progress wherever each is applicable. A completed execution that fails the task remains evaluable for continuous measurements. Missing required outputs, incompatible evaluation revisions, and simulator or dependency failures attributable to infrastructure must be resolved before reporting; they are not grounds for dropping an otherwise eligible comparison. The retained evaluation record distinguishes these cases and binds the submission, annotations, evaluator version, and execution settings.

Uncertainty estimates. The reported HAR confidence intervals use 10,000 source-cluster bootstrap replicates, random seed 20260923, and the 2.5th and 97.5th percentiles. A source cluster retains all target configurations of a source clip together; FurnitureBench clips from the same original episode are also clustered together, and RoboDojo uses the source demonstration record. For each metric, clusters are stratified by their set of eligible semantic families. Clusters are sampled with replacement within these strata, and all eligible instances in a sampled cluster receive the same bootstrap multiplicity. Each replicate recomputes the equal-family macro-average in Eqs. 29–30.

These are pointwise intervals over the observed source population, not estimates of repeated-run variability on a fixed instance. Single-method intervals do not constitute a paired test of method differences. Source-, simulator-, and embodiment-specific breakdowns retain the same metrics definitions and report their sample counts. Geometric fidelity, executed-motion fidelity, and functional correctness are reported separately as component metrics; V2WScore provides an additional aggregate summary as defined in Section C.6.

C.5 Supplementary Effector-Motion Analysis

Scope and paired observations. We supplement object-centric dynamics evaluation with a comparison of source effector motion and the motion of the robot executing the simulated interaction. We use the 40 reference instances satisfying the task-success criterion in the original evaluation: 16 DROID, 13 DexYCB, 5 HOT3D, 3 HOI4D, and 3 OakInk2 instances. These comprise 35 unique videos; four DexYCB videos and one OakInk2 video contribute both arm and bimanual executions. Each video–embodiment pair is one observation. Selection is fixed by the original evaluation and does not depend on effector-motion agreement.

On DROID, both sides use gripper tool-center-point (TCP) poses. On human-video sources, we compare the annotated primary-hand wrist with the corresponding simulated robot hand root: the Panda hand frame for arm executions and the primary Wuji palm root for bimanual executions. The latter is the hand-root frame used as the counterpart of the human wrist in the retargeting convention, rather than a fingertip center. For the OakInk2 bimanual instance, we apply the inverse of the independently fixed MANO-to-Wuji wrist transform to the simulated palm pose. No transform is fitted to improve the present agreement results. All trajectories are expressed in the shared metric, gravity-aligned scene frame.

These human-to-robot comparisons characterize coarse effector motion across embodiments. They do not assume identical anatomy or establish agreement of all arm joints or finger articulation. Human sources have annotated hand motion, not measured robot-arm trajectories. The simulated runs are human-assisted references whose actions use source information; the analysis describes motion preservation in reference execution rather than autonomous behavior recovery or downstream policy-training gains.

Recorded execution and comparison windows. For DROID, real TCP poses are sampled at 7.5 Hz. We replay the 16 originally successful references to log actual TCP poses and elapsed control-step time at 30 Hz, including additional grasp-closure waits. These are actual link poses, not commanded targets. The replay window starts after reset and initial settling and ends at the last action, excluding additional terminal settling. The original success labels define the subset; task success is not re-evaluated for these logging replays.

For the 24 human-source instances, actual robot hand-root poses are taken from the selected reference executions. Twenty-one have archived pose logs with simulation timestamps; the remaining three have execution records with a verified fixed 30 Hz clock and no adaptive waiting. The archived action-row poses agree exactly with the corresponding selected execution logs. We exclude prelude and terminal-settling phases. Source wrist annotations use their recorded timestamps, or their documented sampling rate when timestamps are absent: 15 Hz for HOI4D and 30 Hz for the other human sources.

We restrict each human-source comparison to the interval between the first and last valid source wrist observations and crop the simulation at the corresponding action-row times. Six DexYCB instances have three to five unobserved leading frames; these frames are removed from both sides, rather than treated as measured stationary motion. There are no missing wrist observations inside the retained intervals. Neither trajectory is rescaled in time or fitted spatially to the other.

Preprocessing and motion interval. Positions are linearly interpolated and orientations interpolated by SLERP on a 30 Hz grid. Translational velocity is estimated using a nine-sample, second-order Savitzky–Golay derivative. Let p_k, R_k, and v_k denote position, orientation, and estimated velocity at t_k. Motion onset k_{on} is the first sample of six consecutive samples satisfying $\|p_k - p_0\|_2 > 0.01\,\mathrm{m}$. The end index k_{end} is the start of the final run of at least nine samples with speed below 0.01 m/s through the recording boundary, beginning no earlier than onset. If this terminal run is absent, we use the last sample.

The interval is determined separately for each trajectory. Terminal quiescence is unobserved in 39 of the 40 source trajectories and all 40 simulated trajectories; these cases remain included. Motion duration therefore denotes the observed interval within the recording, including intervening pauses, and does not necessarily measure time to a complete stop. Human-source reference executions also follow the source action cadence, so temporal correspondence should be interpreted in that context.

Motion statistics. We compute the following features separately for source and simulated effector motion, with z denoting the vertical coordinate:

$$\text{Motion duration (s)} = t_{k_{\text{end}}} - t_{k_{\text{on}}}, \tag{31}$$

$$\text{Path length (cm)} = 100 \sum_{k=k_{\text{on}}}^{k_{\text{end}}-1} \|p_{k+1} - p_k\|_2, \tag{32}$$

$$\text{Peak speed (cm/s)} = 100 \max_k \|v_k\|_2, \tag{33}$$

$$\text{Lift height (cm)} = 100 \left(\max_k p_{k,z} - p_{0,z} \right), \tag{34}$$

$$\text{Net displacement (cm)} = 100 \|p_{k_{\text{end}}} - p_0\|_2, \tag{35}$$

$$\text{Net rotation } (^\circ) = \frac{180}{\pi} \arccos\left[\text{clip}\left(\frac{\text{tr}(R_0^\mathsf{T} R_{k_{\text{end}}}) - 1}{2}, -1, 1 \right) \right]. \tag{36}$$

Peak speed and lift height use the full retained interval. Path length uses onset to end; net displacement and net rotation use the initial retained pose as their reference. Lift height can be zero when the effector starts above the subsequent interaction. Net rotation is the full SO(3) initial-to-end angle, without object-symmetry reduction; it is not accumulated rotation or framewise orientation error. These six summaries measure selected motion characteristics rather than complete trajectory equivalence.

