

H2RBench: A Real-to-Sim Benchmark for Evaluating Human-to-Robot Transfer

Chuyang Xiao^{1,*}, Haotian Zhan^{1,*}, Sriram Krishna¹, Peilin Meng², Muhammad Zubair Irshad³,
Sergey Zakharov³, and David Held¹

¹Robotics Institute, Carnegie Mellon University

²University of Michigan

³Toyota Research Institute

*Equal contribution; authors are listed in alphabetical order.

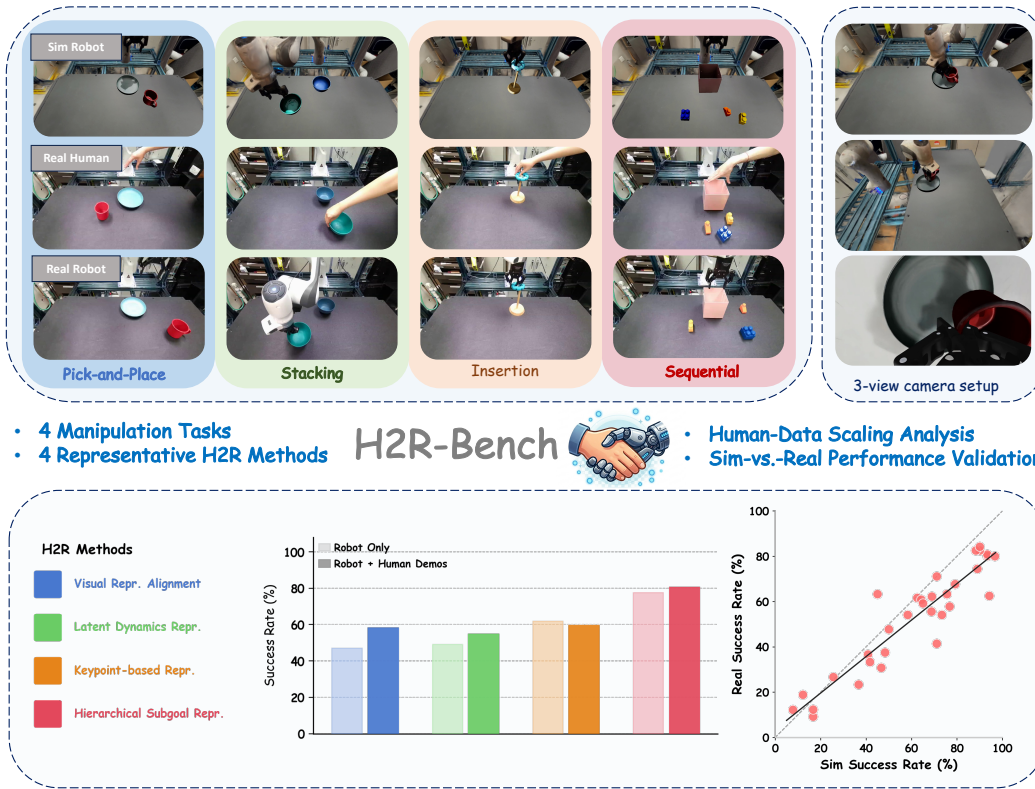


Figure 1: **H2RBench overview.** We propose a real-to-sim benchmark for evaluating human-to-robot (H2R) policy transfer from real human video demonstrations. H2RBench benchmarks multiple H2R methods across four manipulation tasks, enabling systematic analysis of human-data scaling. More videos are available on our [project website](#).

Abstract: Learning robot manipulation policies from human video demonstrations constitutes a promising avenue for scalable robot learning. However, comparing different human-to-robot (H2R) transfer methods remains challenging, as existing approaches are evaluated under different settings, including differing task suites, scene layouts, object instances, and amounts of robot supervision. To address this challenge, we present **H2RBench**, a Real2Sim benchmark for evaluating H2R transfer methods. **H2RBench** provides a standardized protocol built

on real human video demonstrations and simulated robot demonstrations, and includes four manipulation tasks spanning diverse interaction requirements. We evaluate multiple representative H2R transfer methods, each adopting a different strategy for bridging the embodiment gap. Using H2RBench, we systematically characterize how each method scales with the amount of human demonstrations, revealing that methods differ substantially in their ability to leverage additional human data. We further show that simulation performance is broadly predictive of real-world robot performance, with an overall Pearson correlation of $r = 0.89$, Spearman correlation of $\rho = 0.85$ and Mean Maximum Rank Violation (MMRV) of 0.06 across method–task configurations. These results establish H2RBench as a practical and scalable benchmark for comparative H2R evaluation prior to real-world deployment. Videos and code are available on our [project website](#).

Keywords: Human-to-Robot Transfer, Robot Manipulation, Benchmark, Learning from Human Videos, Real-to-Sim

1 Introduction

Learning robot manipulation policies at scale remains fundamentally bottlenecked by the cost of collecting robot demonstrations. Large-scale robot data collection is constrained by hardware cost, teleoperation throughput, and the difficulty of unifying data across diverse robot platforms, control interfaces, and action spaces [1, 2]. Compared with teleoperated robot demonstrations, human video demonstrations offer a promising alternative [3, 4, 5], as they can be collected with off-the-shelf cameras, without robot hardware, and at the natural speed of human manipulation, making them substantially faster and cheaper to scale per task [6, 7, 8]. This opportunity has motivated a growing body of work on task-specific human-to-robot (H2R) transfer [9, 10, 11, 12, 13, 14, 15], which leverages third-person human demonstration videos together with limited robot supervision to train downstream robot manipulation policies. As H2R methods proliferate, a fundamental question remains underexplored: *how should we systematically evaluate and compare them?*

Answering this question is non-trivial for several reasons. First, existing H2R methods are evaluated on heterogeneous task sets with varying object types, scene configurations, and task difficulties, making cross-method comparison unreliable [10, 12, 13]. Second, these methods differ not only in their technical approach, but also in the amount of robot demonstration data they consume, conflating methodological differences with data efficiency differences. Third, real-robot evaluation is expensive, time-consuming, and difficult to reproduce at scale [2].

While simulation offers a scalable alternative, building a meaningful H2R benchmark in simulation is itself challenging, particularly on the human side of the interaction. Most existing H2R transfer methods are designed for human demonstrations captured in the real world: their hand and arm detection front-ends, such as MANO-based hand mesh estimators, are themselves trained on real human imagery [16]. However, reproducing human interaction in simulation is non-trivial, as simulated hands differ from real hands both visually and physically. Existing approaches introduce additional artifacts: replaying recorded hand motion requires hand-model fitting and actuation, leading to reconstruction and contact errors [17], while direct teleoperation introduces retargeting artifacts [18]. As a result, evaluation on simulated hands may not faithfully reflect real-world H2R performance. Moreover, the value of a simulation benchmark hinges on whether relative policy performance in simulation is predictive of real-world performance. Prior work has established this property for robot-only policies—SIMPLER [19], PolaRiS [20], and RobotArena ∞ [21] show that simulation provides a reliable signal of relative real-world policy performance for policies trained on robot demonstrations—but whether the same holds for H2R methods remains unverified.

We address these gaps with **H2RBench**, a real-to-sim benchmark for human-to-robot transfer. We directly use real-world human videos as input to all evaluated methods, mirroring how H2R methods are deployed in practice [10, 13, 12, 11, 14]. Each task environment is reconstructed from real-world

objects and scenes via a real-to-sim pipeline [20]. We vary the amount of human demonstrations to characterize how each method scales with human data. Finally, we validate the benchmark by comparing relative policy performance in simulation and on the real robot.

Our evaluation yields several key findings. First, for current human-to-robot methods, human demonstrations most consistently improve high-level interaction behaviors, while precise late-stage control in contact-rich manipulation remains significantly more challenging. Second, H2R methods differ substantially in how effectively they leverage additional human demonstrations: some scale consistently with increasing human data, while others exhibit only limited gains. Third, we show that simulation performance is broadly predictive of relative real-world policy performance for H2R methods, supporting the use of H2RBench as a practical proxy for real-world evaluation.

In summary, we make the following contributions:

- A real-to-sim benchmark for H2R transfer consisting of manipulation tasks reconstructed from real-world scenes, spanning diverse precision requirements, contact dynamics, and task horizons, with task-specific human and robot demonstrations collected for each task.
- Empirical insights into how H2R methods scale with increasing amounts of human demonstration data, and a taxonomy of failure modes that informs future H2R algorithm design.
- An empirical Sim-vs.-Real validation showing that simulation performance is broadly predictive of relative real-world policy performance for H2R methods.

