HRC-SIMBENCH: BENCHMARKING INTERACTION-CENTRIC HUMAN-ROBOT COLLABORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Current vision-language-action (VLA) benchmarks primarily evaluate isolated manipulation skills while leaving human-robot interaction structure largely unmodeled. However, real-world collaboration fundamentally requires coordination under shared agency, including intent understanding, temporal synchronization, protocol adherence, and safe interaction in dynamic environments. To address this gap, we introduce HRC-SimBench, a diagnostic benchmark for intent-aware human-robot collaboration based on executable interaction scenarios. HRC-SimBench represents collaborative tasks as structured scenario scripts that explicitly model agent roles, temporal dependencies, coordination constraints, and human behavior distributions. Building on this abstraction, HRC-SimBench defines three representative interaction roles: Instructor, Collaborator, and Interferer, covering intent communication, joint coordination, and robustness under human intervention. The benchmark contains 13 role-conditioned tasks with over 650 evaluation episodes generated from diverse interaction trajectories and scene variations. Beyond binary task success, HRC-SimBench introduces interpretable interaction-centric metrics spanning synchronization, responsiveness, protocol compliance, and safety. We evaluate adapted policies based on GR00T, π0.5, and ACT under a unified protocol. Results show that current foundation robot policies struggle substantially in collaborative settings despite strong manipulation ability, revealing major limitations in temporal coordination and intent-aware behavior. Fine-tuning on HRC-SimBench consistently improves collaborative performance. In a real-world adaptation study, simulation data generated by HRC-SimBench improves GR00T N1.5's physical-task success rate from 0.10 to 0.43, demonstrating the benchmark's value for advancing interaction-centric robot learning.

1 INTRODUCTION

Vision-language-action (VLA) models have recently achieved rapid progress in language-conditioned robotic manipulation (Zitkovich et al., 2023; Kim et al., 2024; Black et al., 2024). Large-scale robot policies can now generalize across diverse objects, instructions, and environments, leading to substantial advances in embodied foundation models. Existing benchmarks accordingly focus on manipulation-centric tasks such as pick-and-place, articulated object operation, and goal-conditioned execution. However, most current evaluations implicitly assume that robots operate alone. Real-world robot deployment, by contrast, fundamentally involves interaction with humans. In collaborative settings, robots must not only manipulate objects, but also coordinate actions with people in shared environments. This requires reasoning about human intentions, temporal coordination, interaction protocols, and physical safety under shared agency (Goodrich & Schultz, 2008; Sheridan, 2016; Dragan et al., 2013; Huang & Mutlu, 2016; Strabala et al., 2013; Lasota et al., 2017; Villani et al., 2018; Argall et al., 2009). A policy that succeeds in isolated manipulation may still fail to respond to a collaborator's reaching motion, violate coordination timing, or continue execution after disruptive human intervention. These failures expose a critical gap between manipulation competence and collaborative intelligence.

We argue that this gap arises because existing VLA benchmarks primarily evaluate *manipulation outcomes*, while leaving the underlying *interaction structure* between humans and robots largely unmodeled. Human-robot collaboration is fundamentally a coordination problem rather than a standalone manipulation problem. Effective collaboration depends not only on *what* action is executed,

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

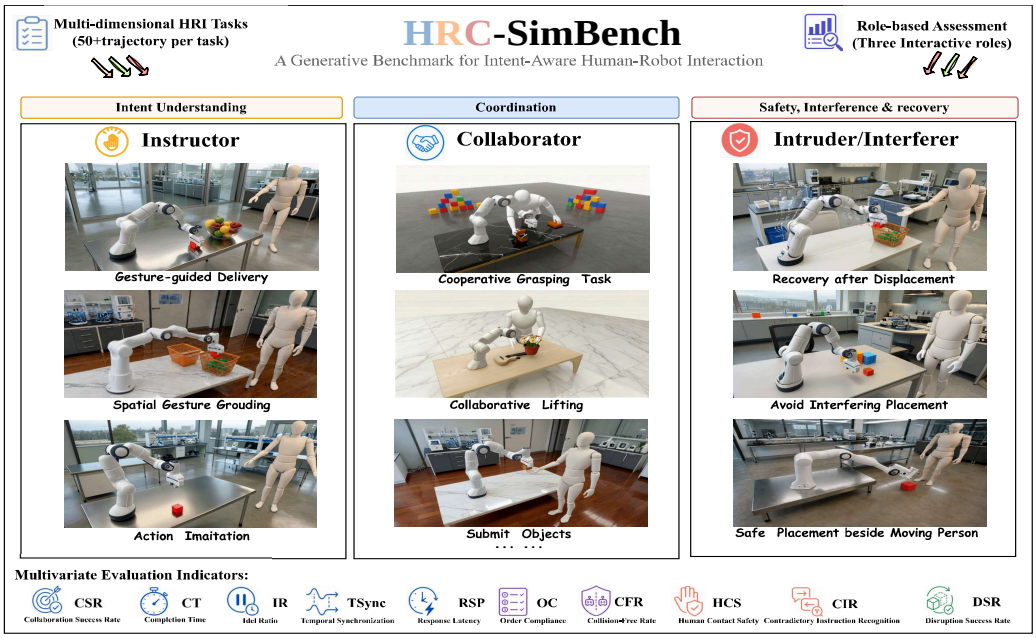


Figure 1: HRC-SimBench evaluates robot policies in executable human-robot interaction scenarios. Each scenario explicitly specifies the human role, the robot objective, temporal coordination constraints, and the human motion distribution.

but also on *when*, *why*, and *in response to whom* the action is performed. This requires explicit modeling of interaction dependencies, including role assignments, temporal ordering, synchronization constraints, and human behavioral dynamics.

To address this limitation, we introduce **HRC-SimBench**, a diagnostic benchmark for intent-aware human-robot collaboration. Rather than treating collaboration as isolated task execution, HRC-SimBench represents interaction as executable coordination structure. Specifically, we introduce *scenario scripts*, a structured and executable representation of collaborative interaction episodes. Each script explicitly specifies participating agents, interaction stages, temporal dependencies, coordination constraints, protocol ordering, and human motion descriptions. This abstraction provides an interpretable semantic layer connecting high-level interaction requirements to executable simulation environments, reward functions, and evaluation metrics.

A key design principle of HRC-SimBench is to organize collaborative behavior around *interaction roles*. HRC-SimBench does not aim to exhaustively cover the full diversity of real human behavior; instead, it provides a controlled, simulation-based approximation of representative interaction patterns. Accordingly, the roles below are not exhaustive human categories, but complementary interaction contracts that characterize how human behavior constrains the correct robot action. We identify three representative role patterns of human behavior frequently encountered in human-robot interaction: *Instructor*, *Collaborator*, and *Interferer*. Instructor scenarios evaluate whether robots can infer intended actions from human demonstrations or gestures. Collaborator scenarios require synchronized multi-agent coordination such as object handover or joint manipulation. Interferer scenarios test robustness under disruptive human intervention, requiring policies to maintain safety and adapt execution online. Importantly, these categories are not merely task groupings; they represent distinct coordination structures with different intent, timing, and safety requirements. Based on this framework, HRC-SimBench instantiates 13 role-conditioned tasks with over 650 executable evaluation episodes generated from diverse interaction trajectories and scene variations. Rather than maximizing benchmark scale, we prioritize interaction completeness, interpretability, and diagnostic value. This design enables failures to be attributed to specific coordination deficiencies rather than hidden dataset heterogeneity.

Beyond binary task completion, HRC-SimBench introduces an interaction-centric evaluation protocol spanning synchronization, responsiveness, protocol compliance, and safety. This allows col-

laborative failures to be decomposed into interpretable categories: a policy may fail because it misunderstands human intent, acts at an inappropriate time, violates coordination order, or enters unsafe proximity. Such disentangled evaluation is critical for understanding the limitations of current embodied foundation models in collaborative settings. We evaluate representative robot policies adapted from modern VLA and imitation-learning backbones, including GR00T N1.5 (Bjorck et al., 2025), $\pi_{0.5}$ (Physical Intelligence et al., 2025), and ACT (Zhao et al., 2023). Experimental results show that although these models exhibit strong manipulation capabilities, they struggle substantially in collaborative interaction scenarios, particularly under temporal coordination and human perturbation: the best CSR (Collaboration Success Rate) remains 0.533 on Instructor tasks and 0.500 on Collaborator tasks, but drops to only 0.100 on Interferer tasks. At the same time, HRC-SimBench data improves real-world adaptation of GR00T N1.5, increasing the average physical-task success rate from 0.10 with Real-only post-training to 0.43 with Sim+Real post-training. These results demonstrate the benchmark's effectiveness for diagnosing interaction-centric failures and providing structured data for improving human–robot collaboration policies.

In summary, the main contributions of this paper are as follows: (1) We introduce **HRC-SimBench**, a diagnostic benchmark for human-robot collaboration that reformulates interaction evaluation as executable coordination modeling. HRC-SimBench contains 13 role-conditioned tasks spanning instructor-guided execution, collaborative coordination, and robustness under human intervention. (2) We propose **scenario scripts**, a structured executable representation of collaborative interaction that explicitly models agent roles, temporal dependencies, coordination constraints, and human behavior distributions. (3) We establish an **interaction-centric evaluation protocol** for embodied policies, including synchronization, responsiveness, protocol compliance, and safety, enabling interpretable analysis of collaborative failure modes across modern robot foundation models.

2 Related Work

Robot manipulation and VLA benchmarks. Robot learning benchmarks have made object manipulation measurable and comparable (Li et al., 2023; Nasiriany et al., 2024). RLBench (James et al., 2020), CALVIN (Mees et al., 2022), ManiSkill (Mu et al., 2021), and Meta-World (Yu et al., 2020) cover multi-task manipulation, language-conditioned execution, articulated-object interaction, and long-horizon control. In parallel, VLA policies such as RT-2 (Zitkovich et al., 2023), OpenVLA (Kim et al., 2024), π_0 (Black et al., 2024), $\pi_{0.5}$ (Physical Intelligence et al., 2025), and GR00T (Bjorck et al., 2025) have improved generalization across objects, scenes, and instructions. These systems are an important foundation for HRC-SimBench: without manipulation competence, collaboration cannot succeed. The limitation is that most of these evaluations still end at an object-centric goal. A robot is usually judged by whether it opens a drawer, places an object, or follows a command, while the human remains outside the execution loop (Shi et al., 2025). In real collaboration, however, the right action depends on a person's cue, timing, movement, or interference (Shi et al., 2025; Jung et al., 2026). HRC-SimBench therefore keeps the policy-level manipulation setting, but makes the human role part of the task definition rather than an external annotation.

Human-aware embodied evaluation. HRI research has long emphasized that collaboration depends on intent communication, legible behavior, temporal coordination, handover timing, and safe shared-space operation (Argall et al., 2009; Sauppé & Mutlu, 2014; Dragan et al., 2013; Huang & Mutlu, 2016; Strabala et al., 2013; Lasota et al., 2017; Jung et al., 2026; Bo et al., 2026; Wang et al., 2026; Bejarano Sepulveda et al., 2026; Hoffman & Breazeal, 2007; Cakmak et al., 2011). Recent embodied benchmarks move in this direction by introducing humans, humanoid agents, or multi-agent task structure (Wang et al., 2024). Habitat 3.0 (Puig et al., 2024) and PARTNR (Chang et al., 2025) study co-present agents and collaborative planning in household environments, while HandoverSim (Chao et al., 2022) focuses on close-proximity handover. HRC-SimBench is complementary to these efforts. It does not aim to replace navigation-scale embodied planning benchmarks or specialized handover simulators. Instead, it asks a narrower but policy-critical question: when a manipulation policy is placed in an executable interaction contract, can it infer the human role, act at the right time, obey the protocol, and recover safely when the human changes the scene (Wang et al., 2026; Jung et al., 2026; Baraglia et al., 2016)?