Paired visualization. All six features have 40 valid pairs. Figure A6(a) plots paired source and simulated measurements with common axis scales, an equality line, and Spearman correlation pooled over all instances. Panel (b) shows $(y_i - x_i)/s_x$, where x_i and y_i are source and simulated values and s_x is the sample standard deviation of the 40 source values. Boxes show medians and interquartile ranges; whiskers extend to observations within 1.5 interquartile ranges. The shaded ± 0.25 SD band is a visual reference. Arrows count points outside the displayed $[-1.5, 1.5]$ SD range; these points remain included in every summary. The right column reports $\text{median}_i\, |y_i - x_i|$ in the original units.

C.6 V2WSCORE

V2WScore summarizes functional correctness, geometric fidelity, and executed-motion fidelity, with successful scene construction as a prerequisite. Build acts as a multiplicative gate, rather than a fourth equally weighted component. All normalization parameters and metric applicability rules are shared across methods, including HAR; a method's score does not depend on the outputs or ranking of other methods.

Per-instance normalization. For a nonnegative error $e_{i,k}$ of instance i under metric k, define its quality as

$$h_k(e_{i,k}) = \max\left(0, 1 - \frac{e_{i,k}}{\lambda_k}\right). \tag{37}$$

Zero error receives quality one; errors at or above λ_k receive zero. These physical-unit scales were introduced after the initial reference runs and are shared across methods. They are scoring parameters, distinct from task-success thresholds. Normalization is performed on each instance before averaging: applying Eq. 37 to a reported column mean generally gives a different result because of clipping and metric-specific coverage.

Component metrics and scales. Geometry uses Scene CD, object Shape CD, and size error, each with $\lambda_k = 10$ cm. Where the annotation supports deformable geometry, visible-scene surface distance, initial-object surface distance, and world-axis-aligned bounding-box size error replace the corresponding rigid measurements, using the same scales. The replacement follows the reference annotation and avoids counting both representations of the same geometric quantity.

Dynamics uses translation APE, rotation APE, and translation RPE from the method's own physical execution, with scales of 20 cm, 90°, and 10 cm, respectively. Deformable trajectory surface distance uses 20 cm. For partial observations, point-position and point-displacement errors use 20 cm and 10 cm; observable long-axis, surface-normal, and symmetry-aware frame errors each use 90°. Only annotation-supported measurements enter the score, retaining their definitions in Section C.2. When

applicable, centroid-motion and deformable-surface errors both contribute because they measure different aspects of motion.

Component weighting and family aggregation. Let $\mathcal{K}_i^G$ and $\mathcal{K}_i^D$ be the nonempty sets of applicable geometry and dynamics metrics. For a successfully built reconstruction, let $s_i \in \{0, 1\}$ be Task Success and $q_i \in [0, 1]$ be Task Progress. Define

$$F_i = \frac{s_i + q_i}{2}, \qquad G_i = \frac{1}{|\mathcal{K}_i^G|} \sum_{k \in \mathcal{K}_i^G} h_k(e_{i,k}), \qquad D_i = \frac{1}{|\mathcal{K}_i^D|} \sum_{k \in \mathcal{K}_i^D} h_k(e_{i,k}). \tag{38}$$

Functionality weights success and progress equally; geometry and dynamics each average their applicable normalized measurements. With Build indicator b_i, the instance score is

$$\begin{aligned} z_i &= \begin{cases} (F_i + G_i + D_i)/3, & b_i = 1, \\ 0, & b_i = 0, \end{cases} \\ \mathrm{V2WScore} &= \frac{100}{|\mathcal{F}_V|} \sum_{f \in \mathcal{F}_V} \frac{1}{|\mathcal{I}_{f,V}|} \sum_{i \in \mathcal{I}_{f,V}} z_i. \end{aligned} \tag{39}$$

Here, $\mathcal{I}_{f,V}$ is the fixed score-eligible population in family f, and $\mathcal{F}_V$ contains families with a nonempty population. Eligibility requires applicable task annotations and at least one geometry and one dynamics measurement. Equal family weights prevent large families from dominating the score. A score of 100 requires successful construction, full functional credit, and zero error in every included fidelity measurement.

Applicability and incomplete evaluation. Eligibility and constituent metrics are determined by the reference annotations, independently of candidate delivery or measurement availability. Confirmed build failures remain in the denominator with $z_i = 0$; their unmeasured geometric and dynamic errors are not assigned fictitious values. Missing applicable measurements, unresolved evaluator failures, or incompatible evaluation versions leave the complete-manifest score unavailable. An average over available records or a diagnostic that substitutes zeros for unresolved evaluations is a different statistic and must be identified separately. Comparisons require the same instance population, normalization parameters, applicability rules, and metric versions. These conditions also apply when comparing coding agents with HAR.

D Additional Experiments

Can agent-built worlds support experience beyond the source video? We test whether successful reconstructions support new configurations using nine source videos. For each evaluated model, we use its successful initial reconstruction of each source video. Each reconstruction is tested on 20 layouts formed by translating either the manipulated object or the target in two predefined table directions at five magnitudes (2, 5, 10, 15, and 20 cm), yielding 180 paired layout evaluations per model. For each source, both directions are verified across all five magnitudes using the human-assisted reconstruction before evaluating candidate layouts. The same videos and displacements are used across models. This experiment therefore measures reuse conditional on successful initial reconstruction and the prescribed layout selection.

Direct reuse executes the original robot behavior without code changes, retaining any existing feedback or planning logic. *Agent adaptation* starts from the same original submission and displaced layout, and allows the corresponding coding agent to revise robot behavior using simulation feedback. The agent receives the source video, original submission, and layout change. The evaluator fixes the displaced scene, assets, physical parameters, and task criteria throughout adaptation. Both settings use the same runtime observation interface. Adaptation starts with a fresh context for each layout and stops after 10 rollout attempts or 10 minutes; the final behavior is evaluated from a clean reset.