2 Related Work

Robot Manipulation Benchmarks and Sim-vs.-Real Evaluation. A large family of manipulation benchmarks evaluates policies trained from robot demonstrations, ranging from broad task suites [22, 23, 24, 25, 26] to imitation-learning-focused protocols [27, 28]. More recent work studies whether simulation can serve as a reliable proxy for evaluating real-world robot policies. SIMPLER [19] introduced this paradigm by constructing simulated counterparts of real robot setups and showing strong correlation between simulation and real-world policy performance. PolaRiS [20] further scales this idea through high-fidelity neural reconstruction of real-world scenes, while RoboArena ∞ [21] studies scalable automatic evaluation in generated real-to-sim environments. However, these benchmarks focus exclusively on robot-only policies and do not evaluate methods that use human videos as a primary source of supervision.

Among prior benchmarks, RoboTube [7] is the most closely related to H2RBench, as it collects human videos together with simulated environments for robot learning from human demonstrations. However, RoboTube primarily focuses on reward-learning approaches that leverage human videos within online reinforcement learning settings; in contrast, we focus on H2R transfer methods trained with limited robot data and more human data. Consequently, RoboTube does not standardize robot-demonstration budgets or explicitly vary the amount of human supervision across methods, making it difficult to systematically measure how effectively different H2R transfer methods leverage additional human demonstrations. Further, H2RBench studies whether simulation performance reliably predicts real-world performance for H2R methods, providing the first systematic Sim-vs.-Real validation in the H2R setting. A broader comparison between H2RBench and prior real-to-sim manipulation benchmarks is summarized in Table 1.

Human-to-Robot Policy Transfer. Learning robot manipulation policies from human video demonstrations has emerged as a promising paradigm for robot learning. However, transferring behavior from human videos to robot policies remains challenging due to the embodiment gap between humans and robots. Prior work has explored using human videos for robot learning through visual representation pretraining [4, 29], reward learning [30, 31], and extracting affordances or interaction priors [5, 32]. More recent H2R methods directly address the embodiment gap between human demonstrations and robot execution through different modeling choices and training strategies [13, 12, 14, 15, 10]. While these methods demonstrate promising results, they are evaluated on different tasks, with different amounts of robot supervision, making direct comparison difficult.

Table 1: Comparison of real-to-sim robot manipulation benchmarks across key benchmark properties for H2R evaluation. **H2R**: human-to-robot transfer evaluation; **Quant. Sim-vs.-Real**: quantitative sim-vs.-real performance validation; **Multi-stage SR**: stage-wise success metrics.

Benchmark	Primary Focus	H2R	Quant. Sim-vs.-Real	Multi-stage SR
SIMPLER [19]	Sim-real correlation for pretrained policies	✗	✓	✗
PolaRiS [20]	Neural real-to-sim reconstruction	✗	✓	✓
RobotArena ∞ [21]	Scalable evaluation of generalist policies	✗	✓	✓
RoboTube [7]	Human-video reward learning	✓	✗	✗
H2RBench (ours)	Controlled H2R evaluation with human data scaling	✓	✓	✓

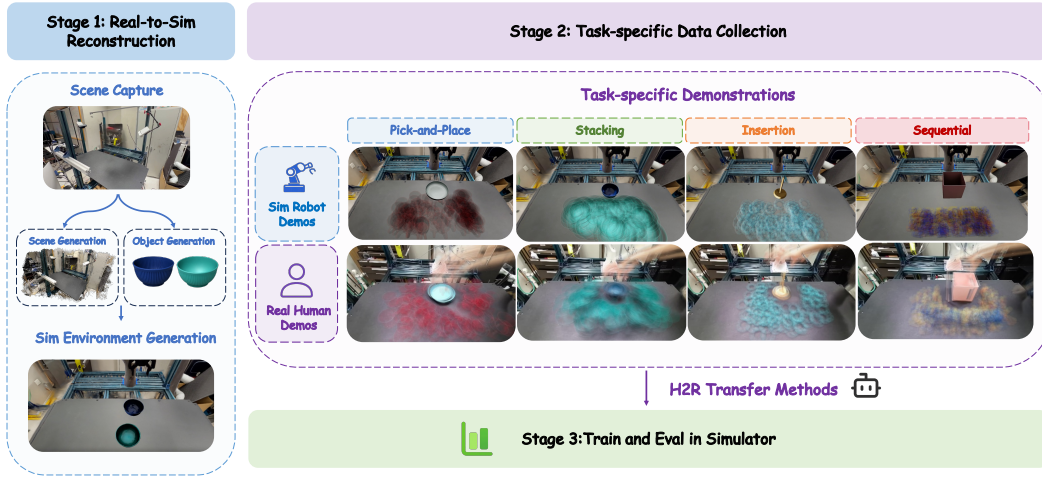


Figure 2: **Overview of the H2RBench evaluation pipeline.** We reconstruct real-world workspaces into Isaac Lab simulation environments and collect task-specific human and robot demonstrations across four manipulation tasks with diverse object placement distributions. Methods are then trained and evaluated in simulation under a shared benchmark protocol for controlled H2R comparison.

H2RBench provides a unified evaluation framework for fair and reproducible comparison across H2R transfer methods under standardized data and evaluation settings.

3 H2RBench

3.1 Overview

H2RBench provides a standardized benchmark for evaluating task-specific H2R transfer methods that use human video demonstrations together with limited robot demonstrations to learn robot manipulation policies. The benchmark consists of four manipulation tasks, each implemented as a simulation environment reconstructed from a real-world scene via a real-to-sim pipeline. All baselines are trained on the same dataset of real human video demonstrations together with simulated robot demonstrations under a shared training and evaluation protocol. Two design choices distinguish H2RBench from existing manipulation benchmarks. First, it targets methods that learn from human videos rather than robot demonstrations alone. Second, we hold the robot demonstration budget fixed across methods while varying the number of human demonstrations, enabling controlled analysis of how different methods scale with human supervision—a core property of H2R transfer that prior benchmarks do not explicitly measure.

H2RBench provides a unified observation-action interface while supporting diverse H2R policy representations. Observations include synchronized multi-view RGB-D images, robot proprioception, and language; human demonstrations use front and side views, while robot demonstrations additionally include wrist-mounted observations. Actions are represented in the robot base frame as 3D translation, 6D rotation [33], and a binary gripper command, following prior H2R methods [13, 12].

3.2 Task Suite

We design four tasks that cover a broad range of manipulation tasks requiring varying levels of required precision and task horizon. Table 2 summarizes the key properties of each task.

Table 2: Overview of the four manipulation tasks in H2RBench, including task horizon, precision requirement, the fixed number of robot demonstrations, and the maximum number of human demonstrations used in scaling experiments.

Task	Horizon	Precision Requirement	Robot Demos	Max Human Demos
Pick-and-Place (Mug → Plate)	Short	Coarse	40	100
Stacking (Bowl → Bowl)	Short	Moderate	100	300
Insertion (Donut → Peg)	Short	Precise	100	300
Sequential Manipulation (Blocks → Box)	Long	Moderate	100	300

3.3 Real-to-Sim Reconstruction and Data Collection

Scene Reconstruction. As illustrated in Fig. 2, H2RBench consists of three stages: real-to-sim reconstruction, task-specific data collection, and simulation-based training and evaluation. All task environments are built with PolaRiS [20] on top of NVIDIA Isaac Lab [34]. For scene reconstruction, we replace the original PolaRiS pipeline with Marble, which produces higher-quality 3D meshes and Gaussian Splatting reconstructions, resulting in more visually consistent and geometrically faithful simulation assets. Interactive objects are reconstructed from multiview reference images using TRELIS [35], then imported into the simulator and scaled to match their real-world dimensions.

Data Collection. For each task, we collect human and robot demonstrations. Human demonstrations are recorded as third-person RGB videos from multiple operators performing the task in the real-world workspace using fixed camera viewpoints consistent with the reconstructed simulation scene. Robot demonstrations are collected in both simulation and the real world with an extra wrist camera. In simulation, trajectories are generated using the cuRobo motion planner [36]. In the real world, demonstrations are collected using a Franka Emika Panda arm via the GELLO teleoperation interface [37]. Additional details on the data collection setup are provided in Sec. A.1.

3.4 Evaluation Protocol

Evaluation Metric. We evaluate each policy using a multi-stage success rate, where each task is decomposed into cumulative stages reflecting task progress. We report both final-stage success rate and the average success rate across all stages as an overall measure of task completion.

Sim-vs.-Real Validation. For each task and each method, we additionally train a separate policy using a real-world training pipeline consisting of real human demonstrations together with real robot demonstrations, and evaluate the resulting policy on the physical robot. We then compare the rankings of methods obtained under the simulation pipeline and the real-world pipeline using rank-correlation metrics (Pearson r , Spearman ρ , Mean Maximum Rank Violation (MMRV)). This evaluation quantifies whether H2RBench provides a reliable proxy for real-world H2R method comparison.