Generative simulation. Generative simulation methods use language models or procedural pipelines to create robot tasks, scenes, rewards, and demonstrations (Wang et al., 2023; Ma et al., 2024; Zhang & Soh, 2023; Nasiriany et al., 2024). Human motion generation further provides diverse motion priors (Guo et al., 2022; Petrovich et al., 2022; Tevet et al., 2022; Guo et al., 2024; Batool et al., 2026; Zhang et al., 2024; 2023). These tools are useful for scaling environment construction, but a benchmark needs more than plausible generated content: episodes must be executable, reproducible, and tied to unambiguous metrics. HRC-SimBench uses typed scripts, labeled human motions, success predicates, and simulation filtering to turn generated scenarios into standardized interaction episodes.

3 BENCHMARK DESIGN

HRC-SimBench is a diagnostic benchmark for evaluating intent-aware human–robot collaboration. It is organized around three interaction roles and contains 13 tasks with more than 50 trajectory variants per task, yielding over 650 executable evaluation episodes. Unlike conventional manipulation benchmarks that treat humans as passive scene context, HRC-SimBench explicitly models the human as part of the task specification. Each role determines what information the human provides, how the human moves during execution, and which collaborative failure modes are exposed.

Table 1 summarizes the diagnostic coverage of HRC-SimBench relative to representative benchmarks. The goal is not to provide a general ranking, since existing embodied benchmarks target complementary capabilities such as large-scale manipulation or embodied planning. Instead, HRC-SimBench focuses on interaction-centric evaluation by jointly supporting role-conditioned scripts, policy-level manipulation under dynamic human motion, perturbation and recovery protocols, interaction-specific metrics, and physical adaptation studies within a unified framework. Additional task details are provided in Appendix B (Tables 9 and 10).

Table 1: Qualitative coverage of representative benchmarks. ✓ indicates explicit support, △ indicates partial or indirect coverage, and × indicates that the axis is not a primary benchmark target. "Role Script" denotes role-conditioned executable interaction scripts; "Human Manipulation" indicates policy-level manipulation under dynamic human motion; "Perturbation" refers to perturbation and recovery protocols; "Interaction Metrics" denotes interaction-specific evaluation metrics; and "Physical Adaptation" indicates whether physical adaptation studies are provided.

Benchmark	Role Script	Human Manipulation	Perturbation	Interaction Metrics	Physical Adaptation
RLBench (James et al., 2020) / CALVIN (Mees et al., 2022)	×	×	×	×	×
ManiSkill (Mu et al., 2021) / Meta-World (Yu et al., 2020)	×	×	×	×	×
Habitat 3.0 (Puig et al., 2024) / PARTNR (Chang et al., 2025)	×	△	△	△	×
HandoverSim (Chao et al., 2022)	△	✓	×	✓	×
RoboGen (Wang et al., 2023) / Eureka-style generation	×	×	×	×	×
HRC-SimBench (ours)	✓	✓	✓	✓	✓

Scope and Task Relevance HRC-SimBench prioritizes diagnostic density rather than benchmark scale. Each task is designed to expose a specific collaborative failure mode that is difficult to evaluate in isolated manipulation settings, including deictic grounding, handover timing, shared-object synchronization, yielding behavior, and recovery from human intervention. As a result, the same manipulation skill may appear under different interaction regimes: a policy capable of grasping an object may still fail when it must wait for a gesture cue, synchronize with a moving collaborator, or stop execution after disruptive human interference.

The role taxonomy is grounded in established HRI requirements. *Instructor* tasks evaluate intent communication through demonstrations or gestures (Argall et al., 2009; Villani et al., 2018). *Collaborator* tasks evaluate close-proximity coordination, where robots must anticipate human motion

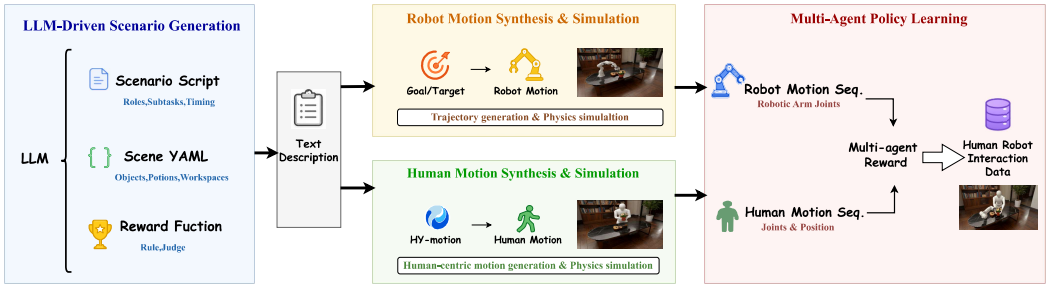


Figure 2: HRC-SimBench generation pipeline. An LLM creates scenario scripts, environment specifications, and reward components from task and object metadata. Human and robot trajectories are synthesized, simulated, filtered, and used to construct policy adaptation and evaluation data.

and execute temporally synchronized actions (Huang & Mutlu, 2016; Mainprice & Berenson, 2013; Chao et al., 2022). *Interferer* tasks evaluate safety and recovery under disruptive human intervention in shared environments (Dragan et al., 2013; Villani et al., 2018). This role-based organization keeps the benchmark compact while ensuring that each category corresponds to a distinct coordination challenge.

Scenario Scripts HRC-SimBench represents collaborative interaction through executable *scenario scripts*. A script is defined as a sequence of interaction acts:

$$\mathcal{S} = \{a_1, \ldots, a_T\}, \quad a_t = (r_t, g_t, c_t, m_t, \rho_t), \tag{1}$$

where r_t denotes the active interaction role, g_t the semantic goal, c_t the temporal or causal constraints, m_t the human motion distribution, and ρ_t the reward and success conditions.

Scenario scripts separate *what* collaboration requires from *how* a policy executes it. This abstraction improves interpretability and controllability: scripts can be inspected by humans, compiled into simulation environments, and instantiated with different objects, trajectories, timing offsets, viewpoints, and distractors while preserving the same interaction structure.

Interaction Paradigms *Instructor* scenarios require robots to infer intended actions from human demonstrations or gestures before execution. *Collaborator* scenarios require synchronized multi-agent coordination, such as object handover or joint manipulation, where timing and responsiveness are critical. *Interferer* scenarios introduce disruptive human interventions and evaluate whether policies maintain safety, reject invalid interference, and recover task execution online. Additional details of the Interferer protocol are provided in Appendix F (Table 17).

Metrics HRC-SimBench evaluates collaboration using three groups of metrics. **Effectiveness** measures task completion quality, including Collaboration Success Rate (CSR), Completion Time (CT), and Idle Ratio (IR). **Coordination** measures interaction timing and protocol adherence, including Temporal Synchronization (TSync), Response Latency (Rsp), and Order Compliance (OC). **Safety and robustness** measures safe and reliable interaction, including Collision-Free Rate (CFR), Human Contact Safety (HCS), Contradictory Instruction Recognition (CIR), and Disruption Success Rate (DSR). These metrics are reported alongside task success because collaborative failures are often multi-dimensional. A policy may complete the manipulation objective while violating temporal coordination, responding to the wrong cue, breaking protocol order, or behaving unsafely around humans. Full metric definitions are provided in Appendix A.

4 Generation Pipeline

HRC-SimBench converts high-level collaboration specifications into executable interaction trajectories through four stages: scenario scripting, scene instantiation, motion synthesis, and simulation validation. The pipeline is designed to preserve the underlying interaction contract while varying scene layout, human behavior, timing, and low-level motion, enabling evaluation of collaborative reasoning rather than memorization of fixed trajectories.

Motion synthesis, rewards, and interaction variants. Human interaction prompts are converted into candidate skeleton trajectories using HY-Motion-1.0 (Wen et al., 2025), followed by task-aware refinement with labeled constraints on hands, torso orientation, and approach directions. These structured labels make collaborative behavior directly measurable. For example, the benchmark can verify whether a pointing gesture enters the target cue region, whether a receiving hand remains stable during handover, or whether an interferer violates a restricted workspace boundary. HRC-SimBench automatically constructs reward functions and success predicates from task completion, coordination quality, and safety requirements. Each task contains more than 50 trajectory variants that preserve the same semantic interaction objective while varying human timing, motion patterns, object poses, and perturbation schedules. This diversity prevents overfitting to fixed trajectories and exposes policies to a broad range of coordination conditions. Additional generation details are provided in Appendix D (Tables 12 and 13).

Simulation validation and filtering. All generated scenarios are executed in simulation before inclusion in the benchmark. We filter out episodes with unreachable states, severe human-object penetration, invalid camera visibility, inconsistent script ordering, missing success predicates, or labeled joints that fail to satisfy target-region constraints. After automatic validation, retained candidates are manually inspected for executability and semantic correctness. Quantitative filtering statistics and annotation agreement are reported in Section 5.2. The resulting validated trajectory pool is then split into policy adaptation and held-out evaluation sets. For Interferer scenarios, perturbation schedules are generated independently of the evaluated policy and kept fixed across models, ensuring consistent intervention conditions and fair comparison of recovery and safety behavior.

5 Experiments

The experiments are designed to test the central hypothesis of HRC-SimBench: strong manipulation capability does not necessarily imply robust collaborative intelligence. We evaluate representative VLA and imitation-learning policies across the three interaction roles, and further conduct a small-scale real-world adaptation study to examine whether HRC-SimBench-generated interaction data improves physical deployment under limited demonstrations.

5.1 Implementation Setup

All simulation experiments use a Franka Panda manipulator and are conducted on four NVIDIA A800 GPUs. For each task, 50 validated trajectories are used for policy adaptation and 10 held-out trajectories are reserved for evaluation. The evaluation trajectories vary human timing, motion patterns, and scene initialization, preventing policies from simply replaying memorized behaviors. For the physical adaptation study, we construct task-matched simulation data for a LeRobot SO-100 arm and collect 20 corresponding real-world teleoperation trajectories for each of three representative tasks, one per interaction role.

5.2 Layered Validation of Human Motions and Interactions

We validate the generated data along complementary dimensions: executable and semantic correctness, kinematic fidelity to recorded human motion, and perceived correspondence to real human–robot interaction. These analyses examine whether trajectories realize the intended collaboration, whether their motion resembles human behavior, and whether participants recognize meaningful interaction requirements. Together, they provide complementary evidence of the data's validity and correspondence to human interaction, with quantitative characterization of residual motion differences.

Executable and semantic validity. Applying the validation pipeline described in Section 4, automatic checks reject 1,066 of 2,000 candidates; manual semantic auditing removes another 148 mechanically valid trajectories that fail to express the intended interaction, leaving 786. Independent double annotation of 100 resampled trajectories yields 94% agreement ($\kappa = 0.87$). This layered screening checks both physical executability and interaction meaning, rather than treating successful simulation alone as sufficient validation (Appendix D).

Kinematic correspondence with recorded human motion. We compare pointing, reaching, and handover using 40 generated and 32 automatically retrieved AMASS (Mahmood et al., 2019)/GRAB (Taheri et al., 2020) clips per primitive. Mean peak-velocity and duration deviations are relatively small, at 6.3% and 9.4%. Time-normalized wrist-speed profiles show close correspondence for pointing and reaching, with modest distributional differences across primitives ($D = 0.09$–0.16). Interaction timing is particularly demanding during handover, where smoothness is slightly reduced, as reflected in 14.8% higher mean jerk and a detectable speed-distribution difference ($p = 0.012$; Table 2; Figure 3). Despite this modest reduction in smoothness, the validated trajectories preserve the overall interaction pattern and successful task completion. Details appear in Appendix E.1.

Table 2: Generated–human motion differences.