Figure A6 shows that successful reconstructions can support executions beyond the observed layout. Across the three models and 540 paired layout evaluations, direct reuse succeeds in 97.22% of cases (525/540), increasing to 99.26% (536/540) after adaptation. Adaptation recovers 86.67% of direct-reuse failures (13/15), while turning 0.38% of initially successful cases into failures (2/525), yielding

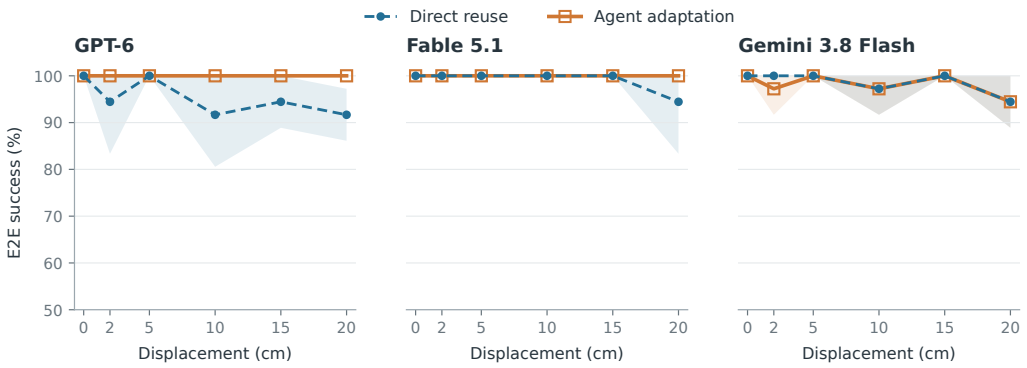

Figure A6: **Robot Experience Beyond the Source Video.** Direct reuse and agent adaptation across translated object and target layouts. The same nine source videos (three per task family) contribute at every displacement, with 180 paired layout evaluations per model. Each source uses the same two table directions and both moved-object roles at every magnitude. Both directions were verified using the human-assisted reconstruction jointly across all five magnitudes before new candidate evaluations. Success is averaged within video and then across task families. Shading shows 95% paired source-video bootstrap intervals within family. The 0 cm points are verified original baselines without adaptation.

a net gain of 11 successful executions. At the largest displacement of 20 cm, success increases from 93.52% to 98.15%, although performance is not monotonic in displacement.

The benefit varies across models. GPT-6 improves from 94.44% to 100%, and Fable 5.1 from 98.89% to 100%, recovering all ten and two of their respective direct-reuse failures. Gemini 3.8 Flash decreases slightly from 98.33% to 97.78%: it recovers one failure but introduces two new failures. Adaptation therefore provides a means of repairing behavior under changed layouts, but does not guarantee improvement. Across the 540 evaluated adaptation sessions, including unsuccessful outcomes, the mean cost is 1.06 rollout attempts and 3.26 minutes per layout, excluding additional infrastructure and recovery launches.

These results demonstrate substantial layout reuse within the selected cohort, with adaptation providing a modest aggregate gain over an already strong direct-reuse baseline. The evidence is conditional on successful initial reconstruction and reference-verified translations of the same nine source tasks; it does not establish generalization to new task families or improved downstream policy learning.

E Cost Analysis

All methods were assigned the same nominal per-episode budgets: a 3-hour wall-clock limit, a $60 API-spend threshold, and at most three package submissions (Appendix A.5). The benchmark-managed executor directly enforces these limits for the seven OpenCode configurations. Fable-5.1 and GPT-6 Astra run through their native agent harnesses under the corresponding limits, while retaining framework-specific checkpoint and refinement semantics. Table A11 reports per-episode cost. For the OpenCode configurations, API cost is the provider-billed amount reported through OpenRouter. For Fable-5.1 and GPT-6 Astra, it is the list-price equivalent computed from the native harness token counts; comparable API-call counts were not recorded. Evaluation uses one NVIDIA L40S per episode for reference execution, simulation of the delivered package, and metric computation.

Table A11: **Per-episode cost of each method.** API cost is in USD; agent time is wall-clock time from the first model call to termination; evaluation time is GPU-minutes on one L40S. Estimated totals extrapolate the reported per-episode means over all 222 benchmark instances.

	API cost (USD)				Agent time	API calls	Eval. GPU
Method	mean	median	max	est. total	mean (min)	mean	mean (min)
Fable-5.1[†]	14.03	12.90	45.19	3,115	37.5	–	10.1
GPT-6 Astra[†]	6.40	3.91	24.58	1,421	14.6	–	9.7
Claude Opus 5	13.83	10.03	61.07	3,070	32.6	59	8.7
Kimi K3	5.51	4.81	13.27	1,223	82.2	72	7.4
GPT-5.6 Sol	1.39	1.00	7.89	309	3.7	17	8.2
DeepSeek V4.1 Flash	0.23	0.21	0.62	51	13.9	53	8.3
Gemini 3.8 Flash	1.53	1.02	8.15	340	18.6	100	7.5
Qwen3.8 Max	0.67	0.60	2.26	149	26.0	34	8.4
GLM-5.3 Flash	0.31	0.23	1.46	69	53.0	60	8.9
All methods	4.88	–	61.07	9,746	31.3	–	8.6

[†]List-price equivalent computed from token counts recorded by the native harness; API-call counts were not recorded. Estimated totals are the displayed means multiplied by 222 and rounded to the nearest dollar.

Running all nine methods over the 222-instance benchmark corresponds to approximately $9.7k in API spend, 1,044 agent-hours, and 286 L40S GPU-hours of evaluation at the reported per-episode averages. API spend is concentrated in Fable-5.1 and Claude Opus 5, which each cost about $14 per episode and together account for approximately 63% of the estimated total. The flash-tier configurations average less than $2 per episode. Cost does not track score monotonically: GPT-6 Astra achieves the second-highest V2WScore at about $6 per episode, while Kimi K3 uses the most agent time, averaging 82 minutes per episode, without a corresponding score advantage.

No episode exceeds the 3-hour wall-clock allowance; the longest recorded episode lasts 164 minutes. One Claude Opus 5 episode records $61.07 because delayed provider reporting can overshoot the $60 observed-spend threshold. Evaluation cost varies comparatively little across methods, from 7.4 to 10.1 GPU-minutes per episode. At the reported average, adding a method over all 222 instances requires approximately 32 L40S GPU-hours of evaluation, in addition to its generation spend and agent time.

F Case Studies Across Demonstration Sources

We examine six demonstrations to explain how geometric fidelity, executed object motion, and task completion can disagree. The cases cover real robot demonstrations, egocentric human video, and simulation-generated robot demonstrations. Three detailed reconstructions use Claude Opus 5; two further demonstrations permit a comparison of all nine coding models on the same instance; and a native stacking task is examined under two target embodiments. Cases were selected for their diagnostic value, rather than to estimate the prevalence of particular errors or rank models.