4 Methods Evaluated

Phantom (Visual Representation Alignment). Phantom [13] edits human demonstrations to visually align them with robot observations by inpainting the human arm and overlaying a rendered robot arm before imitation learning.

Point Policy (Keypoint-Based Representation). Point Policy [12] represents human and robot behavior using shared 3D keypoints extracted from multi-view observations and predicts future robot keypoints directly in this representation space.

AMPLIFY (Latent Dynamics Representation). AMPLIFY [14] decouples interaction dynamics modeling from action inference by first learning a latent dynamics representation from videos and then predicting robot actions from that representation using robot demonstrations.

GHOST (Hierarchical Subgoal Representation). GHOST [15] decomposes H2R transfer into a high-level subgoal planner learned from human and robot demonstrations and a low-level robot controller trained to execute those subgoals from robot demonstrations.

5 Experiments

We use H2RBench to systematically evaluate human-to-robot (H2R) transfer methods. Beyond comparing final task performance, our goal is to characterize how human supervision interacts with different embodiment-bridging strategies across manipulation regimes with varying precision requirements and horizon length. Our experiments are organized around three questions:

Q1: Capability and human transfer effectiveness. How do H2R methods compare in overall task performance, and to what extent do human demonstrations improve or degrade execution relative to robot-only training?

Q2: Scaling with human supervision. How does policy performance evolve as the amount of human supervision increases?

Q3: Sim-vs.-Real fidelity. Does evaluation in H2RBench reliably predict real-world policy performance?

5.1 Capability and Human Transfer Effectiveness (Q1)

Setup. Unless otherwise specified, all methods use the recommended hyperparameters from their original publications and are evaluated under the H2RBench protocol in Section 3.4 over 90 randomized rollouts across 3 seeds. To quantify the effect of human supervision, we compare robot-only and robot+human training under identical architectures and hyperparameters. We report both average task progression (**Prog.**) and final-stage success rate (**SR**). Since some methods already perform strongly under robot-only training, absolute success-rate improvements may not fully reflect the benefit of human demonstrations. We therefore additionally report the log-odds transfer gain:

$$\Delta_{\text{logit}} = \log \frac{p_{R+H}}{1 - p_{R+H}} - \log \frac{p_R}{1 - p_R},$$

where p_R and p_{R+H} denote the success rates under robot-only and robot+human training, respectively. Compared with raw success-rate differences, this metric better accounts for gains near performance saturation and provides a more balanced measure of human transfer effectiveness.

Results. Table 3 compares overall task performance across methods under full human supervision. Overall, GHOST [15] achieves the strongest and most consistent performance across tasks, while AMPLIFY [14] performs particularly well on stacking and Point Policy [12] remains competitive on pick-and-place and stacking. Across methods, approaches leveraging hierarchical decomposition or explicit subgoal representations tend to achieve stronger and more robust performance, suggesting that structured intermediate abstractions are effective for bridging embodiment differences in manipulation. More detailed qualitative analysis is provided in Appendix C.

Fig. 3 further shows that the benefit of human supervision varies substantially across both tasks and methods. Human demonstrations consistently improve performance on pick-and-place and remain beneficial for stacking, indicating that current H2R methods effectively transfer high-level spatial intent. In contrast, performance often degrades for insertion, while improvements are more limited for sequential manipulation, where fine-grained contact dynamics and error accumulation across repeated interaction steps remain challenging. Overall, while human demonstrations generally improve task performance, H2R methods differ substantially in how effectively they leverage addi-

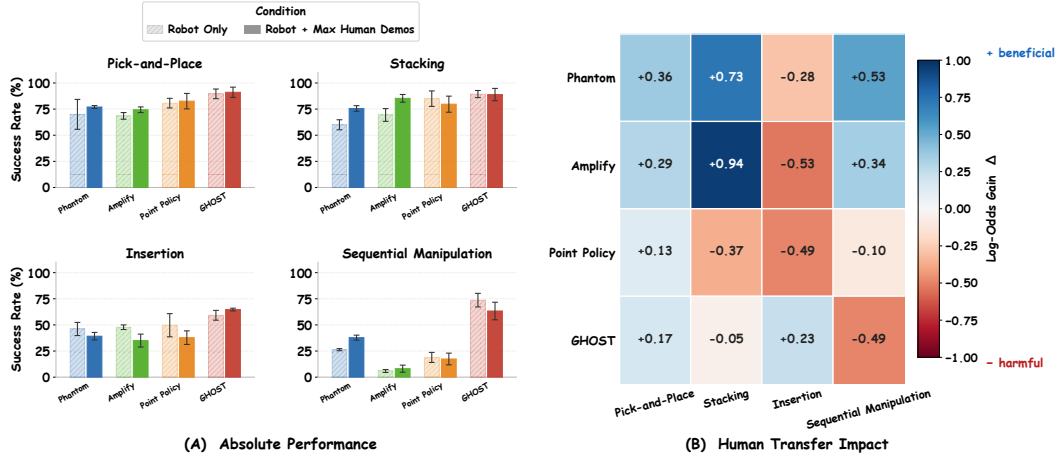


Figure 3: Capability and human transfer impact across H2R methods and tasks. **Left**: absolute task performance. **Right**: human transfer impact (log-odds gain)

tional human data, highlighting the importance of evaluating both overall performance and transfer effectiveness in H2R benchmarks.

Table 3: Multi-seed benchmark results under robot + maximum human demonstration training. We report mean and standard deviation across three random seeds for task progression (**Prog.**) and final-stage success rate (**SR**).

Method	Pick-and-Place		Stacking		Insertion		Sequential Manipulation	
	Prog.	SR	Prog.	SR	Prog.	SR	Prog.	SR
Phantom [13]	77.0 ± 1.4	56.7 ± 2.7	75.6 ± 2.7	25.6 ± 6.8	39.2 ± 3.6	7.8 ± 4.2	37.8 ± 2.4	13.3 ± 4.7
AMPLIFY [14]	74.4 ± 2.7	58.9 ± 4.2	85.3 ± 3.7	61.1 ± 9.6	35.0 ± 6.2	2.2 ± 1.6	8.1 ± 3.4	0.0 ± 0.0
Point Policy [12]	82.6 ± 7.4	62.2 ± 11.7	79.7 ± 7.7	61.1 ± 17.1	37.8 ± 6.5	6.7 ± 11.5	17.4 ± 5.6	0.0 ± 0.0
GHOST [15]	91.1 ± 4.9	82.2 ± 6.9	88.9 ± 5.9	76.7 ± 8.8	64.7 ± 1.3	34.5 ± 6.9	63.3 ± 8.4	38.9 ± 15.0

5.2 Scaling with Human Supervision (Q2)

Setup. We vary the number of human demonstrations while keeping the robot demonstration budget fixed. We evaluate scaling trends using average task progression (**Prog.**) success rate.

Results. Figure 4 shows scaling performance as a function of the number of human demonstrations for each task. For simpler manipulation tasks, most methods improve steadily as additional human demonstrations are added, suggesting that current H2R approaches can effectively leverage human video to improve coarse transport and placement behavior. In contrast, scaling trends become less consistent for more precision-sensitive tasks. Stacking exhibits strong method-dependent behavior, with some methods continuing to improve while others plateau. Insertion further amplifies this effect, where increasing human demonstrations often provides limited or unstable gains. Sequential manipulation reveals diminishing returns under long-horizon execution. While some methods scale well with more human data, others saturate or fluctuate with increased supervision, suggesting that compounding execution errors remain difficult to correct through additional human demonstrations alone. Overall, these results suggest that the value of scaling human demonstrations depends strongly on both task structure and the embodiment-bridging strategy used by the H2R method.

5.3 Sim-vs.-Real Fidelity (Q3)

Setup. For each method-task pair, we train: (i) a *simulation-pipeline* (π_{sim}) policy using real human demonstrations and simulated robot demonstrations, and (ii) a *real-world-pipeline* (π_{real}) policy us-

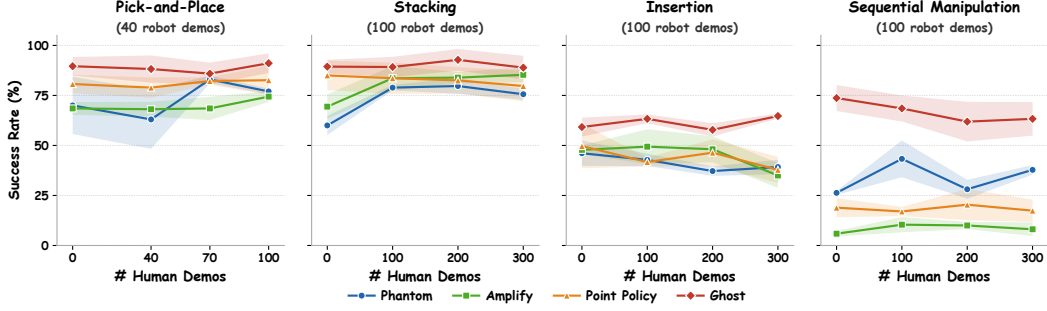


Figure 4: Scaling performance with respect to the number of human demonstrations.

ing real human demonstrations and real robot demonstrations. π_{sim} is evaluated in the reconstructed environment, while π_{real} is evaluated on the physical robot. We then measure Sim-vs.-Real correspondence between simulated and real-world success rates.