Primitive	KS D	p
Pointing	0.09	0.39
Reaching	0.11	0.18
Handover	0.16	0.012

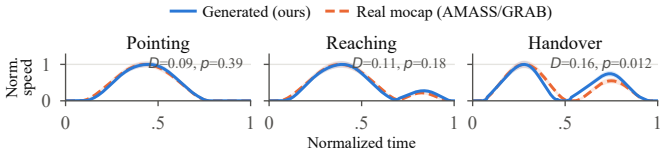


Figure 3: Generated and recorded wrist-speed profiles (mean $\pm$ s.d.).

User study: interaction correspondence and design feedback. The 36-participant study complements motion-level measurements with detailed five-point Likert-scale analysis of how well the scenarios represent real human–robot interaction. Participants inspect role definitions, task cards, and trajectory visualizations. Practical feasibility, typicality, fluency, accuracy, and credibility receive 4.67, 4.38, 4.54, 4.38, and 4.25, respectively, supporting perceived correspondence across both task relevance and interaction quality (Hoffman & Breazeal, 2007; Cho et al., 2014). Role-design rationality, metric-design rationality, and task approval receive 3.82 ± 0.79, 4.50 ± 0.61, and 4.30 ± 0.70 (one-sample Wilcoxon $p < 0.001$ versus the midpoint). All three ratings are significantly above the scale midpoint, indicating positive participant feedback on the role design, evaluation metrics, and task selection. Item-level means, standard deviations, medians, agreement rates, and role-level approval are reported in Appendix E.2. Feedback informs clearer timing labels and task presentation; participant responses are not used for policy training.

Taken together, execution checks and semantic audits establish data correctness, motion comparisons quantify agreement and residual kinematic differences, and questionnaire analysis supports the scenarios' perceived relevance to real collaboration. The consistently positive participant ratings complement the motion-level measurements, providing converging evidence that the generated data captures meaningful characteristics of real human–robot interaction within the tested tasks. The quantitative comparisons further characterize the residual differences, supporting a controlled approximation of human interaction for benchmark evaluation.

5.3 Main Benchmark Results

We evaluate GR00T N1.5 with LoRA adaptation, $\pi_{0.5}$ with LoRA adaptation, and ACT with full fine-tuning under matched data splits, observation preprocessing, and evaluation seeds whenever supported by the architectures. These backbones provide complementary evaluation settings: two pretrained VLA policies with language-conditioned action interfaces, and one imitation-learning baseline based on chunked continuous control. Additional backbone details are provided in Appendix G.2 (Tables 19 and 20). Table 3 reveals a clear role-dependent difficulty progression. Instructor scenarios are comparatively easier because the human mainly provides semantic cues, whereas Collaborator scenarios require continuous temporal coordination with a moving partner. Interferer scenarios are substantially harder because policies must detect that nominal execution is no longer valid and adapt their behavior online under perturbation.

The VLA-based policies generally achieve higher collaboration success and synchronization scores than ACT, suggesting that pretrained vision-language representations improve cue grounding and interaction-context understanding. However, all methods struggle severely in Interferer scenarios. Despite perfect or near-perfect Order Compliance (OC), the Collaboration Success Rate (CSR) remains extremely low, indicating that current policies can follow nominal task orderings but fail to

properly yield, maintain safe clearance, or recover after disruptive human intervention. These results expose a critical gap between manipulation competence and interaction-aware collaborative intelligence. Additional analyses are provided in Appendix H.3 (Table 25).

Table 3: Main HRC-SimBench evaluation. Safety is a compact normalized summary of applicable safety and recovery components, including CFR, HCS, CIR, and DSR. Component-wise results are provided in Appendix H.2 (Table 24).

Role group	Model	CSR↑	CT↓	Rsp↓	TSync↑	OC↑	Safety↑
Instructor	GR00T N1.5-LoRA	**0.533**	10.5	**0.001**	**0.463**	0.667	**0.993**
	$\pi_{0.5}$-LoRA	**0.533**	**10.3**	0.028	0.421	0.667	0.990
	ACT-FT	0.433	14.8	0.165	0.341	0.667	0.984
Collaborator	GR00T N1.5-LoRA	**0.500**	**18.7**	0.207	**0.394**	1.000	**1.000**
	$\pi_{0.5}$-LoRA	0.457	20.1	0.146	0.355	1.000	0.994
	ACT-FT	0.371	22.4	**0.078**	0.286	1.000	0.987
Interferer	GR00T N1.5-LoRA	**0.100**	**25.8**	**0.000**	**0.276**	1.000	**0.735**
	$\pi_{0.5}$-LoRA	**0.100**	26.3	0.006	0.254	1.000	0.729
	ACT-FT	**0.100**	30.0	0.016	0.224	1.000	0.721

5.4 Real-World Adaptation

We further conduct a small-scale real-world adaptation study using GR00T N1.5 on a LeRobot SO-100 arm. The objective is not to establish a full physical benchmark, but to evaluate whether HRC-SimBench-generated interaction data provides a useful initialization when real demonstrations are scarce. **Real-only** performs post-training using 20 real teleoperation trajectories per task. **Sim+Real** first adapts the policy using task-matched HRC-SimBench simulation trajectories generated for SO-100, followed by post-training on the same real-world demonstrations. We report task success rate over 10 evaluation trials per task.

Table 4: Real-world adaptation comparison with GR00T N1.5 on a LeRobot SO-100 arm.

Role group	Real-world task	Real-only SR↑	Sim+Real SR↑
Instructor	Deliver via gesture commands	2/10	**5/10**
Collaborator	Collaborative object retrieval from a box	1/10	**4/10**
Interferer	Avoid having someone else do the placement	0/10	**4/10**
Average	–	0.10	**0.43**

Across 30 physical evaluation trials, Real-only succeeds in only 3 cases, while Sim+Real succeeds in 13 cases. The improvement is consistent across all three interaction roles, including gesture-conditioned execution, collaborative retrieval in shared workspaces, and interference-aware placement. Although the physical study remains limited in scale, the results suggest that structured interaction data from HRC-SimBench provides transferable priors for real-world collaborative behavior.

5.5 Failure Mode Analysis

Beyond aggregate success rates, HRC-SimBench exposes distinct role-dependent collaborative failure modes. Instructor failures are primarily caused by incorrect cue grounding or premature execution before the human signal is completed. Collaborator failures often occur despite correct manipulation intent, because the robot reacts too early, too late, or fails to synchronize with the human partner during shared-object interaction. Interferer failures are qualitatively different: policies frequently continue the original execution plan after disruptive human intervention, leading to unsafe proximity, invalid recovery behavior, or failure to restore the intended task state. These findings highlight why binary task success alone is insufficient for evaluating collaborative embodied intelligence. A policy may complete the object-level goal while still violating temporal coordination, interaction protocol, or safety requirements. The full failure-mode breakdown is provided in Appendix H.3 (Table 25).

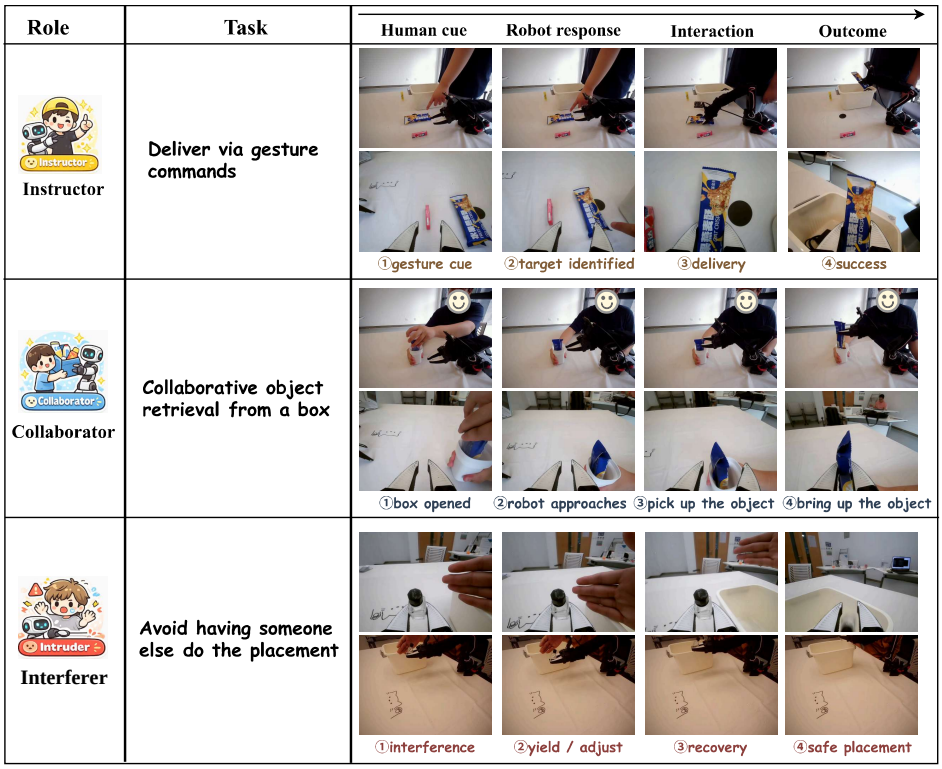


Figure 4: Representative real-world SO-100 tasks: gesture-conditioned delivery, collaborative retrieval, and interference-aware placement.

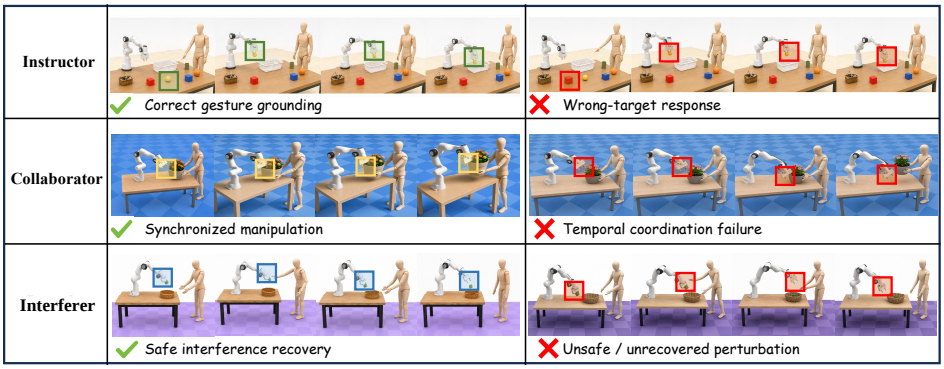


Figure 5: Representative successful and failed HRC-SimBench rollouts across different interaction roles.

6 Conclusion

HRC-SimBench reframes human–robot collaboration as executable, role-conditioned interaction rather than isolated manipulation. By combining scenario scripts, generated human motion, simulation validation, and interaction-centric metrics, it diagnoses collaborative failures in terms of intent grounding, temporal coordination, safety, and recovery. Experiments with GR00T N1.5, $\pi_{0.5}$, and ACT show that current robot policies remain substantially limited in collaborative settings, especially under human intervention. A small-scale SO-100 study further suggests that HRC-SimBench-generated simulation data can improve real-world adaptation when demonstrations are scarce. Overall, HRC-SimBench provides a diagnostic benchmark for evaluating and improving interaction-aware robot learning.

AI USE STATEMENT

Generative AI tools were used during benchmark construction to generate synthetic datasets, as detailed in the paper. They were additionally used to assist with language editing and LaTeX formatting.

ETHICS STATEMENT

This work studies human-robot collaboration through executable interaction scenarios and a real-world adaptation study. HRC-SimBench is a simulation benchmark and does not involve real human participants; human behavior is modeled through structured scenario scripts and simulated interaction distributions. The real-world adaptation study involves human participants. These participants provided informed consent under an IRB-approved protocol, and all human-subject data were de-identified. No personally identifiable information is released. The benchmark and trained policies are intended for research on human-robot collaboration and should not be deployed in shared human spaces without safety constraints, human oversight, and fail-safe mechanisms. We encourage users to audit the benchmark and policies for bias, unsafe interaction, and unintended social impact.