Each result refers to a particular submitted scene, its own robot behavior, and the recorded physical execution. Scores are taken from that submission's retained evaluation record. The accompanying evidence identifies the instance, model, package, scoring record, and execution used for each analysis. Image sequences are arranged by task phase unless an explicit time axis is shown; phase correspondence does not imply equal execution speed. Scene Chamfer Distance (Scene CD), translation absolute pose error (APE), and Task Success retain their distinct meanings from Appendix C.

F.1 Robot Demonstrations: Pose and Task-Critical Geometry

The first two cases reconstruct DROID demonstrations with a Panda arm and Robotiq gripper. They illustrate two different limitations of interpreting a favorable geometric or endpoint measurement as complete reconstruction quality.

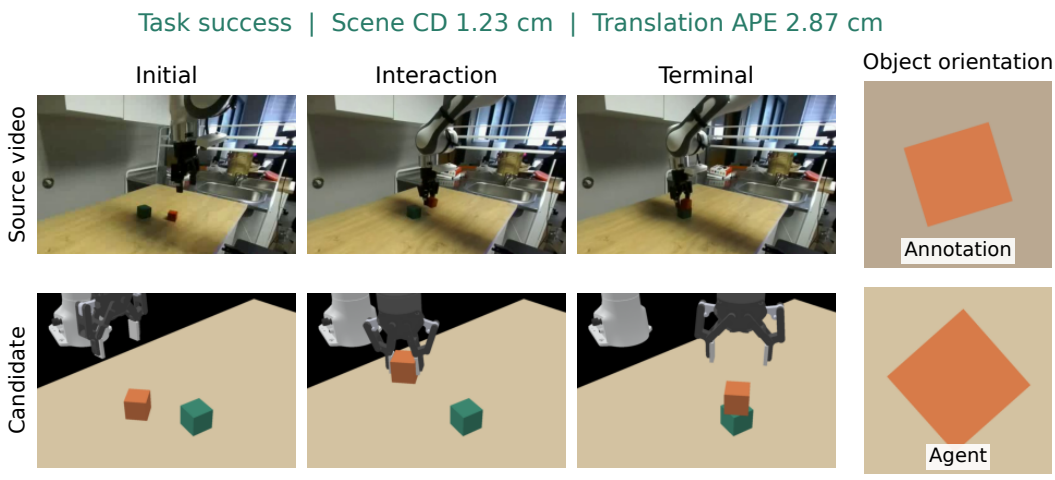

Figure A7: **Task completion despite pose disagreement.** DROID block stacking with a Panda–Robotiq embodiment. The source and candidate execution are shown at three task phases. The top-view inset compares the submitted and annotated block orientations. Successful stacking coexists with a large object-frame rotation error; the rigidly aligned shape score does not measure the submitted world orientation.

Task completion despite pose disagreement. In the first demonstration, an orange block is placed on a green block. The Opus reconstruction completes this interaction and reaches all four recorded progress stages. Its terminal relative-position error is 1.08 cm, with translation APE of 2.87 cm and translation RPE of 1.80 cm. Geometry also gives small surface errors: Scene CD is 1.23 cm and object Shape CD is 0.77 cm. Nevertheless, initial centre error is 5.92 cm, size error is 1.31 cm, and rotation APE is $144°$.

Figure A7 shows why these quantities need separate interpretation. The submitted object is a 4.5 cm cube, whereas the annotated dimensions are approximately $3.5 \times 3.8 \times 4.1$ cm. Its submitted yaw is $-138.4°$, compared with $17°$ in the annotation. Reducing the cube's yaw modulo its quarter-turn symmetry gives $41.6°$; even this surface-equivalent representative differs from the annotated orientation. Shape CD is evaluated after rigid alignment, so it does not measure this world-pose discrepancy. Scene CD measures registered visible surfaces and can remain small for a compact, nearly symmetric object. Rotation APE instead evaluates the corresponding object frames under the annotated symmetry, which does not remove the reported disagreement.

Successful stacking therefore establishes that the submitted behavior can manipulate the submitted block and satisfy the specified relation. It does not establish accurate object dimensions, initial placement, or orientation history. Conversely, the large angular discrepancy does not negate the observed completion of an orientation-tolerant stacking task.

A plausible endpoint without the required receiving geometry. In the second demonstration, a blue block is inserted through an opening in the top of a shape-sorting box. The reconstruction grasps, lifts, and releases the block, which rests only 1.12 cm from the annotated receiver-relative goal. Translation APE is 6.75 cm. Despite this favorable endpoint, Task Success is zero and Task Progress is 75%.

The submitted receiver is an open container with 8 mm walls, rather than a box with an apertured top (Figure A8). Its top-face coverage is 0.23, below the required 0.50; the annotated box has coverage 0.85. The receiver therefore lacks a task-defining structure, even though the block reaches an admissible relative position. The coverage test verifies the presence of upper receiving geometry; it should not be interpreted as an independent measurement of passage through a particular aperture.

Scene CD is 4.33 cm, while the manipulated block's Shape CD is 0.70 cm. The weighted missing-geometry fraction is 0.50. Here that diagnostic does not mean that the block was omitted: the delivered block exceeds the surface-distance threshold after registration. These measurements describe different

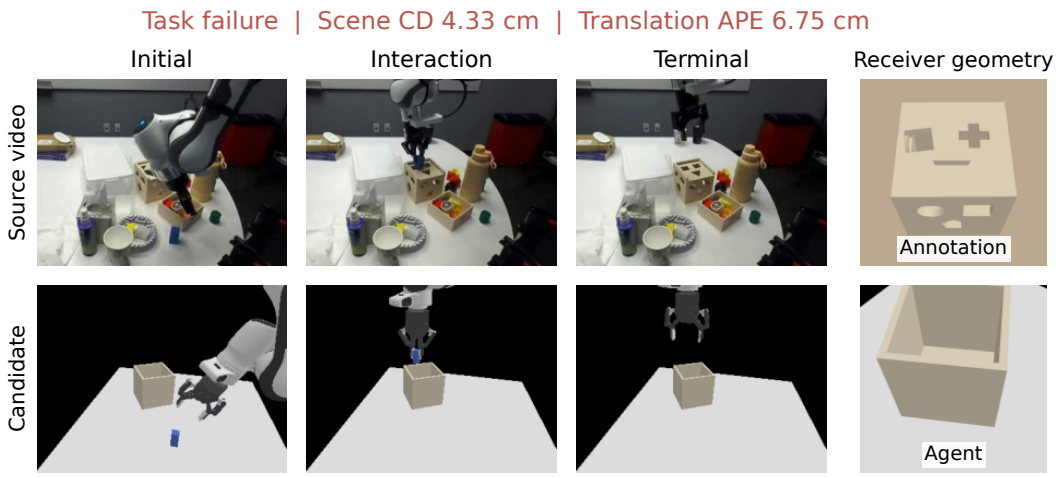

Figure A8: **Endpoint proximity does not establish the required interaction.** DROID shape-sorter insertion. The block is transferred into the submitted open container, but the receiver lacks the apertured top visible in the annotation. The final relative-position error passes its tolerance; top-face coverage fails its separate requirement. The source and candidate rows show task phases, rather than equal timestamps.