Results. Fig. 5 shows a strong correlation between simulated and real-world policy performance across method-task pairs. Table 4 further shows high Pearson r and Spearman ρ , together with a low Mean Maximum Rank Violation (MMRV) of 0.06, indicating that simulation reliably preserves both performance trends and method rankings. Overall, these results support **H2RBench** as a practical proxy for H2R evaluation, enabling efficient benchmarking in simulation while reducing the cost of real-world experimentation.

6 Limitations

H2RBench has several limitations. First, it currently focuses on tabletop manipulation with primarily rigid objects and does not yet cover deformable manipulation, articulated objects, or bi-manual coordination. Second, H2RBench evaluates H2R methods using task-specific human demonstrations collected under controlled third-person camera setups; extending benchmark evaluation to egocentric video or human demonstrations collected “in-the-wild” remains an important direction for future work. Finally, H2RBench inherits limitations of current real-to-sim reconstruction pipelines, including imperfect physical fidelity and limited support for complex dynamics.

7 Conclusion

We introduced **H2RBench**, a real-to-sim benchmark for evaluating human-to-robot transfer under a unified evaluation framework. Across four manipulation tasks and representative H2R methods, our experiments show that the benefit of human demonstrations is highly task- and method-dependent: current H2R methods transfer high-level interaction behavior more reliably than precise control. We further show that **H2RBench** is broadly predictive of relative real-world policy performance, supporting the use of real-to-sim environments as a practical proxy for comparative H2R evaluation.

Table 4: Sim-vs.-Real fidelity metrics under **Robot + Max Human Demos**.

Pearson $r \uparrow$	Spearman $\rho \uparrow$	MMRV $\downarrow$
0.894	0.851	0.06

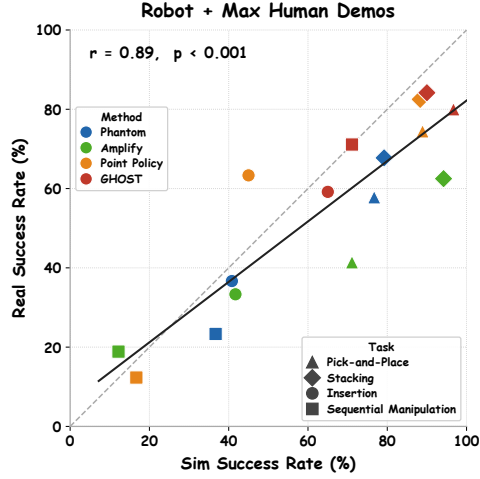


Figure 5: Sim-vs.-Real correlation between simulated and real-world policy performance.

Acknowledgments

This material is based upon work supported by the Toyota Research Institute. David Held holds concurrent appointments as a Professor at CMU and as an Amazon Scholar. This paper describes work performed at CMU and is not associated with Amazon.

References

- [1] A. O’Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.
- [2] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.
- [3] S. Kareer, K. Pertsch, J. Darpinian, J. Hoffman, D. Xu, S. Levine, C. Finn, and S. Nair. Emergence of human to robot transfer in vision-language-action models. *arXiv preprint arXiv:2512.22414*, 2025.
- [4] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.
- [5] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13778–13790, 2023.
- [6] S. Kareer, D. Patel, R. Punamiya, P. Mathur, S. Cheng, C. Wang, J. Hoffman, and D. Xu. Egomimic: Scaling imitation learning via egocentric video. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13226–13233. IEEE, 2025.
- [7] H. Xiong, H. Fu, J. Zhang, C. Bao, Q. Zhang, Y. Huang, W. Xu, A. Garg, and C. Lu. Robotube: Learning household manipulation from human videos with simulated twin environments. In *6th Annual Conference on Robot Learning*, 2022.
- [8] V. Liu, A. Adeniji, H. Zhan, S. Haldar, R. Bhirangi, P. Abbeel, and L. Pinto. Egozero: Robot learning from smart glasses, 2025. URL <https://arxiv.org/abs/2505.20290>.
- [9] M. Xu, Z. Xu, C. Chi, M. Veloso, and S. Song. Xskill: Cross embodiment skill discovery. In *Conference on robot learning*, pages 3536–3555. PMLR, 2023.
- [10] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023.
- [11] J. Ren, P. Sundaresan, D. Sadigh, S. Choudhury, and J. Bohg. Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8802–8810. IEEE, 2025.
- [12] S. Haldar and L. Pinto. Point policy: Unifying observations and actions with key points for robot manipulation. *arXiv preprint arXiv:2502.20391*, 2025.
- [13] M. Lepert, J. Fang, and J. Bohg. Phantom: Training robots without robots using only human videos. *arXiv preprint arXiv:2503.00779*, 2025.
- [14] J. A. Collins, L. Cheng, K. Aneja, A. Wilcox, B. Joffe, and A. Garg. Amplify: Actionless motion priors for robot learning from videos. *arXiv preprint arXiv:2506.14198*, 2025.

- [15] S. Krishna, B. Eisner, H. Zhan, Y. Yuan, H. Zhen, C. Gan, S. Tulsiani, and D. Held. Ghost: Hierarchical sub-goal policies for generalizing robot manipulation. In *Robotics: Science and Systems (RSS)*, 2026.
- [16] G. Pavlakos, D. Shan, I. Radosavovic, A. Kanazawa, D. Fouhey, and J. Malik. Reconstructing hands in 3d with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9826–9836, 2024.
- [17] X. Xu, J. Sun, S. Chen, L. Ma, K. Sun, B. Zhao, J. Yuan, S. Yi, H. Zhu, Y. Lu, et al. Dexcanvas: Bridging human demonstrations and robot learning for dexterous manipulation. *arXiv preprint arXiv:2510.15786*, 2025.
- [18] Y. Qin, H. Su, and X. Wang. From one hand to multiple hands: Imitation learning for dexterous manipulation from single-camera teleoperation. *IEEE Robotics and Automation Letters*, 7(4): 10873–10881, 2022.
- [19] X. Li, K. Hsu, J. Gu, K. Pertsch, O. Mees, H. R. Walke, C. Fu, I. Lunawat, I. Sieh, S. Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.
- [20] A. Jain, M. Zhang, K. Arora, W. Chen, M. Torne, M. Z. Irshad, S. Zakharov, Y. Wang, S. Levine, C. Finn, et al. Polaris: Scalable real-to-sim evaluations for generalist robot policies. *arXiv preprint arXiv:2512.16881*, 2025.
- [21] Y. Jangir, Y. Zhang, P.-C. Lo, K. Yamazaki, C. Zhang, K.-H. Tu, T.-W. Ke, L. Ke, Y. Bisk, and K. Fragkiadaki. Robotarena ∞ : Scalable robot benchmarking via real-to-sim translation. *arXiv preprint arXiv:2510.23571*, 2025.
- [22] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.
- [23] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.
- [24] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao, et al. Maniskill2: A unified benchmark for generalizable manipulation skills. *arXiv preprint arXiv:2302.04659*, 2023.
- [25] S. Tao, F. Xiang, A. Shukla, Y. Qin, X. Hinrichsen, X. Yuan, C. Bao, X. Lin, Y. Liu, T.-k. Chan, et al. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv preprint arXiv:2410.00425*, 2024.
- [26] M. Heo, Y. Lee, D. Lee, and J. J. Lim. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. *The International Journal of Robotics Research*, 44 (10-11):1863–1891, 2025.
- [27] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.
- [28] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.
- [29] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pages 416–426. PMLR, 2023.