REPRODUCIBILITY STATEMENT

We provide comprehensive details of HRC-SimBench, including its scenario scripts, agent roles, temporal dependencies, coordination constraints, human behavior distributions, 13 role-conditioned tasks, over 650 evaluation episodes, and interaction-centric metrics, in the main paper and appendix. We will release the simulation benchmark, generation and evaluation code, prompts, configuration files, and policy adaptation scripts.

REFERENCES

Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

Jimmy Baraglia, Maya Cakmak, Yukie Nagai, Rajesh Rao, and Minoru Asada. Initiative in robot assistance during collaborative task execution. In *2016 11th ACM/IEEE international conference on human-robot interaction (HRI)*, pp. 67–74. IEEE, 2016.

Faryal Batool, Iana Zhura, Valerii Serpiva, Roohan Ahmed Khan, Ivan Valuev, Issatay Tokmurziyev, and Dzmitry Tsetserukou. HumanDiffusion: A vision-based diffusion trajectory planner with human-conditioned goals for search and rescue UAV. In *Companion Proceedings of the 21st ACM/IEEE International Conference on Human-Robot Interaction*, pp. 1023–1027, 2026.

Edison Jair Bejarano Sepulveda, Valerio Bo, Alberto Sanfeliu, and Anais Garrell. Perception–awareness–decision: Socially-aware robot navigation and interaction. In *Companion Proceedings of the 21st ACM/IEEE International Conference on Human-Robot Interaction*, pp. 258–262, 2026.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Valerio Bo, Anais Garrell, and Alberto Sanfeliu. Fast or accurate? how intention-recognition models shape human perception of a mobile robot. In *Companion Proceedings of the 21st ACM/IEEE International Conference on Human-Robot Interaction*, pp. 502–506, 2026.

Maya Cakmak, Siddhartha S Srinivasa, Min Kyung Lee, Sara Kiesler, and Jodi Forlizzi. Using spatial and temporal contrast for fluent robot-human hand-overs. In *Proceedings of the 6th international conference on Human-robot interaction*, pp. 489–496, 2011.

Matthew Chang, Gunjan Chhablani, Alexander Clegg, Mikael Dallaire Cote, Ruta Desai, Michal Hlavac, Vladimir Karashchuk, Jacob Krantz, Roozbeh Mottaghi, Priyam Parashar, et al.

PARTNR: A benchmark for planning and reasoning in embodied multi-agent tasks. In *International Conference on Learning Representations*, volume 2025, pp. 65205–65268, 2025.

Yu-Wei Chao, Chris Paxton, Yu Xiang, Wei Yang, Balakumar Sundaralingam, Tao Chen, Adithyavairavan Murali, Maya Cakmak, and Dieter Fox. HandoverSim: A simulation framework and benchmark for human-to-robot object handovers. In *2022 International conference on robotics and automation (ICRA)*, pp. 6941–6947. IEEE, 2022.

Hyunyi Cho, Lijiang Shen, and Kari Wilson. Perceived realism: Dimensions and roles in narrative persuasion. *Communication Research*, 41(6):828–851, 2014. doi: 10.1177/0093650212450585. URL https://doi.org/10.1177/0093650212450585.

Anca D Dragan, Kenton CT Lee, and Siddhartha S Srinivasa. Legibility and predictability of robot motion. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 301–308. IEEE, 2013.

Michael A Goodrich and Alan C Schultz. Human–robot interaction: a survey. *Foundations and trends® in human–computer interaction*, 1(3):203–275, 2008.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3D human motions from text. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5152–5161, 2022.

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. MoMask: Generative masked modeling of 3D human motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1900–1910, 2024.

Guy Hoffman and Cynthia Breazeal. Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pp. 1–8, 2007.

Chien-Ming Huang and Bilge Mutlu. Anticipatory robot control for efficient human-robot collaboration. In *2016 11th ACM/IEEE international conference on human-robot interaction (HRI)*, pp. 83–90. IEEE, 2016.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. RLBench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

Magnus Jung, Qiaoyue Yang, Leander von Seelstrang, Dominykas Strazdas, Sven Wachsmuth, and Ayoub Al-Hamadi. SEMIAC: A two-site human-robot collaboration dataset for exploring socially-enriched models for implicit action coordination. In *Companion Proceedings of the 21st ACM/IEEE International Conference on Human-Robot Interaction*, pp. 1028–1033, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Przemyslaw A Lasota, Terrence Fong, and Julie A Shah. A survey of methods for safe human-robot interaction. *Foundations and Trends® in Robotics*, 5(4):261–349, 2017.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. BEHAVIOR-1K: A benchmark for embodied AI with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pp. 80–93. PMLR, 2023.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Jim Fan, et al. Eureka: Human-level reward design via coding large language models. In *International conference on learning Representations*, volume 2024, pp. 26516–26560, 2024.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Jim Mainprice and Dmitry Berenson. Human-robot collaborative manipulation planning using early prediction of human motion. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 299–306. IEEE, 2013.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.

Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. ManiSkill: Generalizable manipulation skill benchmark with large-scale demonstrations. *arXiv preprint arXiv:2107.14483*, 2021.

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. *arXiv preprint arXiv:2406.02523*, 2024.

Mathis Petrovich, Michael J Black, and Gül Varol. TEMOS: Generating diverse human motions from textual descriptions. In *European conference on computer vision*, pp. 480–497. Springer, 2022.

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

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars, and robots. In *International Conference on Learning Representations*, volume 2024, pp. 15306–15336, 2024.

Abhinanda R. Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J. Black. BABEL: Bodies, action and behavior with english labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 722–731, June 2021.

Allison Sauppé and Bilge Mutlu. Robot deictics: How gesture and context shape referential communication. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pp. 342–349, 2014.

Thomas B Sheridan. Human–robot interaction: status and challenges. *Human factors*, 58(4):525–532, 2016.

Zhonghao Shi, Enyu Zhao, Nathaniel Dennler, Jingzhen Wang, Xinyang Xu, Kaleen Shrestha, Mengxue Fu, Daniel Seita, and Maja Matarić. HRIBench: Benchmarking vision-language models for real-time human perception in human-robot interaction. *arXiv preprint arXiv:2506.20566*, 2025.

Kyle Wayne Strabala, Min Kyung Lee, Anca Diana Dragan, Jodi Lee Forlizzi, Siddhartha Srinivasa, Maya Cakmak, and Vincenzo Micelli. Towards seamless human-robot handovers. *Journal of Human-Robot Interaction*, 2(1):112–132, 2013.

Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *European Conference on Computer Vision (ECCV)*, 2020. URL `https://grab.is.tue.mpg.de`.

Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022.

Valeria Villani, Fabio Pini, Francesco Leali, and Cristian Secchi. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*, 55:248–266, 2018.

Chenxu Wang, Boyuan Du, Jiaxin Xu, Peiyan Li, Di Guo, and Huaping Liu. Demonstrating HumanTHOR: a simulation platform and benchmark for human-robot collaboration in a shared workspace. *arXiv preprint arXiv:2406.06498*, 2024.

Hong Wang, Katie Winkle, and Ginevra Castellano. Anticipating safety and security risks of embodied AI: A multidimensional risk framework for foundation models in HRI. In *Companion Proceedings of the 21st ACM/IEEE International Conference on Human-Robot Interaction*, pp. 1039–1044, 2026.

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. *arXiv preprint arXiv:2311.01455*, 2023.

Yuxin Wen, Qing Shuai, Di Kang, Jing Li, Cheng Wen, Yue Qian, Ningxin Jiao, Changhai Chen, Weijie Chen, Yiran Wang, et al. HY-Motion 1.0: Scaling flow matching models for text-to-motion generation. *arXiv preprint arXiv:2512.23464*, 2025.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

Bowen Zhang and Harold Soh. Large language models as zero-shot human models for human-robot interaction. In *2023 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 7961–7968. IEEE, 2023.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen, and Ying Shan. Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14730–14740, 2023.

Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. MotionDiffuse: Text-driven human motion generation with diffusion model. *IEEE transactions on pattern analysis and machine intelligence*, 46(6):4115–4128, 2024.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.

A METRIC DEFINITIONS

This section gives the episode-level definitions used by HRC-SimBench. Metrics are computed per rollout and then averaged over task variants and role groups unless otherwise specified. The current evaluation uses a single adapted checkpoint per backbone and a fixed held-out split; therefore, the reported values should be read as fixed-split averages rather than multi-seed confidence estimates. Let $p_r(t)$ and $p_h(t)$ be robot and human reference positions, $v_r(t)$ and $v_h(t)$ their velocities, K the number of human cue or perturbation events, and $\mathcal{E}$ the set of scripted events in an episode.

Evaluation constants and thresholds. Table 5 lists the default thresholds used by the evaluator in the current benchmark release. These values are scenario-level defaults: individual scripts may override them when an object, workspace, or human-motion primitive requires a narrower or wider region, but all overrides are stored in the executable scenario script.

Table 5: Default metric thresholds and event windows. Values are stored in the evaluator configuration and can be overridden by task scripts when needed.

Quantity	Default value	Use in evaluation
Episode timeout	35.0 s	Maximum rollout duration before failure termination.
Response timeout	6.0 s	Assigned to missing cue, handover, or perturbation responses.
Synchronization lag bound $\tau_{\max}$	2.0 s	Maximum lag considered by TSync cross-correlation.
Coordination window	1.5 s	Time window around script-marked joint-action events.
Safe human clearance d_{safe}	0.10 m	Minimum clearance between robot end-effector or held object and labeled human body region.
Near-contact warning distance	0.05 m	Used for safety-stop logging and manual audit review.
Handover zone radius	0.12 m	Distance from receiver hand center for valid handover approach and release.
Gesture target radius	0.18 m	Region around the indicated object or delivery target for valid cue grounding.
Human hand stability window	0.50 s	Minimum stable receiving or pointing pose duration.
Interferer restricted-zone radius	0.18 m	Region around active robot manipulation target used to trigger takeover or yield checks.
Robot idle threshold δ_r	0.01 m/s	End-effector speed below which the robot is treated as idle.
Human active threshold δ_h	0.05 m/s	Labeled human joint speed above which the human is treated as active.
Safe contact-force proxy F_{safe}	10 N	Force threshold used when simulator or real logs provide a force estimate.

Collaboration Success Rate For N episodes and successful set S,

$$\text{CSR} = \frac{|S|}{N}. \tag{2}$$

An episode is counted as successful only if the task goal is achieved, the script-level ordering constraints are satisfied, and required safety constraints are not violated. This definition prevents an unsafe or protocol-violating completion from being counted as a valid collaborative success.

Completion Time

$$\text{CT}_i = t_i^{\text{end}} - t_i^{\text{start}}. \tag{3}$$

CT is reported over successful episodes unless otherwise specified. Failed episodes are excluded from CT to avoid mixing speed with non-completion; their effect is captured by CSR and safety metrics.

Idle Ratio For robot and human action intensities $u_r(t)$ and $u_h(t)$,

$$\mathrm{IR} = \frac{1}{T} \sum_t \mathbf{1}\left[u_r(t) < \delta_r \wedge u_h(t) > \delta_h \wedge q(t) = 1\right], \tag{4}$$

where $q(t)$ indicates that the script expects robot responsiveness. Correct waiting is not penalized: $q(t) = 0$ during scripted hold, observation, or yielding intervals.

Temporal Synchronization

$$\mathrm{TSync} = \max_{\tau \in [-\tau_{\max}, \tau_{\max}]} \mathrm{corr}\left(v_r(t), v_h(t-\tau)\right). \tag{5}$$

TSync measures whether robot motion aligns with the relevant human motion window after allowing a bounded lag. For tasks without continuous human motion, synchronization is evaluated only over script-marked coordination windows. If a rollout terminates before a required coordination window opens, the missing window receives TSync $= 0$; if a task has no synchronization requirement, TSync is marked not applicable and excluded from role averages.