aspects of the reconstruction. Surface discrepancy characterizes geometric correspondence, whereas the task-specific receiver condition determines whether the intended insertion environment was constructed. An endpoint tolerance alone would miss this failure.

F.2 Nine Models on the Same Instances

We next compare Claude Opus 5, GPT-5.6, Gemini 3.8 Flash, GLM-5.3 Flash, Kimi K3, DeepSeek V4.1 Flash, Qwen3.8 Max, GPT-6 Astra, and Fable 5.1 on two shared instances. All nine have a completed physical execution and evaluation record for both cases. Comparisons below hold the demonstration, target embodiment, and task specification fixed within each instance. They concern these particular submissions, rather than model-level success rates. Figures A9 and A10 show every model's reconstruction, with the same model order in both grids. Each panel pairs the last recorded review-video frame with the submission's geometric error, motion error, and complete evaluation decision. Figure A11 complements these views with the task-specific decision variables.

Receiver-relative completion in bowl nesting. The DROID task is to place a black bowl *inside a peach-coloured bowl*. The receiver is therefore another bowl, not the tabletop. Opus, GPT-5.6, and DeepSeek complete the task, with terminal relative-position errors of 1.36, 2.47, and 4.47 cm, respectively. Gemini, Qwen, Kimi, GLM, Astra, and Fable fail, with errors of 6.14, 16.16, 23.57, 29.43, 25.02, and 25.73 cm. Figure A11(a) shows these errors alongside the 5 cm positional tolerance. The plotted distance is one component of the decision; completion also uses the recorded interaction, release, and terminal-rest conditions.

Geometric correspondence does not determine these outcomes. Fable has a lower Scene CD than Opus, 2.40 versus 2.87 cm, but fails to establish the receiving relation. Their translation APE values are 18.02 and 8.10 cm, respectively. GPT-5.6 succeeds despite a Scene CD of 8.10 cm and translation APE of 12.23 cm. These examples show that geometric fidelity and task completion can disagree within a fixed instance; they do not establish statistical independence between the two dimensions.

Terminal lift versus physically valid execution. The second instance is a DexYCB demonstration of lifting a can, retargeted to a Franka arm. Opus and Kimi succeed, with terminal height increases of 19.60 and 18.84 cm, above the 10 cm requirement. Their executions also pass the applicable physical checks. Their translation APE values are 4.71 and 6.82 cm, and their Scene CD values are 6.63 and 5.45 cm.

Qwen and Astra satisfy the ungated terminal task condition: their height increases are 16.90 and 20.67 cm, and the object is at rest. Both nevertheless receive zero Task Success. Qwen violates the inverse-kinematics feasibility check; Astra violates the robot-penetration check. These are distinct recorded violations. Astra's Scene CD is 2.48 cm and translation APE is 6.31 cm, illustrating why neither favorable surface correspondence nor a close object trajectory replaces physical validity.

DeepSeek and Fable instead pass the physical checks but do not complete the lift. The object's terminal height changes by approximately -0.20 cm and 0.006 cm, respectively. Fable's Scene CD is 1.75 cm, the lowest among these nine reconstructions, while translation APE is 8.39 cm. Its geometry is therefore relatively accurate under the recorded surface metric, but its behavior does not achieve the required state change. GPT-5.6, Gemini, and GLM record both robot penetration and infeasible inverse kinematics. The first two also have extreme negative terminal displacements, approximately -16.0 and -69.4 m; these are abnormal executed states, not plausible lifting motions. GLM's terminal rise is only 1.04 cm and does not meet the height requirement.

Figure A11(b) distinguishes valid but unsuccessful execution from execution rejected by physical checks. In particular, the Qwen and Astra outcomes demonstrate that the validity condition changes the success decision even when the ungated terminal task condition passes. The height and rest tests alone do not independently certify sustained grasp contact; claims about holding must rely on the corresponding execution evidence.

F.3 Egocentric Human Demonstrations under a Robot Embodiment

Task completion with spatial and temporal mismatch. A HOT3D head-camera recording shows a person moving a food package into a bowl. Opus reconstructs the scene and executes the interaction with a Franka arm. The execution passes the applicable physical checks and all five recorded progress stages. The package finishes 2.7 cm from the axis of the submitted bowl, whose rim radius is 14.0 cm. Its terminal relative-position error against the demonstrated relation is 6.88 cm. This error is a diagnostic rather than the bowl-containment success threshold.

The successful interaction occurs in a displaced reconstruction. Scene CD is 6.79 cm, initial object-centre error is 28.2 cm, and the recorded scene-registration translation has magnitude 28.4 cm. The submitted tabletop lies approximately 26 cm above the annotated support. The side view in Figure A12 exposes this disagreement, which is less apparent in the original camera view. The submitted package is also initially 38.6 cm from its bowl in the support plane, compared with 55.3 cm in the annotation. A single global registration cannot remove this relative-layout discrepancy.

The weighted missing-geometry fraction is 0.49, although the package and kettle are present in the submission. Their registered median surface residuals are approximately 17.1 and 5.1 cm, so they exceed the coverage threshold. As in the blue-block case, this diagnostic describes inadequate surface correspondence rather than necessarily absent semantic entities.

The execution clock explains an additional discrepancy. Translation APE is 41.2 cm, rotation APE is 167°, and translation RPE is 10.6 cm. The submitted behavior contains 37 command rows at 0.2 s intervals, spanning 7.2 s, whereas the demonstration spans 1.6 s. On the demonstration clock, much of the demonstrated transport is compared with an execution that has not yet performed the corresponding motion. Figure A12 plots displacement against the separate recorded clocks, without stretching either trajectory. The recorded duration-normalized translation error is lower, 18.1 cm, but remains a diagnostic outside the reported APE.