- [30] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1134–1141. IEEE, 2018.
- [31] K. Zakka, A. Zeng, P. Florence, J. Tompson, J. Bohg, and D. Dwibedi. Xirl: Cross-embodiment inverse reinforcement learning. In *Conference on Robot Learning*, pages 537–546. PMLR, 2022.
- [32] R. Mendonca, S. Bahl, and D. Pathak. Structured world models from human videos. *arXiv preprint arXiv:2308.10901*, 2023.
- [33] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019.
- [34] M. Mittal, P. Roth, J. Tigue, A. Richard, O. Zhang, P. Du, A. Serrano-Munoz, X. Yao, R. Zurbrugg, N. Rudin, et al. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.
- [35] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, and J. Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21469–21480, 2025.
- [36] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. Van Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos, et al. curobo: Parallelized collision-free minimum-jerk robot motion generation. *arXiv preprint arXiv:2310.17274*, 2023.
- [37] P. Wu, Y. Shentu, Z. Yi, X. Lin, and P. Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12156–12163. IEEE, 2024.
- [38] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2024.
- [39] R. Punamiya, D. Patel, P. Aphiwetsa, P. Kuppili, L. Y. Zhu, S. Kareer, J. Hoffman, and D. Xu. Egobridge: Domain adaptation for generalizable imitation from egocentric human data. In *Human to Robot: Workshop on Sensorizing, Modeling, and Learning from Humans*, 2025.
- [40] L. Tang, M. Jia, Q. Wang, C. P. Phoo, and B. Hariharan. Emergent correspondence from image diffusion, 2023. URL <https://arxiv.org/abs/2306.03881>.
- [41] N. Karaev, I. Rocco, B. Graham, N. Neverova, A. Vedaldi, and C. Rupprecht. Cotracker: It is better to track together, 2024. URL <https://arxiv.org/abs/2307.07635>.

Appendix

A Implementation Details

A.1 Detailed data collection pipeline

Human Demonstrations. For each task, we collect real-world human demonstrations from multiple operators performing the task in the reconstructed workspace. H2RBench does not assume paired human-to-robot trajectories. Instead, we evaluate whether H2R methods can transfer useful behavior priors from human demonstrations under a shared observation-action interface.

Robot Demonstrations. Robot demonstrations are collected in both simulation and the real world. In simulation, we generate robot trajectories using the cuRobo motion planner [36]. We use 40 demonstrations for pick-and-place and 100 for the remaining tasks, and keep this robot demonstration budget fixed across methods for fair comparison. In the real world, demonstrations are collected using a Franka Emika Panda arm through the GELLO teleoperation interface [37].

A.2 Detailed Simulator Parameters

Unless otherwise specified, all environments use the default physics configuration from Isaac Lab / Isaac Sim. We make only minimal task-specific adjustments to improve sim-to-real consistency.

In particular, we tune contact-related parameters such as object friction and gripper fingertip friction for selected objects when necessary. These adjustments primarily improve grasp stability and contact behavior in tasks involving stacking or insertion, where small physical mismatches can significantly affect success.

We avoid extensive per-task tuning and keep the simulator settings as close as possible to the default Isaac Sim configuration to maintain a consistent evaluation protocol across tasks.

A.3 Detailed H2R Methods Implementation

Unless otherwise noted, all H2R baselines are implemented following the official codebases and default hyperparameter settings reported in their original papers. To ensure fair comparison, we preserve each method’s original architecture and training recipe whenever possible, and introduce only minimal modifications necessary to adapt the method to the H2RBench task setup and observation modalities. For hand pose estimation model, we just use the same model specified in the original paper. Below, we highlight several method-specific implementation details, particularly camera-view usage and preprocessing choices.

Table 5: Camera views used during training for each H2R method.

Method	Front	Left	Wrist
Phantom	✓	✗	✗
Point Policy	✓	✓	✗
AMPLIFY	✓	✓	✗
GHOST	✓	✓	✓

A.3.1 Camera View Usage

Although all methods are evaluated under the same task suite and data collection protocol, the camera views used during training differ across methods due to differences in how each method bridges the embodiment gap between human and robot demonstrations. Table 5 summarizes the camera views used by each evaluated method. In general, whether a camera view can be used depends on whether human demonstrations can be consistently aligned with robot observations in that perspective.

Phantom [13] is trained using only the **front view**, since its robot-inpainted human videos are generated through end-effector retargeting and inverse kinematics, which do not guarantee temporally consistent rendering across multiple viewpoints. AMPLIFY [14] and Point Policy [12] use both **front-view** and **left-view** observations, as they learn cross-embodiment representations directly from shared human and robot videos without requiring explicit rendered alignment. Neither method uses the robot wrist camera because human demonstrations do not provide a paired egocentric view. For AMPLIFY specifically, although the inverse dynamics model is trained using robot-only demonstrations, it depends on motion tracks predicted by upstream modules trained only on the shared front-view and left-view videos, preventing consistent use of wrist-camera observations. GHOST [15] additionally uses the robot **wrist camera** during low-level policy training, since this stage is trained on robot-only demonstrations and does not require paired human observations.

Overall, as shown in Table 5, these differences reflect an inherent trade-off in H2R learning: methods that require explicit human-robot alignment are restricted to shared external viewpoints, while hierarchical methods with robot-only downstream control can additionally leverage robot-specific egocentric observations such as the wrist camera.

A.3.2 Practical Annotation Cost Analysis

Beyond task performance, H2R methods also differ in the amount of manual supervision required during data preprocessing and annotation. Since annotation overhead directly affects the scalability and practicality of H2R pipelines, we summarize the manual labeling effort required by each evaluated method.

Phantom [13], AMPLIFY [14], and GHOST [15] primarily rely on automatic data processing pipelines and require minimal manual annotation beyond dataset collection. Point Policy [12] requires manual keypoint initialization on the first frame of a single demonstration sequence per task, resulting in annotation effort that scales linearly with the number of tasks.

GHOST [15] uses gripper-open close to automatically get subgoal key frames; for human demos, we compute the subgoal keyframes when the distance between the thumb and index fingers is less than a threshold for a given number of frames. The threshold is selected manually on a per-task basis.

A.4 Task Details

H2RBench includes four manipulation tasks with varying interaction complexity. For each task, we define a language instruction and a step-wise evaluation rubric used to compute task progression and final success.

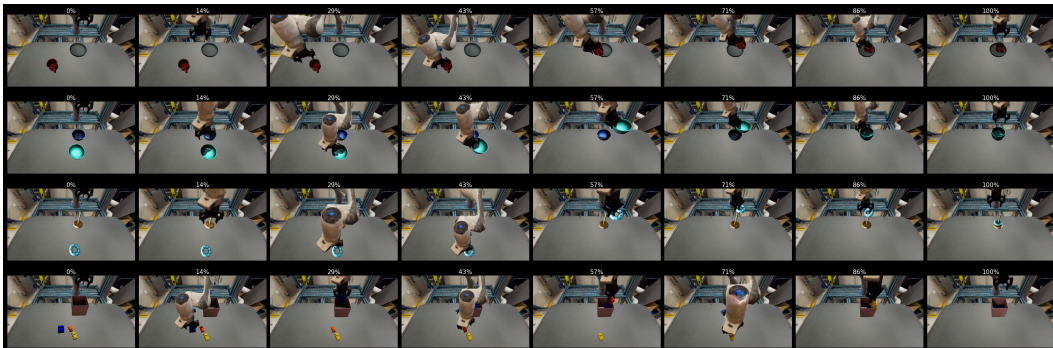


Figure 6: Visualization of the task progression for each task. We show representative intermediate states from the initial scene configuration to successful task completion.

Table 6 summarizes the task instructions and evaluation rubric for each task. Figure 6 shows representative intermediate states during task execution from initial configuration to successful comple-

tion. These visualizations are illustrative and may contain finer-grained transitions than the discrete stages used for evaluation.

Table 6: Task prompts and corresponding step-by-step evaluation rubrics for the four H2RBench tasks. Scores are normalized between 0 and 1 for comparability across environments.

Task	Instruction	Step-by-Step Rubric
Pick-and-Place	Put the mug onto the plate	1. Reach for the red cup 2. Lift the red cup 3. Place the red cup onto the blue plate
Stacking	Stack the bowl onto the other bowl	1. Reach for the green bowl 2. Lift the green bowl 3. Reach for the blue bowl 4. Place the blue bowl on top of the green bowl
Insertion	Insert the donut onto the peg	1. Reach for the blue donut 2. Lift the blue donut 3. Move the donut above the peg 4. Insert the donut onto the peg
Sequential Manipulation	Pick up the toys and put them into the box	1. Place the blue toy into the pink box 2. Place the orange toy into the pink box 3. Place the yellow toy into the pink box

B Additional Results

B.1 Additional Stage-wise Transfer Analysis

To localize where human demonstrations affect execution, we further measure the stage-wise transfer impact:

$$\Delta \text{SR}_s = \text{SR}_s^{R+H} - \text{SR}_s^R,$$

where SR_s^{R+H} and SR_s^R denote the robot+human and robot-only success rates at stage s , respectively.

Analysis. Fig. 7 shows that the effect of human demonstrations varies across execution stages. Across tasks, transfer gains are typically strongest in earlier or intermediate stages and become weaker or less consistent near task completion.

For pick-and-place and stacking, human demonstrations mainly improve transport and placement behavior, suggesting that current H2R methods effectively transfer high-level spatial intent. In contrast, gains are limited in later stages requiring precise pose alignment or contact-rich interaction, especially for insertion.