Response Latency For cue time t_k^I and first valid robot response t_k^A,

$$\mathrm{Rsp} = \frac{1}{K} \sum_{k=1}^{K} (t_k^A - t_k^I). \tag{6}$$

A valid response must satisfy the event predicate associated with the cue, such as moving toward the indicated target, entering a handover approach phase, pausing after takeover, or replanning after displacement. Missing responses are assigned the response timeout in Table 5 before averaging. Rsp is computed over all episodes with a valid cue event, including failed episodes, because a policy can fail precisely by never responding to the human event.

Order Compliance Given required event order $E^\star$ and executed events E,

$$\mathrm{OC} = \frac{1}{|E^\star|} \sum_j \mathbf{1}\left[\mathrm{order}_E(e_j^\star) \text{ satisfies script constraints}\right]. \tag{7}$$

The script constraints include before/after relations, mutually exclusive branches, and required waits for human readiness predicates.

Collision-Free Rate

$$\mathrm{CFR} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{1}\left[\min_t d(p_r(t), o_k(t)) \geq d_{\mathrm{safe}}\right], \tag{8}$$

where $o_k(t)$ is the human body part or object region associated with the k-th safety event.

Human Contact Safety When contact-force estimates are available,

$$\mathrm{HCS} = \mathrm{clip}\left(\min\left(\frac{d_{\min}}{d_{\mathrm{safe}}}, \frac{F_{\mathrm{safe}}}{\max(F_{\max}, \epsilon)}\right), 0, 1\right). \tag{9}$$

When force estimates are not reliable, HRC-SimBench reports the clearance component and records the missing force channel as not applicable rather than silently treating it as safe.

Contradictory Instruction Recognition

$$\mathrm{CIR} = \frac{N_{\text{correct conflict}}}{N_{\text{conflict}}}. \tag{10}$$

CIR is active in variants where the human instruction conflicts with the current object state, the previous cue, or the script-level safety condition. A correct response rejects, pauses, or requests a safe recovery action instead of blindly executing the latest cue.

Disruption Success Rate

$$\mathrm{DSR} = \frac{N_{\text{success after disruption}}}{N_{\text{disrupted}}}. \tag{11}$$

DSR is active for Interferer variants and measures whether the policy maintains safety and restores the task-relevant state after a scripted perturbation.

Safety summary The compact Safety column reported in main-paper Table 2 averages the normalized safety and robustness components that are applicable to a given episode:

$$\mathrm{Safety}_i = \frac{1}{|\mathcal{A}_i|} \sum_{m \in \mathcal{A}_i} \mathrm{clip}(m_i, 0, 1), \tag{12}$$

where $\mathcal{A}_i \subseteq \{\mathrm{CFR}, \mathrm{HCS}, \mathrm{CIR}, \mathrm{DSR}\}$ is determined by the task script. Components that are not defined for a role or variant are excluded from the denominator.

Table 6: Applicability of safety and robustness components. Script-dependent components are evaluated only for variants whose scenario script contains the corresponding event.

Role group	CFR	HCS	CIR	DSR	Primary use
Instructor	Applicable	Applicable	Script-dependent	Not applicable	Safe cue grounding
Collaborator	Applicable	Applicable	Not applicable	Not applicable	Safe close-proximity coordination
Interferer	Applicable	Applicable	Script-dependent	Applicable	Interference recovery

Table 7: Metric applicability and failed-rollout handling. This table complements the formal definitions by specifying which denominator each metric uses.

Metric	Role coverage	Failed-episode handling
CSR	All role groups.	Failed episodes contribute 0.
CT	All role groups.	Reported over successful episodes only.
IR	All role groups.	Computed over valid pre-termination frames.
Rsp	Cue variants, coordination variants, and perturbation variants.	Missing responses receive the response timeout.
TSync	Gesture-motion, collaborator-motion, and perturbation-motion variants.	Missing required windows receive 0; not-applicable windows are excluded.
OC	All role groups.	Violated or skipped required events contribute 0 for that event.
CFR	All role groups.	Unsafe termination still contributes to CFR/HCS.
HCS	Variants with clearance or force-proxy measurements.	Missing force logs use clearance only and are marked in metadata.
CIR	Contradictory-cue and unsafe-interference variants.	Variants without conflicts are excluded.
DSR	Interferer variants.	Failed recovery contributes 0.

B TASK SUITE

The current benchmark contains 13 tasks, corresponding to the role-prefixed folders in the benchmark release. Each task has more than 50 trajectory variants generated by varying human motion, timing, object pose, articulation state, and scene context. Table 9 summarizes the full task suite, Table 8 clarifies the interaction contract behind each role, and Table 10 reports the per-task generation and evaluation split statistics used by the experiments.

Table 8: Role contracts used to turn broad HRI requirements into executable HRC-SimBench scenarios.

Role	Human process	Robot obligation	Main diagnostic failures
Instructor	Provides gesture, demonstration, or deictic cue before robot execution.	Infer the intended target/action and act only after the cue is observable.	Wrong cue grounding, premature execution, contradictory-instruction failure.
Collaborator	Moves in the shared workspace as an active partner.	Coordinate timing, handover, shared object motion, or complementary manipulation.	Temporal desynchronization, late response, unsafe proximity, handover instability.
Interferer	Perturbs the nominal task or enters a restricted workspace region.	Yield, maintain clearance, reject unsafe interference, and recover the task state.	Continuing nominal execution, collision risk, failure to replan after displacement.

Table 9: HRC-SimBench task suite. The release contains 13 role-conditioned tasks: 3 Instructor, 7 Collaborator, and 3 Interferer tasks.

Role	Task	Primary capability tested
Instructor	Deliver via gesture commands	Map human gestures to delivery actions and timing.
Instructor	Perform the corresponding action based on the gesture	Select and execute the correct action from gesture cues.
Instructor	Select the correct delivery location based on the gesture	Ground deictic gestures to spatial delivery targets.
Collaborator	Building with blocks together	Coordinate shared construction and ordered block placement.
Collaborator	Collaborative carrying of flower pots	Maintain synchronized motion while carrying a shared object.
Collaborator	Hand the object over to a moving person	Time handover to a moving receiver.
Collaborator	Remove the block from the hand	Safely retrieve an object from a human hand.
Collaborator	Retrieve the object after handing it to someone	Recover or re-acquire an object after a prior transfer.
Collaborator	Sorting and picking tasks	Divide object sorting and picking actions with a human partner.
Collaborator	Collaborative object retrieval from a box	Coordinate human access and robot retrieval around a container.
Interferer	Avoid having someone else do the placement	Detect and respond when a person interferes with placement.
Interferer	Change the position to complete the operation	Recover from a human-induced object displacement.
Interferer	Place object beside	Maintain goal and safety when an interferer changes nearby object layout.



Figure 6: Full HRC-SimBench task gallery. All 13 tasks grouped by role, each panel labeled by task ID.

Table 10: Per-task dataset statistics for the current HRC-SimBench release. The six validated trajectories beyond the fixed 50/10 split are kept as reserve trajectories for manual audit and replacement when a held-out episode is found invalid.

ID	Role	Task	#Generated	#Validated	#Train/adapt	#Eval	Objects	Human variants	Perturbation type
I1	Instructor	Deliver via gesture commands	151	61	50	10	cup, tray, target pads	3 gestures, 4 timings	None
I2	Instructor	Perform action based on gesture	149	60	50	10	block, button, tool	4 gestures, 3 viewpoints	None
I3	Instructor	Select delivery location by gesture	153	61	50	10	parcel, bins, table zones	5 pointing targets	None
C1	Collaborator	Building with blocks together	154	60	50	10	colored blocks, base plate	3 partners, 4 pace profiles	None
C2	Collaborator	Collaborative carrying of flower pots	153	61	50	10	flower pot, shared tray	4 walking speeds	None
C3	Collaborator	Hand object to moving person	156	60	50	10	handheld object, receiver hand	4 approach paths	None
C4	Collaborator	Remove block from hand	152	61	50	10	block, human hand	3 grasp poses, 4 holds	None
C5	Collaborator	Retrieve object after handover	154	60	50	10	cube, shelf zone	3 return timings	None
C6	Collaborator	Sorting and picking tasks	155	61	50	10	objects, bins, distractors	4 division patterns	None
C7	Collaborator	Object retrieval from box	156	60	50	10	box, lid, target object	4 reach-in timings	None
N1	Interferer	Avoid someone else doing placement	156	61	50	10	target block, placement pad	3 entry paths	Takeover
N2	Interferer	Change position to complete operation	155	60	50	10	target object, distractor	4 displacement offsets	Object-displace
N3	Interferer	Place object beside	156	60	50	10	two objects, table zones	3 layout changes	Layout-change
–	Total	–	2000	786	650	130	–	–	–

Variant axes Within each task, variants preserve the same semantic objective while changing timing offsets, human trajectory clips, object poses, articulation states, distractor layout, camera view, and perturbation schedule when applicable. This design makes the benchmark medium-scale but diagnostic: a failed rollout can usually be traced to a role-specific interaction requirement rather than to an unrelated task change. The full gallery in Figure 6 uses the same task IDs as Table 10, so the visual examples can be read directly against the dataset split.

C Scenario Script Format

As described in the main paper's generation-pipeline section, the prompt requires explicit substep boundaries, active agents, timing strings, coordination constraints, and human motion descriptions. This mirrors the HRIGen construction in which `parsed_collaboration_task.json` contains `task_name`, `description`, `collaboration_type`, and `substeps` fields, with each substep dispatched according to its `agent` field. HRC-SimBench extends this structure with fields needed for executable evaluation, including success predicates, termination rules, evaluator state variables, and variant axes.

Table 11: Required prompt-schema fields used before simulation instantiation.

Field	Required content
`task_name`	Unique task identifier aligned with the benchmark folder name.
`role`	One of `Instructor`, `Collaborator`, or `Interferer`.
`scene`	Object categories, articulation states, initial object poses, robot start region, human start region, workspace bounds, and camera views.
`substeps`	Ordered list of human, robot, or joint actions with timing windows and active agent labels.
`coordination`	Readiness conditions, wait conditions, handover regions, synchronization windows, or perturbation triggers.
`success`	Measurable terminal predicates over object state, robot state, human labels, and safety state.
`termination`	Timeout, success, unrecoverable failure, and safety-stop conditions.
`variant_axes`	Parameters allowed to vary across trajectory variants while preserving the same collaborative objective.