The instance therefore demonstrates two independent limitations of task completion as a reconstruction criterion. The containment condition can be satisfied relative to the submitted bowl even when the scene is globally displaced, and the same interaction can be completed at a substantially different rate. Geometry and motion fidelity expose those discrepancies separately.

F.4 Native Robot Demonstrations across Target Embodiments

The final demonstration is a RoboDojo three-block stacking task, assigned separately to dual ARX X5 arms and dual xArm7 arms. Both configurations use the same source video and semantic objective. We examine the formally admitted instances and the submitted executions of Astra and Fable. Figure A13 shows the demonstrated arrangement and the four recorded terminal states.

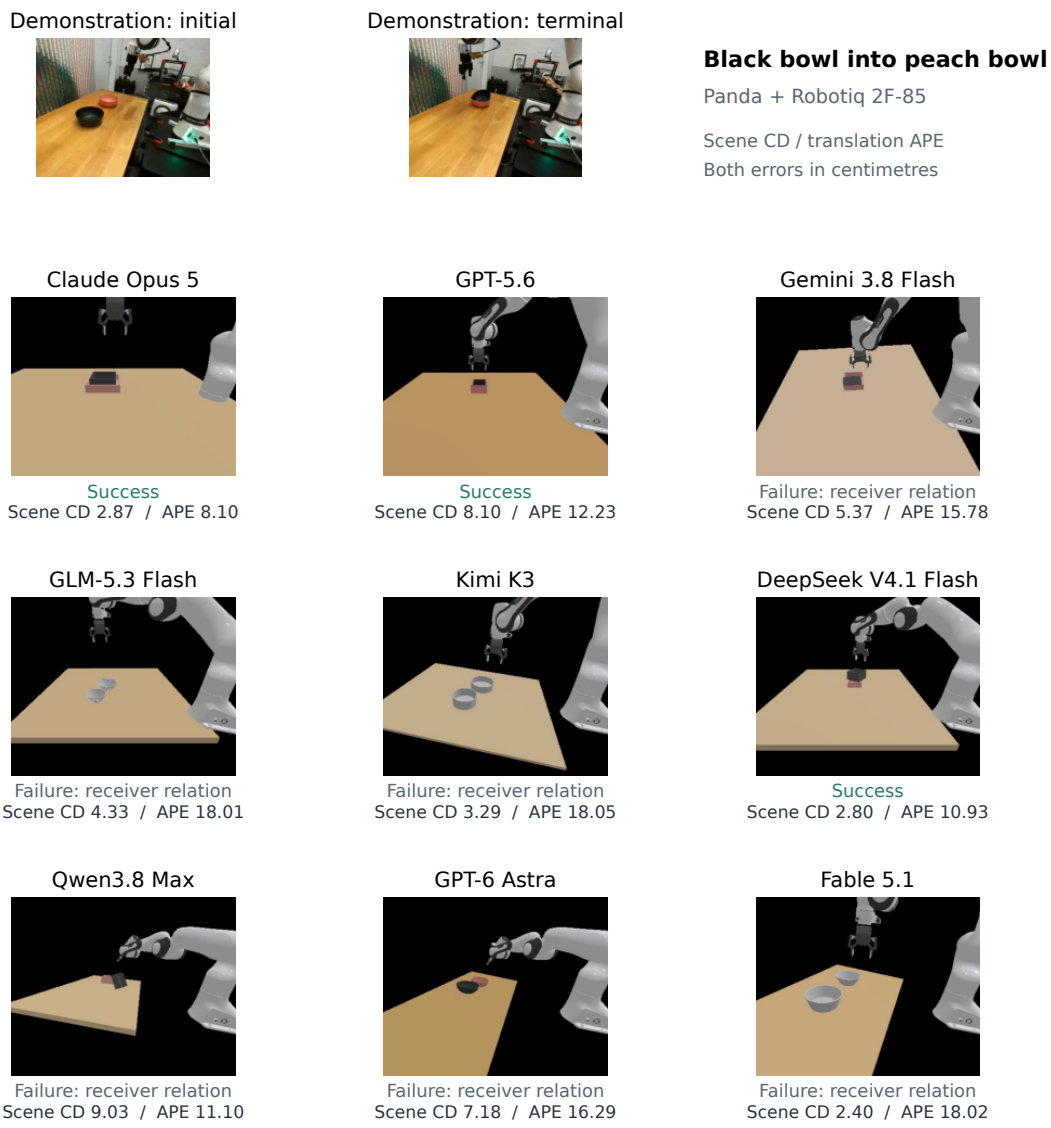

Figure A9: **Nine reconstructions of the same bowl-nesting demonstration.** Top: the source video's initial and terminal configurations. The nine model panels show the last frame of each submission's evaluation review video, using a common crop within this instance. Scene CD and translation APE are in centimetres. Opus, GPT-5.6, and DeepSeek succeed; the other six fail the receiving relation. Fable's smaller surface error than Opus does not imply successful placement. Scores and decisions come from the complete execution records, including terminal settling; the displayed video endpoints are qualitative views rather than substitutes for those records.



Figure A10: **Nine reconstructions of the same can-lifting demonstration.** Source frames and model panels follow Figure A9, with Scene CD and translation APE in centimetres. Opus and Kimi succeed. Qwen and Astra meet the ungated terminal lift condition but violate inverse-kinematics feasibility and robot-penetration checks, respectively. DeepSeek and Fable have valid executions without the required lift. The remaining three violate both checks. In particular, an elevated can in a rendered frame does not establish physical validity, which is determined from the execution record.