Sequential manipulation shows a similar trend: improvements are often largest early in execution but diminish over repeated interaction steps as errors accumulate.

Overall, these results suggest that current H2R methods transfer coarse spatial behavior more effectively than fine-grained contact dynamics, with embodiment mismatch becoming most evident in later stages and long-horizon execution.

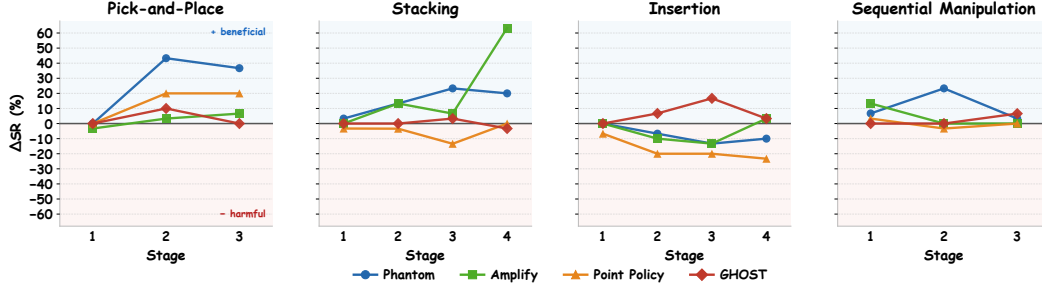


Figure 7: Stage-wise human transfer impact measured as $\Delta SR_s = SR_s^{R+H} - SR_s^R$. Positive values indicate that adding human demonstrations improves stage completion, while negative values indicate degraded performance.

B.2 Max Human Demos VS Best number of Human Demos

Table 7: Extended benchmark results across all four tasks. For each task, we report performance under the maximum human-demo setting (**Prog.@MaxH**), the best progression score achieved across the human-demo sweep (**Best Prog.**), and the corresponding number of human demonstrations (**# Human**).

Method	Pick-and-Place			Stacking		
	Prog.@MaxH	Best Prog.	# Human	Prog.@MaxH	Best Prog.	# Human
Phantom [13]	76.7	76.7	100	79.2	79.2	300
Point Policy [12]	88.9	88.9	100	88.3	93.3	100
AMPLIFY [14]	71.1	72.2	40	94.2	99.2	200
GHOST [15]	96.7	96.7	100	90.0	95.0	200

Method	Insertion			Sequential Manipulation		
	Prog.@MaxH	Best Prog.	# Human	Prog.@MaxH	Best Prog.	# Human
Phantom [13]	40.8	46.7	100	36.7	36.7	300
Point Policy [12]	45.0	45.0	300	16.7	16.7	300
AMPLIFY [14]	41.7	41.7	100	12.2	15.6	100
GHOST [15]	65.0	65.8	100	71.1	71.1	300

B.3 Object Placement Generalization Experiments

To evaluate whether H2R methods can transfer spatial priors from human demonstrations beyond the robot training distribution, we conduct object placement generalization experiments on the *Stacking* task.

All methods are trained with **100 robot demonstrations** and **100 human demonstrations**. During robot data collection, the manipulated object (the *green bowl*) is initialized only on the **right side** of the table. In contrast, human demonstrations are collected with object placements spanning the **full tabletop workspace**, covering a broader spatial distribution.

At evaluation time, we intentionally place the object on the **left side** of the table, outside the robot training distribution. This setup isolates each method’s ability to leverage human demonstrations for spatial generalization, measuring whether the policy can transfer object-placement priors observed in human videos to unseen robot initializations.

We evaluate four H2R transfer methods: Phantom [13], AMPLIFY [14], Point Policy [12], and GHOST [15]. We observe that Point Policy achieves the strongest overall performance, obtaining the highest progression score and the highest success rates across later task stages. GHOST achieves the second-best performance, while Phantom and AMPLIFY exhibit substantially larger performance degradation under the distribution shift.

One possible explanation for Point Policy’s strong generalization ability is its use of a unified action space for both human and robot demonstrations. By representing demonstrations from both embodiments within the same action space, Point Policy can directly leverage human demonstrations without requiring embodiment-specific action mappings. This unified formulation facilitates knowledge transfer from human demonstrations and may improve robustness to variations in object placement, scene configuration, and task execution.

In contrast, GHOST benefits from human demonstrations primarily through its high-level subgoal policy, while the low-level goal-conditioned policy is trained exclusively on robot demonstrations. As a result, only part of the policy hierarchy directly benefits from human data. Under substantial object placement shifts, the low-level goal-conditioned policy may remain constrained by the diversity of the robot demonstrations and therefore generalize less effectively than Point Policy.

Table 8: Object placement generalization evaluation under out-of-distribution object initialization on the Stacking task. All methods are trained with 100 robot demonstrations and 100 human demonstrations. Robot demonstrations are collected with object placement restricted to the right side of the table, while evaluation is performed with object placement on the left side. Best per column is **bolded**; second-best is underlined.

Method	Prog.	SR ₁	SR ₂	SR ₃	SR ₄
Phantom	48.3	100.0	73.3	20.0	0.0
AMPLIFY	41.7	100.0	60.0	6.7	0.0
Point Policy	78.3	100	100	66.7	46.7
GHOST	<u>58.3</u>	<u>100</u>	<u>86.7</u>	<u>30</u>	<u>16.7</u>

B.4 Multi-Seed Evaluation

To assess whether our benchmark results are sensitive to training randomness, we additionally evaluate selected experiments across multiple random seeds. For each selected method–task configuration, we repeat policy training with different random initializations while keeping the dataset, hyperparameters, and evaluation protocol fixed.

We report two complementary multi-seed analyses. First, we summarize the final performance under the robot + maximum human demonstration setting, reporting the mean and standard deviation of task progression (**Prog.**) and final-stage success rate (**SR**) across seeds (Table 9 and Figure 9). Second, we plot human-data scaling curves averaged across seeds, with error bands indicating seed-to-seed variation (Figure 8).

These results are intended to verify that the performance trends reported in the main paper are not dominated by random initialization or optimization noise. In particular, consistent method ordering and low variance across seeds would indicate that the observed differences between H2R methods reflect systematic differences in transfer behavior rather than stochastic training effects.

Table 9: Multi-seed benchmark results under robot + maximum human demonstration training. We report mean and standard deviation across random seeds for task progression (**Prog.**) and final-stage success rate (**SR**).

Method	Pick-and-Place		Stacking		Insertion		Sequential Manipulation	
	Prog.	SR	Prog.	SR	Prog.	SR	Prog.	SR
Phantom	77.0 ± 1.4	56.7 ± 2.7	75.6 ± 2.7	25.6 ± 6.8	39.2 ± 3.6	7.8 ± 4.2	37.8 ± 2.4	13.3 ± 4.7
AMPLIFY	74.4 ± 2.7	58.9 ± 4.2	85.3 ± 3.7	61.1 ± 9.6	35.0 ± 6.2	2.2 ± 1.6	8.1 ± 3.4	0.0 ± 0.0
Point Policy	82.6 ± 7.4	62.2 ± 11.7	79.7 ± 7.7	61.1 ± 17.1	37.8 ± 6.5	6.7 ± 11.5	17.4 ± 5.6	0.0 ± 0.0
GHOST	91.1 ± 4.9	82.2 ± 6.9	88.9 ± 5.9	76.7 ± 8.8	64.7 ± 1.3	34.5 ± 6.9	63.3 ± 8.4	38.9 ± 15.0

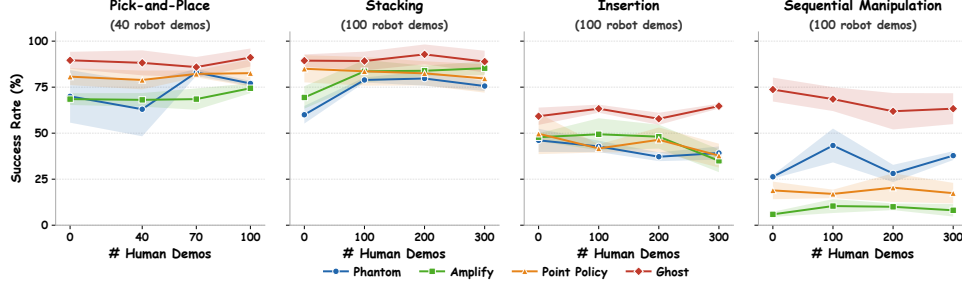


Figure 8: Multi-seed human-data scaling performance. Curves show average performance across random seeds as the number of human demonstrations increases, with shaded regions indicating seed-to-seed variation.