Listing 1: Instructor scenario script example: gesture-guided delivery.

```
{
  "task_name": "deliver_via_gesture_commands",
  "role": "Instructor",
  "scene": {
    "robot": "Franka Panda",
    "objects": ["cup", "tray", "left_pad", "right_pad"],
    "human_start_region": "front-left of table",
    "workspace_bounds": "shared tabletop",
```

```
    "camera_views": ["front", "wrist"]
  },
  "substeps": [
    {
      "agent": "human",
      "timing": "t=0.0-2.5s",
      "action": "point to the requested delivery pad",
      "coordination": "robot must wait until the pointing hand is stable for 0.5s",
      "human_motion": "raise right arm, point toward target pad, hold pose"
    },
    {
      "agent": "robot",
      "timing": "t=2.5-10.0s",
      "action": "pick the cup and deliver it to the indicated pad",
      "coordination": "move only after the gesture predicate is true",
      "success": "cup center is within gesture_target_radius of indicated pad"
    }
  ],
  "termination": ["success", "timeout", "wrong_target", "unsafe_contact"],
  "variant_axes": ["gesture_target", "gesture_duration", "cup_pose", "distractor_layout", "
     camera_view"]
}
```

Listing 2: Collaborator scenario script example: handover to a moving person.

```
{
  "task_name": "hand_object_to_moving_person",
  "role": "Collaborator",
  "scene": {
    "robot": "Franka Panda",
    "objects": ["small handheld object"],
    "human_start_region": "left side of exchange corridor",
    "workspace_bounds": "tabletop shared workspace",
    "camera_views": ["front", "side", "wrist"]
  },
  "substeps": [
    {
      "agent": "human",
      "timing": "t=0-3s",
      "action": "walk toward the exchange region and extend the receiving hand",
      "coordination": "robot waits until the hand is visible and inside the handover zone",
      "human_motion": "walk forward, slow down, extend right arm with open hand"
    },
    {
      "agent": "both",
      "timing": "t=3-7s",
      "action": "robot transfers the object while the human maintains a stable receiving pose
     ",
      "coordination": "end-effector approaches the hand, releases after proximity and
     stability predicates hold",
      "success": "object is supported by the human-side target region without unsafe contact"
    }
  ],
  "termination": ["success", "timeout", "unsafe_contact"],
  "variant_axes": ["human_speed", "handover_time", "object_pose", "camera_view"]
}
```

Listing 3: Interferer scenario script example: avoid someone else doing the placement.

```
{
  "task_name": "avoid_someone_else_doing_the_placement",
  "role": "Interferer",
  "scene": {
    "robot": "Franka Panda",
    "objects": ["target_block", "placement_pad", "distractor_block"],
    "human_start_region": "right side of table",
    "workspace_bounds": "shared tabletop with restricted manipulation zone",
    "camera_views": ["front", "overhead"]
  },
  "substeps": [
    {
      "agent": "robot",
      "timing": "t=0.0-4.0s",
      "action": "grasp target block and move toward placement pad",
      "coordination": "monitor restricted zone while executing nominal placement"
    },
    {
      "agent": "human",
      "timing": "trigger=robot_near_pad",
```

```
      "action": "reach toward target block and attempt takeover",
      "coordination": "interferer hand enters restricted-zone radius"
    },
    {
      "agent": "robot",
      "timing": "within 6.0s after perturbation",
      "action": "yield or pause, keep clearance, then recover the placement when safe",
      "coordination": "resume only after human hand exits the restricted zone",
      "success": "block is placed on pad and no unsafe clearance violation is logged"
    }
  ],
  "termination": ["success", "timeout", "unsafe_contact", "unrecovered_disruption"],
  "variant_axes": ["interferer_entry_time", "interferer_path", "block_pose", "pad_pose", "
     camera_view"]
}
```


D GENERATION DETAILS

Reward and success specification For each act, HRC-SimBench constructs a reward specification with three components:

$$R_t = R_{\text{task}} + \lambda_c R_{\text{coord}} + \lambda_s R_{\text{safety}}. \tag{13}$$

R_{task} measures progress toward object and end-effector goals, R_{coord} rewards appropriate temporal or spatial coordination with the human, and R_{safety} penalizes unsafe proximity, collision, or high-contact-force states. Generated rewards are compiled, unit-tested on synthetic states, and checked against script-level success predicates.

Scene instantiation The scene generator converts the typed script into simulator state by sampling object poses, articulation states, robot base and end-effector initialization, human start regions, workspace bounds, and camera parameters. Sampling is constrained by reachability, visibility, and collision checks so that the resulting variants remain executable rather than merely visually diverse.

Human behavior model Human behavior is built in two stages. First, the script-level motion text is sent to HY-Motion-1.0 to generate multiple candidate 3D human motion clips for the same substep. Second, each candidate is converted into a labeled skeleton trajectory with task-relevant body and hand joints, such as the pointing hand, receiving hand, support hand, torso, and approach direction. These labels are used by both the evaluator and the human-motion refinement objective.

The final human trajectory is refined to approach the task-relevant region smoothly while satisfying role-specific constraints. For Instructor tasks, the labeled hand must make the gesture target observable before the robot is allowed to act. For Collaborator tasks, the receiving or supporting hand approaches the object or end-effector with bounded velocity and low jerk. For Interferer tasks, the skeleton follows a fixed perturbation schedule so that every robot policy faces the same human intervention. A hand-object interaction is valid when the labeled hand joints enter a task-specific proximity region, remain stable for a minimum duration, and avoid unsafe end-effector clearance violations.

Domain augmentation and trajectory variants HRC-SimBench supports visual domain augmentation through renderer randomization and video-based appearance transfer. Since video generation can alter geometry or contact relations, augmented observations are accepted only when consistency checks preserve task-relevant object state, human pose, and robot-object spatial relationships. Each task contains more than 50 trajectory variants that preserve the same semantic objective while changing low-level execution, such as pointing time, hand approach speed, or interferer entry time.

Simulation validation and filtering Generated scenarios are executed in simulation before entering the benchmark. Table 12 summarizes the validation filters. From 2,000 generated interaction trajectories, 786 pass automatic validation and manual executable-and-correct screening. The retained pool supplies the role-conditioned policy adaptation and held-out evaluation splits used in the main experiments. Figure 7 gives the same validation flow in retained-count form, with the first-failing rejection categories broken out in Table 13.

Table 12: Validation checks used to filter generated HRC-SimBench trajectories before policy adaptation and evaluation.

Check	Rejects	Purpose
Schema validation	Missing agents, timing windows, predicates, or termination rules.	Ensures every scenario is executable and measurable.
Reachability	Robot targets outside reachable workspace or invalid articulation states.	Prevents impossible manipulation episodes.
Human-object consistency	Severe human-object penetration or missing hand/object contact labels.	Preserves physically meaningful human motion.
Camera coverage	Key cue, handover, or perturbation regions not visible.	Avoids failures caused only by invalid observation setup.
Predicate execution	Missing state variables or predicates that never trigger on test states.	Ensures rewards and metrics can be computed.
Ordering and timing	Substeps that violate required before/after constraints.	Maintains the collaborative contract specified by the script.
Manual audit	Semantically incorrect but mechanically valid trajectories.	Removes cases where automatic checks miss task meaning.

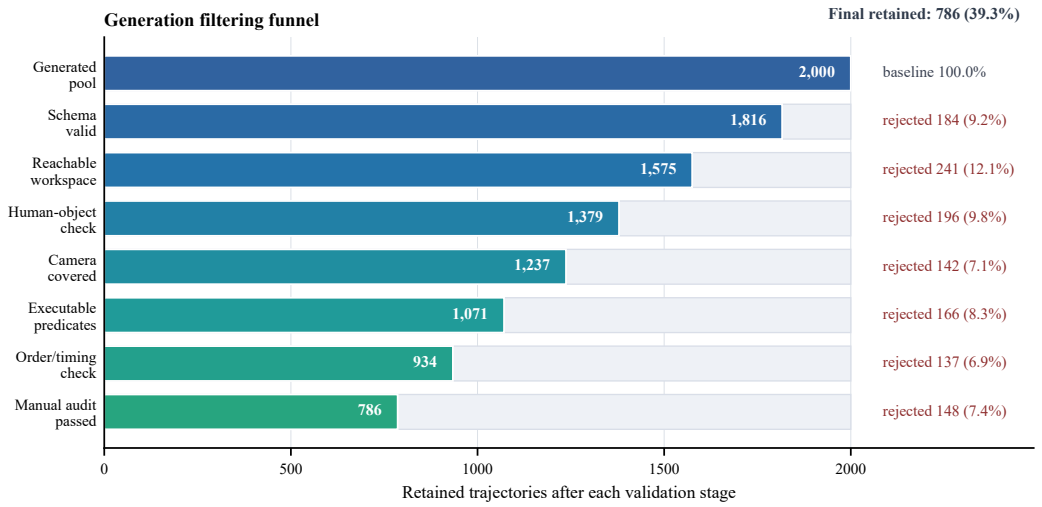


Figure 7: Generation filtering funnel. Bars show the retained trajectory count after each validation stage, and annotations report first-failing rejection counts and percentages from Table 13.

Table 13: Filtering rejection breakdown for 2,000 generated trajectories. Categories are counted by first failing validation stage, so percentages sum to 100%.

First failing validation stage	Count	Percentage	Typical cause
Schema invalid	184	9.2%	Missing agent labels, timing windows, or termination predicates.
Unreachable robot target	241	12.1%	Target pose or articulation state outside the robot workspace.
Human-object penetration or invalid contact	196	9.8%	Hand/object labels inconsistent with the generated skeleton motion.
Camera coverage failure	142	7.1%	Gesture, handover, or perturbation event outside required camera view.
Predicate never triggered	166	8.3%	Success, cue, or perturbation predicate failed on synthetic state tests.
Ordering or timing violation	137	6.9%	Script substeps violated required before/after constraints.
Manual semantic rejection	148	7.4%	Mechanically valid rollout did not match intended interaction semantics.
Retained	786	39.3%	Passed automatic validation and manual executable-and-correct audit.

E Human-Motion Validation and User Feedback

E.1 Motion-Capture Comparison Protocol

Reference clips are selected automatically from AMASS using BABEL (Punnakkal et al., 2021) frame-level "point" and "reach" labels, and from GRAB "pass"-intent sequences for handover. The selection retains single-arm segments of 1–4 s with a tabletop-height workspace and no marker dropout, giving 32 reference clips per primitive. Each primitive is compared with 40 generated clips using time-normalized SMPL-X wrist-speed profiles. Figure 3 shows the mean and standard deviation; Table 2 summarizes the quantitative results. The two-sample KS analysis uses 200 pooled normalized speed samples per group. These within-trajectory samples are not independent experimental units, so the statistics describe distributional discrepancies and are not a clip-level equivalence test. The mean peak-velocity and duration deviations are 6.3% and 9.4%; handover has 14.8% higher mean jerk. This analysis concerns selected upper-limb motion primitives, not the realism of unrestricted human behavior or reactive human–robot coordination.

E.2 Task-Relevance User Study

We recruited 36 participants from a campus population. Participants viewed role definitions, task cards, and representative trajectory visualizations and assessed task relevance, role consistency, and the clarity of required robot behavior. Five-point Likert ratings measure familiarity, understanding, or agreement, with higher scores indicating stronger endorsement. The study functions as a design filter and feedback source; responses are not used to train or adapt policies. Tasks are retained when judged practically meaningful, role-consistent, and mappable to observable robot failures. The quantitative task-card retention rule is reported in Table 16.

The reported one-sample Wilcoxon tests against the scale midpoint yield $p < 0.001$ for role-design rationality, metric-design rationality, and overall task approval. These participant judgments concern benchmark relevance and interpretability, rather than objective motion fidelity. The additional five-dimension assessment covers practical feasibility, typicality, fluency, accuracy, and credibility; all five reported scores are at least 4.25 out of 5 (Figure 8).

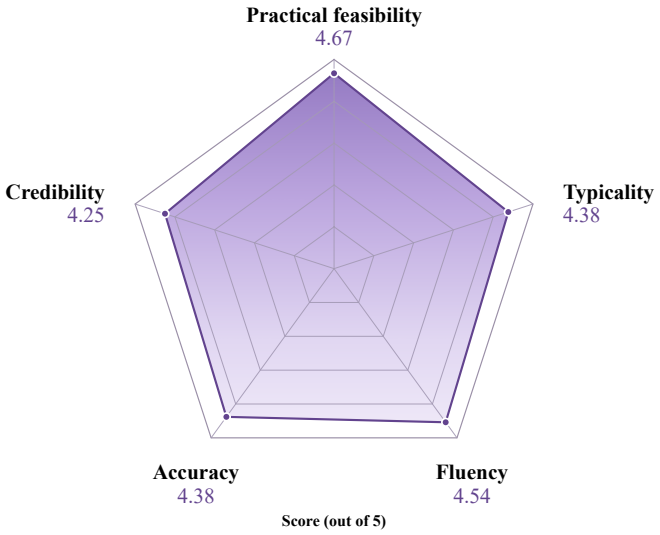


Figure 8: Participant assessment of five perceived benchmark qualities on a five-point scale. These ratings complement task-relevance feedback and do not replace the motion-capture comparison.