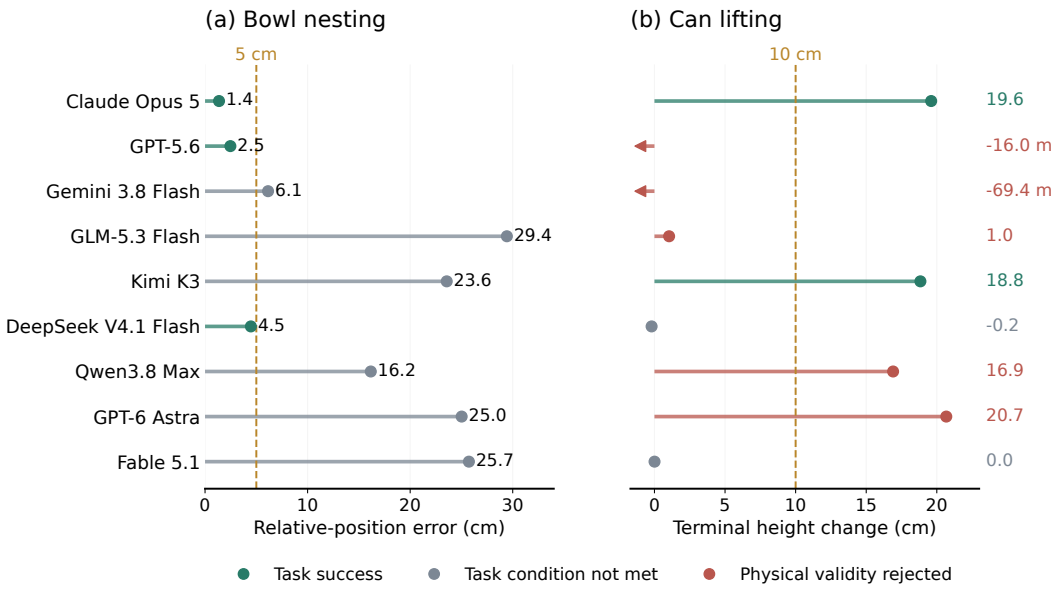

Figure A11: **Nine models, two tasks, and different reasons for failure.** (a) Terminal receiver-relative position error for black-bowl nesting. (b) Terminal can-height change for DexYCB lifting. Dashed lines mark the task-specific positional and height thresholds, which are components of the complete predicates. Green denotes Task Success, grey an unmet task condition, and red rejection by physical-validity checks. Qwen and Astra exceed the lift threshold but are rejected for inverse-kinematics infeasibility and robot penetration, respectively. Left-pointing markers denote out-of-range negative displacements, with their values stated explicitly. No aggregate model ranking is inferred from these two selected cases.

An accurate scene without the required object motion. Astra's X5 reconstruction has Scene CD of 1.66 cm but zero Task Progress and zero Task Success. The three blocks remain stationary to numerical precision throughout the recorded execution, and the native stacking predicate is false. Translation APE is 12.02 cm, averaged over the task objects, while terminal relative-position error is 33.99 cm. These observations locate the failure in the realized interaction without attributing an unmeasured contact or actuator mechanism to it.

Under the xArm7 embodiment, Astra succeeds on the same demonstration. Scene CD is 1.95 cm, translation APE is 1.19 cm, and terminal relative-position error is 0.16 cm. The different outcomes illustrate why the target robot and its control interface are part of the reconstruction instance: scene correspondence alone does not establish that the submitted behavior will manipulate the objects under that embodiment.

A correct-looking stack with unmet execution requirements. Fable succeeds with the X5 arms, with Scene CD of 1.42 cm, translation APE of 1.50 cm, and terminal relative-position error of 0.67 cm. Its xArm7 execution produces an apparently correct stack and has even smaller geometric and endpoint errors: Scene CD is 0.79 cm and terminal relative-position error is 0.45 cm, with translation APE of 1.55 cm. Nevertheless, Task Success and reported progress are zero.

The execution record identifies two deficiencies that a still image of the stack does not establish. The stacking subpredicate passes, but the required robot-return condition fails. In addition, coupled gripper joints violate their prescribed mechanical relation: the largest recorded follower-joint residual is approximately 0.685 rad, against a tolerance of 0.02 rad. The evaluator rejects this execution and suppresses its raw partial progress of 0.15. Thus, the favorable object arrangement is insufficient both as evidence of the complete task and as evidence of valid robot mechanics. The native checks certify their recorded constraints; they do not constitute a comprehensive collision certificate.

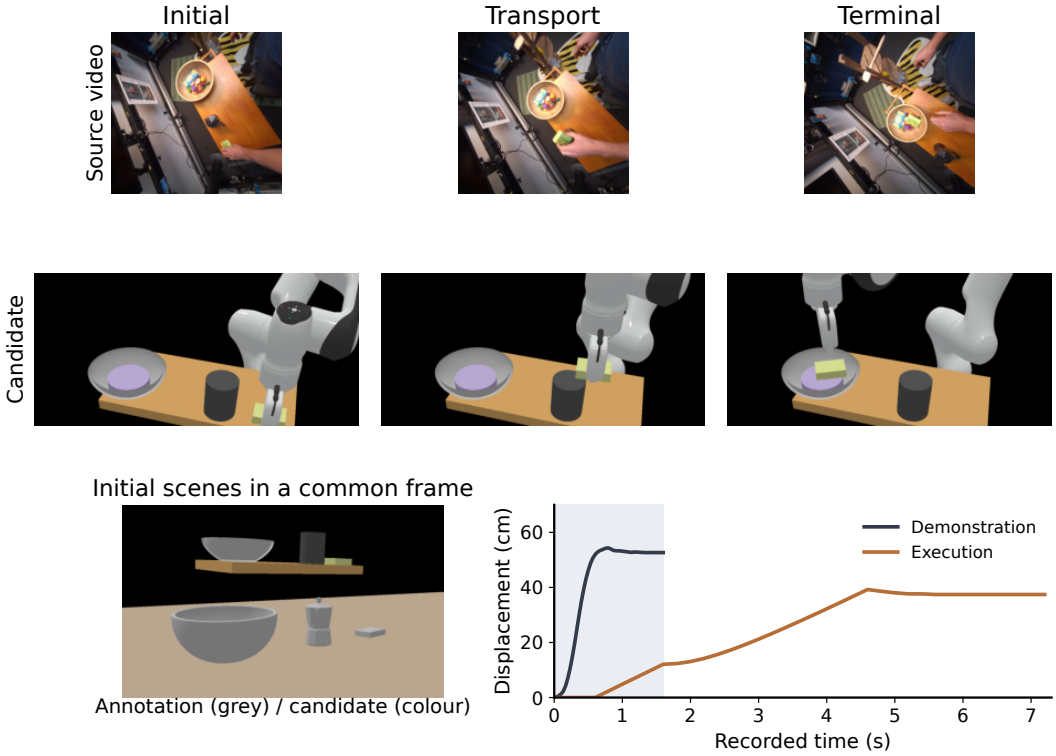

Figure A12: **Successful containment with spatial and temporal mismatch.** HOT3D food-package placement by a Franka arm. Top: source and candidate execution at selected task phases. Bottom left: the annotated scene in grey and the reconstruction in colour, viewed from the side in a common frame. Bottom right: displacement of each recorded object origin from its own initial position. The shaded interval is the 1.6 s demonstration window; the candidate execution continues to 7.2 s. This displacement plot illustrates timing and is not itself an APE curve.