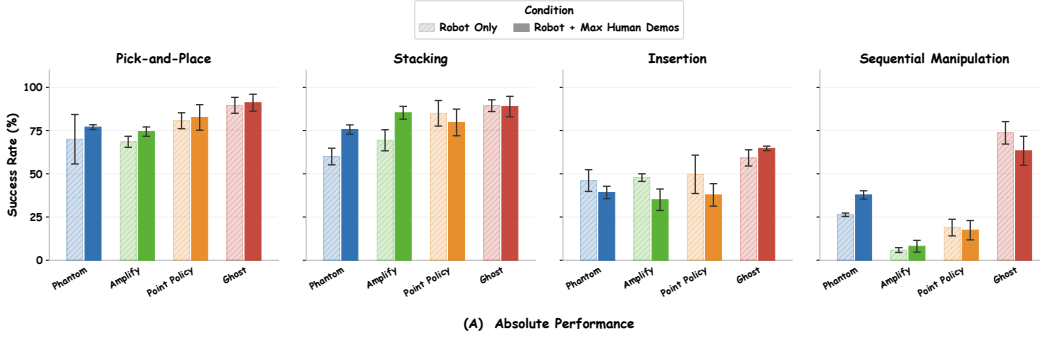


Figure 9: Multi-seed absolute performance comparison across H2R methods and tasks. Hatched bars indicate robot-only training; solid bars indicate robot + maximum human demonstrations. Error bars show standard deviation across random seeds.

B.5 XSkill (Additional H2R Method Trial)

We additionally evaluate XSkill [9], which trains a skill encoder, a diffusion policy conditioned on the inferred skill representations and a skill alignment transformer. We utilized the left-view observation for skill discovery and all three camera views for skill transferring and composing. Our experiments reveal two failure modes across tasks.

During training, the skill encoder frequently exhibits representation collapse, especially for simpler tasks such as pick-and-place. This degenerate state is validated by the observation that all encoded skill prototypes collapse to a uniform probability distribution, assigning the same score to each prototype at any given timestamp. An analysis of the training loss reveals the mechanics behind this convergence. Specifically, the total loss, defined as

$$\mathcal{L} = \lambda_{\text{prototype}} \mathcal{L}_{\text{prototype}} + \lambda_{\text{tcn}} \mathcal{L}_{\text{tcn}}$$

stabilizes at a value of

$$\mathcal{L} = 0.5 \log K + \log \frac{\text{neg} + \text{pos}}{\text{pos}} \approx 4.57$$

when $\lambda_{\text{prototype}} = 0.5$, $\lambda_{\text{tcn}} = 1$, $K = 32$ prototypes are used, $\text{neg} = 16$ negative clips and $\text{pos} = 1$ positive clip are sampled (utilizing the real-world hyperparameters settings reported in the original paper). This stabilization point corresponds to the theoretical minimum of the loss under a fully collapsed representation. When the encoder maps all video clips to a singular point in the embedding space, the Sinkhorn-Knopp algorithm assigns uniform target probabilities $\frac{1}{K}$ to each prototype. Consequently, the predicted probabilities also become uniform at $\frac{1}{K}$, yielding $\mathcal{L}_{\text{prototype}} = \log K$, while the time-contrastive loss $\mathcal{L}_{\text{tcn}}$ simultaneously reduces to $\log \frac{\text{neg} + \text{pos}}{\text{pos}}$. At this theoretical minimum, the gradients vanish and training stalls. Without meaningful variance in the encoded skill representations, the skill-conditioned diffusion policy subsequently degrades into an unconditioned

diffusion policy. We hypothesize that this representation collapse is because simpler tasks feature fewer distinct manipulation phases and therefore lack the visually separable states necessary to drive prototype specialization.

For more complex tasks, the skill encoder does converge, but performance drops substantially compared to simply training an unconditioned diffusion policy using robot demonstrations, as shown in Tab. 10. To probe whether the converged encoder captures semantically meaningful skills, we recorded a human demonstration of the sequential manipulation task with a reordered picking sequence (original: blue $\rightarrow$ orange $\rightarrow$ yellow; new: orange $\rightarrow$ blue $\rightarrow$ yellow) and used it as the prompt video at inference time. Across 10 rollouts, the robot consistently reached for the blue toy first without hesitation, indicating that it ignored the prompt entirely. In Fig. 10, We also visualize the encoded prototype scores for random selected human and robot demonstration in the sequential manipulation task ; no cross-embodiment alignment is apparent. We hypothesize that the skill encoder fails to learn semantically grounded skill representations, and instead captures low-level visual statistics or temporal progress that happen to be stable under augmentation. The uninformative skill representations introduce noise to the diffusion policy and degrade policy’s performance relative to the unconditioned baseline.

We identify the primary failure mode as insufficient diversity in skill ordering across demonstrations. XSkill’s cross-embodiment alignment relies on the same skills appearing in varying sequential contexts so that the encoder is pressured to represent skill identity. In our tasks, however, all human and robot demonstrations follow a fixed object-picking order, even for sequential manipulation task. This contrasts with the original XSkill real-world experiments, where the three kitchen sub-tasks (Draw, Light, Oven) are executed in different orders across demonstrations, providing the combinatorial diversity necessary to disentangle skill identity from temporal context.

Table 10: Simulation results on three additional tasks. Diffusion Policy is trained with 100 robot demonstrations. XSkill is trained with 100 human demonstrations and 100 robot demonstrations.

Method	Stacking		Insertion		Sequential Manipulation	
	Prog.	SR	Prog.	SR	Prog.	SR
Diffusion Policy [38]	95.0	80.0	41.7	0.0	82.2	63.3
XSkill [9]	28.3	0.0	10.0	0.0	10.0	0.0

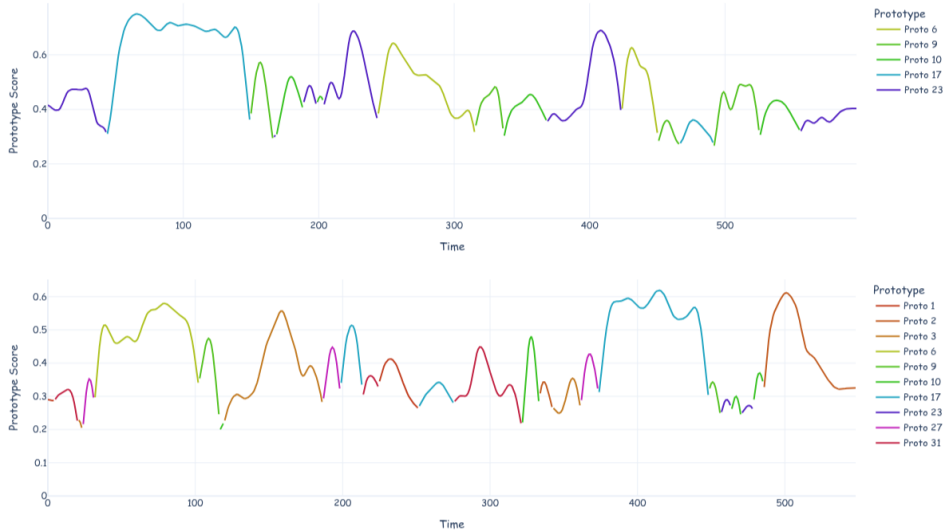


Figure 10: Encoded prototype scores of random selected demonstration from sequential manipulation task. Different colors represent different prototypes. For visual clarity, only the prototype with the highest score at each time step is plotted. **Up**: human demo. **Down**: robot demo.

C Failure Cases and Analyses

C.1 Method-Specific Analysis

C.1.1 Phantom Failure Cases

As shown in the main benchmark results, Phantom performs competitively on coarse manipulation tasks but remains more challenging on tasks requiring precise contact-rich interaction such as stacking and insertion. To better understand this behavior, we analyze the generated robot-inpainted demonstrations used as Phantom’s training supervision.

Figure 11 visualizes representative robot-inpainting results across the four H2RBench tasks. Phantom bridges the embodiment gap by retargeting human hand motion into robot end-effector trajectories and rendering corresponding robot demonstrations through inverse kinematics. While this produces visually plausible robot behavior, we observe two recurring sources of error.

First, the generated robot images occasionally exhibit visual artifacts or imperfect alignment between the rendered robot and the scene, introducing noise in the visual supervision signal.

More importantly, the retargeted robot actions are often not geometrically precise with respect to the manipulated objects. Across tasks, the generated end-effector trajectory can deviate slightly from the intended grasp pose, contact location, or placement configuration. These errors are typically small at individual timesteps but become particularly noticeable in precise manipulation settings where task success depends strongly on accurate object contact and relative pose.

Together, these observations suggest that Phantom’s limitations arise not only from image-generation quality, but also from inaccuracies in human-to-robot motion retargeting. Since the policy is trained on these generated robot demonstrations, imprecision in the retargeted actions can directly propagate into the learned behavior, making fine-grained manipulation tasks more difficult to transfer reliably.