Table 14: Participant background distribution for the task-relevance user study ($N = 36$).

Primary background	Count	Percentage	Notes
Robotics	9	25.0%	Manipulation, control, or robot learning.
Human-robot interaction	5	13.9%	HRI, human factors, or collaborative robotics.
Embodied AI / vision-language models	8	22.2%	Embodied agents, VLA models, or simulation.
General computer science	10	27.8%	ML, systems, graphics, or software engineering.
Other STEM	4	11.1%	Engineering or applied science outside the above groups.

Table 15: User-study summary on a five-point Likert scale. "Agree ≥ 4" reports the fraction of participants selecting 4 or 5.

Question group	Mean	Std	Median	Agree ≥ 4
Self-reported familiarity with embodied AI	3.28	1.06	3	44.4%
Self-reported familiarity with human-robot interaction	3.19	1.02	3	41.7%
Self-reported familiarity with robot benchmarks	2.36	0.96	2	22.2%
Rationality of the role design	3.82	0.79	4	72.2%
Rationality of the evaluation metric design	4.50	0.61	5	91.7%
Overall task design approval	4.30	0.70	4	86.1%

Table 16: Role-level task approval statistics and retention decisions. A task card was retained when mean approval was at least 4.0 or when participant comments indicated that only wording or visualization, not the task concept, required revision.

Role group	#Tasks	Mean	Std	Agree ≥ 4	Retention decision
Instructor	3	4.34	0.66	88.0%	All retained; gesture grounding was judged clear and practically meaningful.
Collaborator	7	4.26	0.72	84.1%	All retained; lower-scoring variants were revised for clearer timing labels.
Interferer	3	4.31	0.76	85.2%	All retained; participants identified perturbation recovery as a distinct requirement.

Questionnaire wording. The questionnaire contained the following item groups: (1) "How familiar are you with embodied AI?" (2) "How familiar are you with human-robot interaction?" (3) "How familiar are you with robot benchmarks?" (4) "Are the three role categories Instructor, Collaborator, and Interferer reasonable for describing human-robot collaboration tasks?" (5) "Are the proposed evaluation metrics reasonable for judging collaborative behavior beyond task success?" (6) "For each task card, does this task represent a meaningful and evaluable human-robot collaboration scenario?" Participants could also provide free-form comments; these comments were used only to refine task wording and visualization clarity.

F Interferer Protocol

For Interferer tasks, the perturbation schedule is generated before policy evaluation and replayed identically for every model. Each schedule contains an event type in {`takeover`, `object-displace`, `layout-change`}, a trigger time or trigger predicate, a target object, and a bounded displacement or human approach trajectory. A valid response depends on the event: the robot should yield or pause for takeover attempts, replan after object displacement, and preserve safe clearance when a person moves near the end-effector. We count unsafe contact as a failure even if the object goal is eventually reached.

Table 17: Interferer perturbation events and expected policy responses.

Event type	Trigger	Expected policy behavior	Main metrics
`takeover`	Human enters the manipulation region or reaches toward the target.	Pause or yield while maintaining clearance; resume only after the script permits it.	CFR, HCS, OC
`object-displace`	Human moves the target or distractor object from its nominal state.	Detect the changed state and replan to the updated goal without unsafe contact.	DSR, CSR, CT
`layout-change`	Human changes nearby object layout or blocks part of the workspace.	Preserve the intended goal while avoiding the new obstacle or human region.	CFR, HCS, DSR

Fixed schedule design. Each Interferer evaluation episode samples one perturbation event from the script and stores the trigger predicate, human entry trajectory, target object, displacement vector, and recovery deadline. These schedules are generated before policy rollout and replayed identically for GR00T N1.5, $\pi_{0.5}$, and ACT. For takeover events, the perturbation starts when the robot enters the manipulation approach phase. For object-displace events, the target object is moved after the robot commits to the nominal goal. For layout-change events, a human changes the local obstacle or distractor configuration before the robot reaches the final placement region.

Table 18: Event-level Interferer results. Event-level CSR follows the task-level Interferer split in Table 22: takeover corresponds to N1, object-displace to N2, and layout-change to N3. Recovery time is measured from perturbation trigger to the first valid safe recovery action and is reported over episodes where a recovery action is detected; missing recovery receives the response timeout for DSR but is not included in the recovery-time mean.

Event type	Model	CSR	CFR	HCS	DSR	Recovery time (s)
Takeover	GR00T N1.5-LoRA	0.200	0.800	0.764	0.780	4.3
	$\pi_{0.5}$-LoRA	0.100	0.750	0.736	0.690	5.4
	ACT-FT	0.000	0.724	0.710	0.620	6.2
Object-displace	GR00T N1.5-LoRA	0.100	0.760	0.746	0.710	5.8
	$\pi_{0.5}$-LoRA	0.200	0.770	0.758	0.760	5.7
	ACT-FT	0.100	0.740	0.728	0.700	6.9
Layout-change	GR00T N1.5-LoRA	0.000	0.720	0.728	0.622	7.1
	$\pi_{0.5}$-LoRA	0.000	0.736	0.726	0.650	7.3
	ACT-FT	0.200	0.768	0.752	0.765	6.1

G Experiment Details

G.1 Simulation Setup and Data Splits

All simulation experiments use a Franka Panda manipulator. From the 786 retained trajectories, each task contributes 50 filtered trajectories for policy adaptation and 10 held-out trajectories for evaluation, giving 650 adaptation trajectories and 130 held-out evaluation trajectories. The held-out trajectories change human timing, motion, object pose, and scene initialization relative to the adaptation trajectories. Interferer perturbation schedules are generated independently of the evaluated policy and are fixed across backbones.

Role-level results are computed by first averaging episode metrics within each task and then averaging tasks within the same role group. CT is averaged over successful held-out rollouts only. Rsp, TSync, OC, and safety components follow the failed-episode rules in Table 7. The current results use single-seed adaptation with fixed held-out variants and fixed evaluation seeds; no multi-seed variance is claimed. Observation preprocessing is matched across models whenever the backbone interface supports the same camera, language, and robot-state channels.

G.2 Policy Backbones

For GR00T N1.5 and $\pi_{0.5}$, LoRA adapters are inserted into transformer attention and feed-forward projection layers while the pretrained backbone is frozen. We use matched adaptation settings across role splits whenever the architectures permit. ACT does not use PEFT in our current implementation; instead we use the LeRobot ACT training protocol as a strong low-learning-rate baseline. The ACT configuration uses one observation step, an ImageNet-pretrained ResNet18 visual encoder, an 8-dimensional robot state, an 8-dimensional action, chunk size 50, and 50 action steps per prediction. ACT is trained with standard full fine-tuning. This protocol avoids claiming PEFT support where the ACT implementation is trained by standard fine-tuning.

Table 19: Policy-backbone comparison used in the HRC-SimBench evaluation.

Backbone	Adaptation method	Action/observation interface	Notes
GR00T N1.5	LoRA adaptation	Language-conditioned VLA policy with visual observations and robot state when supported.	Pretrained backbone frozen.
$\pi_{0.5}$	LoRA adaptation	Language-conditioned VLA policy with matched HRC-SimBench observations when supported.	Pretrained backbone frozen.
ACT	Full fine-tuning	Chunked continuous actions with visual encoder and robot state.	LeRobot ACT protocol; no PEFT claim.

Table 20: Backbone adaptation hyperparameters. ACT additionally uses weight decay 10^{-4}, gradient clipping at 10.0, dropout 0.1, KL weight 10.0, and checkpointing every 2k steps. LoRA models use the same held-out evaluation split as ACT.

Model	Train steps	Batch size	LR	LoRA rank	LoRA alpha	Optimizer	GPUs	Adaptation data	Eval episodes
GR00T N1.5-LoRA	30000	64	2×10^{-5}	16	32	AdamW	4 A800	650	130
$\pi_{0.5}$-LoRA	30000	64	1×10^{-5}	16	32	AdamW	4 A800	650	130
ACT-FT	20000	32	1×10^{-5}	–	–	AdamW	4 A800	650	130

G.3 Real-World Adaptation Details

The real-world adaptation study uses a LeRobot SO-100 arm and three task-matched physical scenarios, one from each role group. For each physical task, we collect 20 real teleoperation trajectories. The Real-only condition post-trains GR00T N1.5 on these real demonstrations. The Sim+Real condition first adapts on task-matched HRC-SimBench simulation trajectories generated for the SO-100 setting and then post-trains on the same real demonstrations. Each condition is evaluated over 10 physical trials per task. This study is intended to test transfer potential under limited real data, not to certify deployment safety.

Table 21: Physical SO-100 adaptation protocol used for the real-world transfer study.

Role group	Physical task	Real data	Evaluation
Instructor	Deliver via gesture commands	20 teleoperation trajectories	10 trials
Collaborator	Collaborative object retrieval from a box	20 teleoperation trajectories	10 trials
Interferer	Avoid having someone else do the placement	20 teleoperation trajectories	10 trials

G.4 Main-paper Result References

The role-level HRC-SimBench benchmark summary is reported in main-paper Table 2, and the real-world SO-100 adaptation comparison is reported in main-paper Table 3. To avoid duplicating tables that already appear in the main paper, this appendix only provides task-level, component-level, and trial-level breakdowns. The per-task and safety analyses below use the same role-level targets as main-paper Table 2, while the physical trial details expand the main-paper Table 3 result.

H Additional Results and Failure Analysis

H.1 Per-task Main Results

Table 22 reports per-task CSR using the task IDs defined in Table 10. Table 23 gives the corresponding full metric set. These tables make the role-level averages reported in main-paper Table 2 inspectable and show that the low Interferer performance is distributed across perturbation tasks rather than caused by a single outlier. When CSR is 0, CT is undefined because CT is averaged over successful rollouts only. Rsp, TSync, OC, and Safety can still be reported for failed rollouts because the evaluator logs response events, synchronization windows, script-order events, and safety distances before termination.

Table 22: Per-task collaboration success rate (CSR). Each task has 10 held-out evaluation episodes.

Task ID	Role	GR00T N1.5-LoRA	$\pi_{0.5}$-LoRA	ACT-FT
I1	Instructor	0.700	0.400	0.500
I2	Instructor	0.400	0.700	0.300
I3	Instructor	0.500	0.500	0.500
C1	Collaborator	0.700	0.600	0.500
C2	Collaborator	0.300	0.300	0.200
C3	Collaborator	0.600	0.500	0.400
C4	Collaborator	0.400	0.400	0.400
C5	Collaborator	0.400	0.400	0.300
C6	Collaborator	0.600	0.400	0.400
C7	Collaborator	0.500	0.600	0.400
N1	Interferer	0.200	0.100	0.000
N2	Interferer	0.100	0.200	0.100
N3	Interferer	0.000	0.000	0.200

Table 23: Per-task full benchmark results. CT and Rsp are in seconds; TSync, OC, and Safety are normalized to $[0, 1]$.