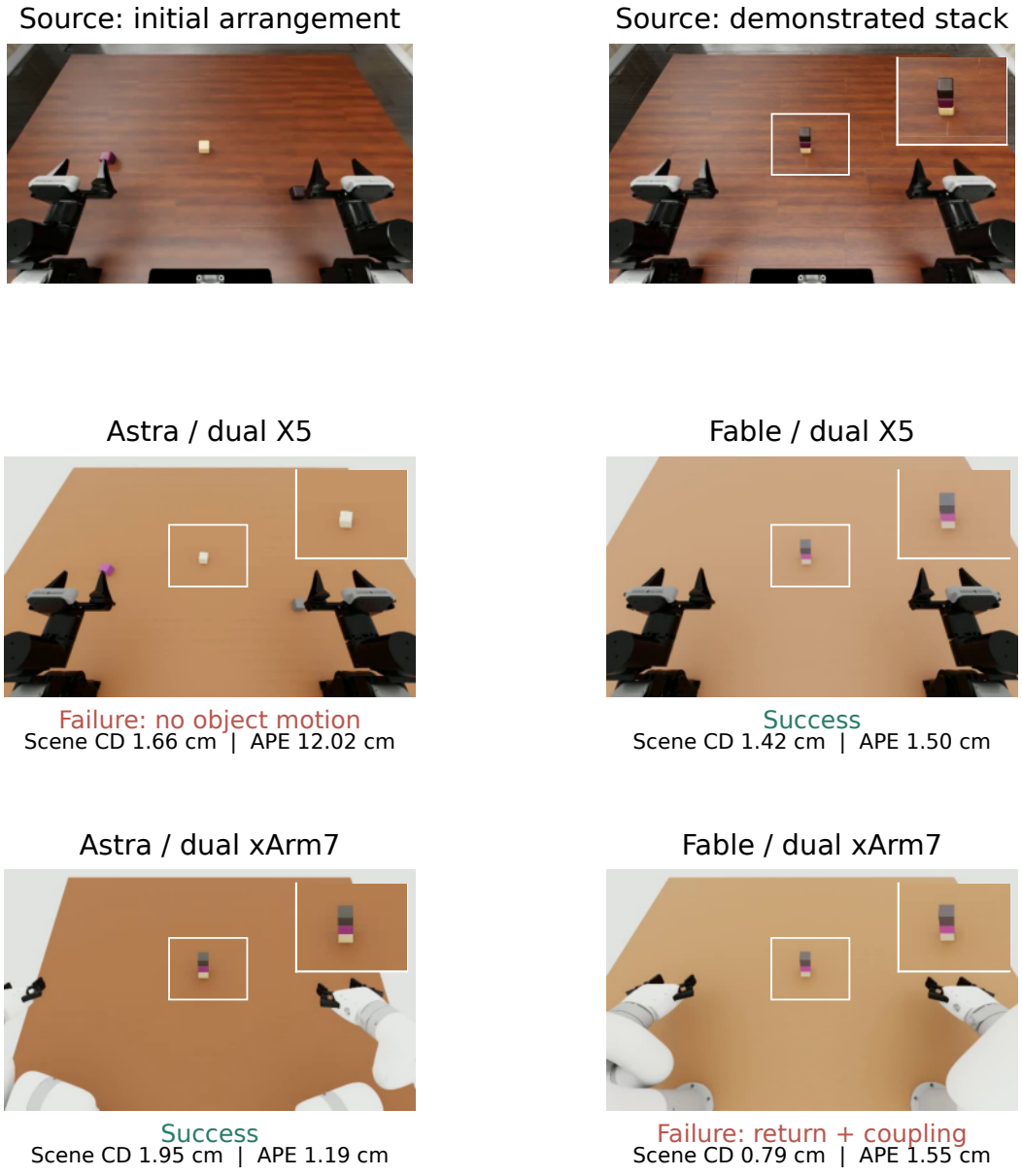

Figure A13: **One demonstration, two target robots, and distinct execution outcomes.** Top: the RoboDojo source video's initial and terminal arrangements. Middle and bottom: terminal states of Astra and Fable with X5 and xArm7 arms. Each panel is drawn from its own recorded physical execution; insets magnify the outlined receiver region. Astra's X5 objects remain stationary, while its xArm7 submission succeeds. Fable succeeds with X5; its xArm7 stack satisfies the object-stacking subpredicate but fails robot return and gripper-coupling checks. Scene CD and translation APE are shown in centimetres. A visually plausible final stack therefore does not certify the complete interaction.

G LIMITATIONS AND FUTURE WORK

Video2World evaluates autonomous video-to-simulation end to end, while several aspects of interactive world reconstruction remain open. First, functional correctness is measured from the joint execution of the reconstructed environment and the agent-generated robot behavior. Consequently, a failed rollout may originate from errors in the reconstructed world, the behavior policy, or their interaction. Future evaluation could disentangle these factors through standardized probe behaviors, multiple independently generated controllers, or controlled interventions on the reconstructed environment.

Second, the current benchmark primarily evaluates whether a reconstructed world can reproduce the demonstrated interaction. This leaves open a stronger notion of world reconstruction: whether the recovered environment remains physically and functionally valid under actions, configurations, and goals that are not observed in the source video. Evaluating such counterfactual behavior could distinguish task-specific reconstructions from reusable interactive worlds and provide a more direct measure of whether an agent has recovered the underlying structure of the physical environment.

Third, measured performance reflects the complete coding-agent system, including the underlying foundation model, agent scaffold, context management, tool use, and execution strategy. This system-level evaluation matches how coding agents are deployed in practice, but does not isolate which component is responsible for performance differences. Future work could independently vary models, scaffolds, tools, and feedback mechanisms to better characterize the sources of progress in autonomous world construction.

More broadly, Video2World focuses on reconstruction from a single embodied demonstration. Extending this setting to multi-view observations, multiple demonstrations, active information gathering, and progressively accumulated experience could move autonomous video-to-simulation from reconstructing individual interactions toward learning persistent, reusable world models.