Task	Model	CSR	CT	Rsp	TSync	OC	Safety
I1	GR00T N1.5-LoRA	0.700	9.8	0.000	0.490	0.667	0.995
I1	$\pi_{0.5}$-LoRA	0.400	10.1	0.020	0.430	0.667	0.991
I1	ACT-FT	0.500	14.5	0.150	0.350	0.667	0.986
I2	GR00T N1.5-LoRA	0.400	10.9	0.001	0.445	0.667	0.993
I2	$\pi_{0.5}$-LoRA	0.700	10.2	0.028	0.410	0.667	0.990
I2	ACT-FT	0.300	15.0	0.165	0.340	0.667	0.984
I3	GR00T N1.5-LoRA	0.500	10.8	0.002	0.455	0.667	0.991
I3	$\pi_{0.5}$-LoRA	0.500	10.6	0.036	0.423	0.667	0.989
I3	ACT-FT	0.500	14.9	0.180	0.333	0.667	0.982
C1	GR00T N1.5-LoRA	0.700	17.8	0.170	0.415	1.000	1.000
C1	$\pi_{0.5}$-LoRA	0.600	19.4	0.130	0.370	1.000	0.995
C1	ACT-FT	0.500	21.7	0.070	0.300	1.000	0.988
C2	GR00T N1.5-LoRA	0.300	19.4	0.210	0.375	1.000	1.000
C2	$\pi_{0.5}$-LoRA	0.300	20.8	0.155	0.340	1.000	0.993
C2	ACT-FT	0.200	23.1	0.085	0.270	1.000	0.986
C3	GR00T N1.5-LoRA	0.600	18.5	0.210	0.405	1.000	1.000
C3	$\pi_{0.5}$-LoRA	0.500	19.9	0.140	0.365	1.000	0.996
C3	ACT-FT	0.400	22.2	0.075	0.295	1.000	0.989
C4	GR00T N1.5-LoRA	0.400	19.2	0.220	0.380	1.000	1.000
C4	$\pi_{0.5}$-LoRA	0.400	20.5	0.160	0.340	1.000	0.993
C4	ACT-FT	0.400	22.7	0.083	0.275	1.000	0.986

Continued on next page

Table 23 (continued)

Task	Model	CSR	CT	Rsp	TSync	OC	Safety
C5	GR00T N1.5-LoRA	0.400	18.8	0.200	0.395	1.000	1.000
C5	$\pi_{0.5}$-LoRA	0.400	20.1	0.150	0.350	1.000	0.994
C5	ACT-FT	0.300	22.4	0.078	0.280	1.000	0.986
C6	GR00T N1.5-LoRA	0.600	18.1	0.219	0.398	1.000	1.000
C6	$\pi_{0.5}$-LoRA	0.400	19.8	0.152	0.365	1.000	0.995
C6	ACT-FT	0.400	22.0	0.082	0.290	1.000	0.988
C7	GR00T N1.5-LoRA	0.500	19.0	0.220	0.390	1.000	1.000
C7	$\pi_{0.5}$-LoRA	0.600	20.4	0.140	0.350	1.000	0.992
C7	ACT-FT	0.400	22.6	0.076	0.292	1.000	0.983
N1	GR00T N1.5-LoRA	0.200	25.5	0.000	0.290	1.000	0.748
N1	$\pi_{0.5}$-LoRA	0.100	26.0	0.004	0.265	1.000	0.738
N1	ACT-FT	0.000	–	0.012	0.235	1.000	0.731
N2	GR00T N1.5-LoRA	0.100	26.4	0.000	0.275	1.000	0.735
N2	$\pi_{0.5}$-LoRA	0.200	26.5	0.006	0.252	1.000	0.729
N2	ACT-FT	0.100	29.6	0.016	0.224	1.000	0.721
N3	GR00T N1.5-LoRA	0.000	–	0.000	0.263	1.000	0.722
N3	$\pi_{0.5}$-LoRA	0.000	–	0.008	0.245	1.000	0.720
N3	ACT-FT	0.200	30.2	0.020	0.213	1.000	0.711

H.2 Safety Component-wise Results

Table 24 expands the compact Safety column reported in main-paper Table 2. The role-level Safety value is the average of the applicable components, following Eq. 12; not-applicable components are excluded rather than set to zero.

Table 24: Safety component-wise results. CFR is collision-free rate, HCS is human-contact safety, CIR is contradictory-instruction recognition, and DSR is disruption success rate.

Role group	Model	CFR	HCS	CIR	DSR	Safety
Instructor	GR00T N1.5-LoRA	0.996	0.994	0.989	–	0.993
Instructor	$\pi_{0.5}$-LoRA	0.993	0.990	0.987	–	0.990
Instructor	ACT-FT	0.988	0.984	0.980	–	0.984
Collaborator	GR00T N1.5-LoRA	1.000	1.000	–	–	1.000
Collaborator	$\pi_{0.5}$-LoRA	0.997	0.991	–	–	0.994
Collaborator	ACT-FT	0.990	0.984	–	–	0.987
Interferer	GR00T N1.5-LoRA	0.760	0.746	0.730	0.704	0.735
Interferer	$\pi_{0.5}$-LoRA	0.752	0.740	0.724	0.700	0.729
Interferer	ACT-FT	0.744	0.730	0.715	0.695	0.721

To make the role-level differences easier to compare, Figure 9 plots these component scores without assigning zeros to non-applicable entries.

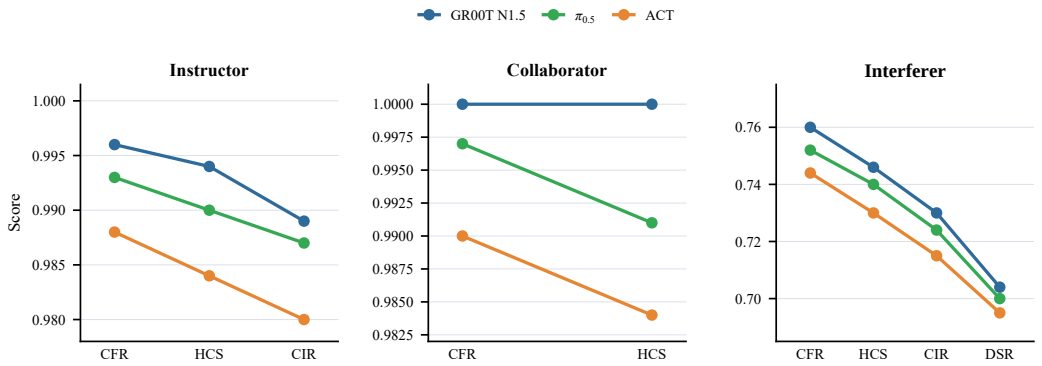


Figure 9: Safety component-wise visualization generated from Table 24. Each panel reports applicable CFR, HCS, CIR, and DSR scores for one role group; non-applicable components are omitted rather than set to zero.

H.3 Failure-mode Breakdown

Qualitative failure cases are categorized into intent misunderstanding, premature action, delayed response, temporal desynchronization, order violation, unsafe proximity, and recovery failure. Table 25 reports the normalized breakdown over failed rollouts for each role and model.

Table 25: Failure-mode breakdown over failed rollouts. Values are percentages and each row sums to 100. "Intent" denotes intent misunderstanding; "Delay" denotes delayed response; "Desync" denotes temporal desynchronization; "Unsafe" denotes unsafe proximity; "Recovery" denotes failure after human interference.

Role	Model	Intent	Premature	Delay	Desync	Order	Unsafe	Recovery
Instructor	GR00T N1.5-LoRA	34	20	16	8	12	8	2
Instructor	$\pi_{0.5}$-LoRA	31	18	18	10	12	8	3
Instructor	ACT-FT	28	22	22	10	10	7	1
Collaborator	GR00T N1.5-LoRA	14	18	20	28	8	10	2
Collaborator	$\pi_{0.5}$-LoRA	15	16	18	30	8	11	3
Collaborator	ACT-FT	12	20	17	25	10	12	4
Interferer	GR00T N1.5-LoRA	6	12	10	14	8	22	28
Interferer	$\pi_{0.5}$-LoRA	7	10	11	13	8	24	27
Interferer	ACT-FT	5	12	10	12	8	25	30

Figure 10 uses the same rows as Table 25 to show how the dominant failure modes shift across roles.

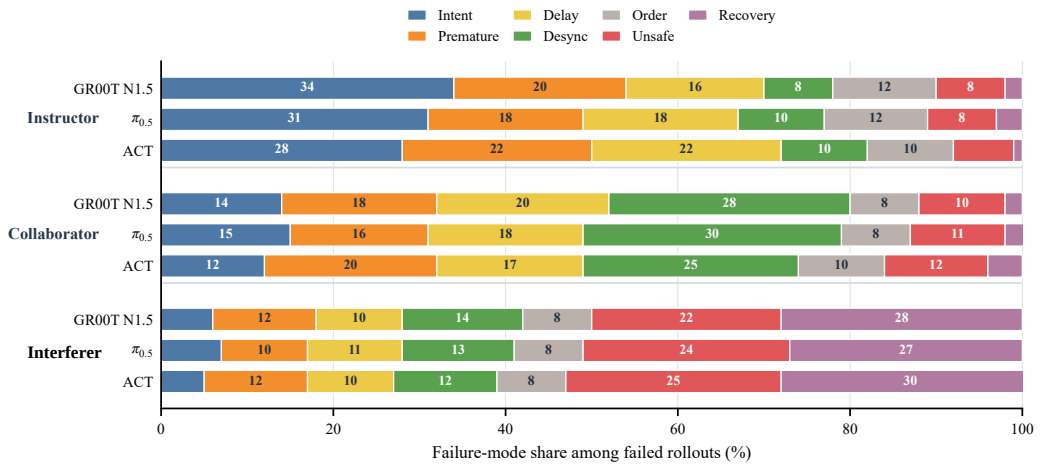


Figure 10: Role- and model-conditioned failure-mode visualization generated from Table 25. Stacked bars show the percentage breakdown over failed rollouts for each role group and policy backbone.

Table 26: Qualitative summary of the numeric failure-mode trends in Table 25.

Failure mode	Instructor	Collaborator	Interferer
Intent misunderstanding	High	Medium	Low
Premature action	Medium	High	Medium
Response delay	Medium	High	Medium
Temporal desynchronization	Low	High	Medium
Order violation	Medium	Medium	Medium
Insufficient safety distance	Low	Medium	High
Recovery failure after interference	Low	Low	High

H.4 HRC-SimBench Data Ablations

Table 27 gives a lightweight ablation on GR00T N1.5 to test whether interaction-structured HRC-SimBench data contributes beyond object-only adaptation. The full HRC-SimBench split improves all role groups, while object-only generated data helps manipulation but leaves temporal and perturbation failures largely unresolved.

Table 27: Lightweight data ablation on GR00T N1.5.

Adaptation setting	Instructor CSR	Collaborator CSR	Interferer CSR
Object-only generated trajectories	0.367	0.133	0.000
Full HRC-SimBench interaction trajectories	**0.533**	**0.500**	**0.100**

H.5 Real-world Trial Details

Table 28 expands the main-paper Table 3 real-world adaptation result with Wilson confidence intervals and dominant failure reasons. The physical experiment is a limited sim-to-real adaptation study rather than a full physical benchmark, so the intervals are intentionally shown to make the small sample size explicit.

Table 28: Real-world trial-level summary for the SO-100 adaptation study. Wilson intervals are included to make uncertainty from 10 trials per task explicit.

Role	Condition	Success	SR	Wilson 95% CI	Dominant failure reasons
Instructor	Real-only	2/10	0.20	[0.06, 0.51]	Wrong gesture target (4), premature move (3), grasp miss (1).
Instructor	Sim+Real	5/10	0.50	[0.24, 0.76]	Late cue response (3), release error (2).
Collaborator	Real-only	1/10	0.10	[0.02, 0.40]	Blocked box approach (4), hand timing miss (3), object slip (2).
Collaborator	Sim+Real	4/10	0.40	[0.17, 0.69]	Retrieval timing miss (3), grasp miss (2), box occlusion (1).
Interferer	Real-only	0/10	0.00	[0.00, 0.28]	Continued after intrusion (5), unsafe proximity (3), no recovery (2).
Interferer	Sim+Real	4/10	0.40	[0.17, 0.69]	Delayed yield (2), incomplete replan (3), final placement miss (1).
Average	Real-only	3/30	0.10	[0.03, 0.26]	Aggregated over three physical tasks.
Average	Sim+Real	13/30	0.43	[0.27, 0.61]	Aggregated over three physical tasks.