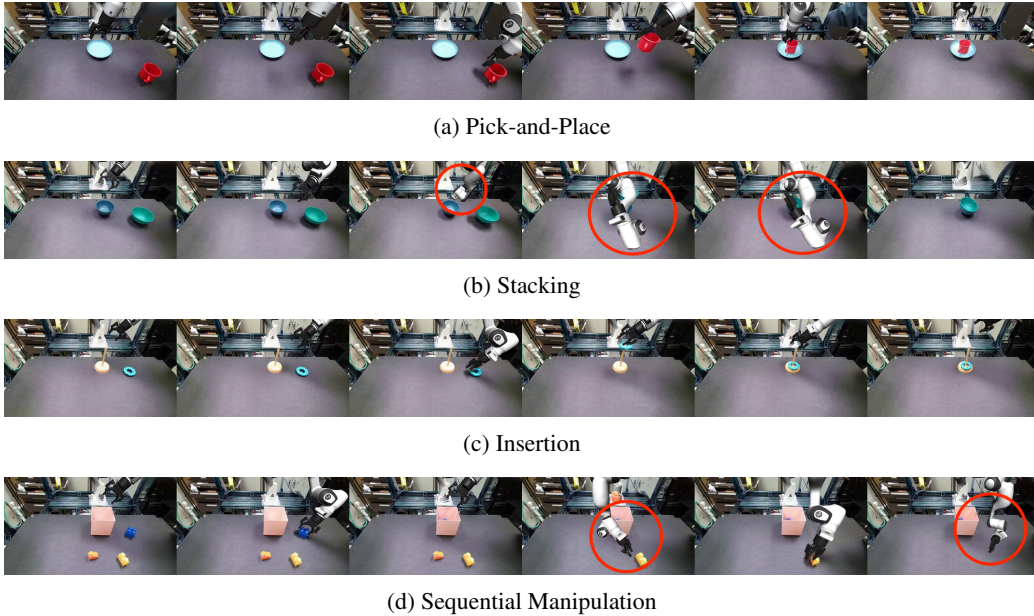


Figure 11: Representative robot-inpainting results generated by Phantom [13] across the four H2RBench tasks.

C.1.2 Amplify Failure Cases

To better understand the task-dependent performance of AMPLIFY [14], we further analyze the learned motion latent space using two complementary metrics: (1) the 2-Wasserstein (Fréchet) distance between human and robot latent distributions, and (2) t-SNE visualization of latent embeddings.

Following recent analysis in [39], we compute the 2-Wasserstein distance (W_2) between human and robot latent features using a Gaussian approximation in PCA-reduced latent space. This metric captures both centroid shift and covariance mismatch between the two distributions, and provides a quantitative measure of cross-embodiment alignment.

Table 11 summarizes the W_2 distance across tasks under different human demonstration scales. We observe that **Stacking**, the task where AMPLIFY achieves the strongest performance improvement in simulation, exhibits a monotonic decrease in W_2 distance as more human demonstrations are added. This suggests that the motion tokenizer learns increasingly aligned latent representations between human and robot trajectories, improving cross-embodiment transfer.

In contrast, **Insertion** shows a monotonic increase in W_2 distance with additional human demonstrations, indicating growing distributional mismatch between human and robot motions. We hypothesize that this is caused by the fundamentally different interaction dynamics between human fingertip insertion and robot gripper-based peg alignment, which makes latent alignment more difficult and limits transfer performance.

Figure 12 further visualizes the latent embeddings for human and robot demonstrations using t-SNE. Across all four tasks, human and robot trajectories form partially separated clusters with limited overlap. This suggests that AMPLIFY does not fully learn a shared cross-embodiment latent skill space. Instead, the learned representation remains partially domain-specific, which may explain why performance gains from human demonstrations are task-dependent and why transfer remains challenging in some manipulation settings.

Together, these results suggest that latent distribution alignment is closely correlated with AMPLIFY’s H2R transfer performance: tasks with stronger human–robot latent alignment tend to benefit more from human demonstrations, while tasks with persistent distributional separation remain more difficult to transfer.

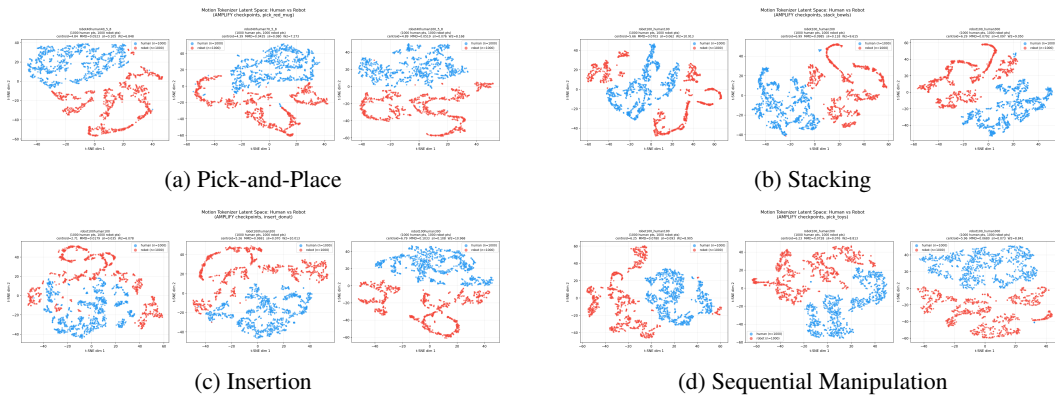


Figure 12: t-SNE visualization of AMPLIFY motion latent embeddings for human and robot demonstrations across four H2RBench tasks. Human and robot trajectories form partially separated clusters with limited overlap, indicating incomplete alignment of the learned cross-embodiment latent representation. The degree of overlap varies across tasks and correlates with downstream H2R transfer performance.

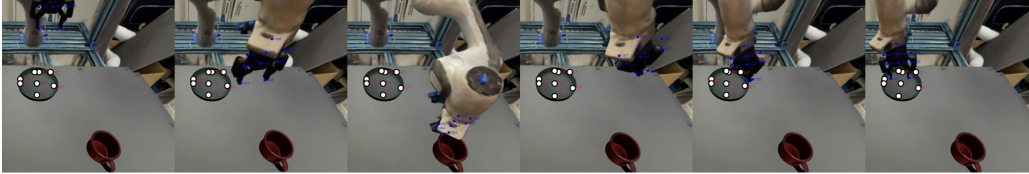
Table 11: 2-Wasserstein (W_2) distance between human and robot latent feature distributions for AMPLIFY under different human demonstration scales. Lower values indicate stronger alignment between human and robot latent representations.

Task	Low Human	Mid Human	High Human
Stacking	10.91	9.62	9.05
Pick-and-Place	6.85	7.27	8.17
Insertion	6.08	10.01	10.97
Sequential Manipulation	8.91	8.81	8.84

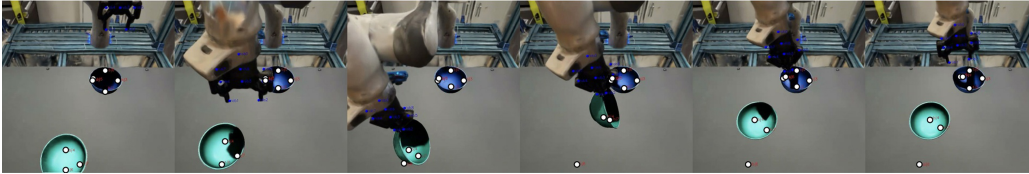
C.1.3 Point Policy Failure Cases

As shown in the main benchmark results, Point Policy [12] performs well across a variety of manipulation tasks, but its performance drops significantly on long-horizon pick-and-place tasks. To better understand this limitation, we analyze the visual correspondence and tracking pipeline used to construct the policy state representation.

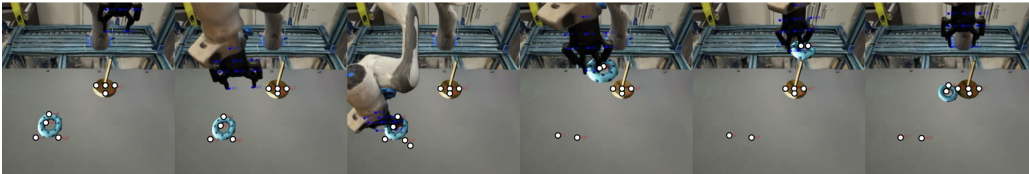
Point Policy uses DIFT [40] to transfer task-relevant object keypoints from a manually annotated reference image to the first frame of an observed trajectory. These keypoints are then used to initialize Co-Tracker [41], which tracks them throughout the remainder of the trajectory. The resulting point trajectories serve as the primary state representation provided to the policy.



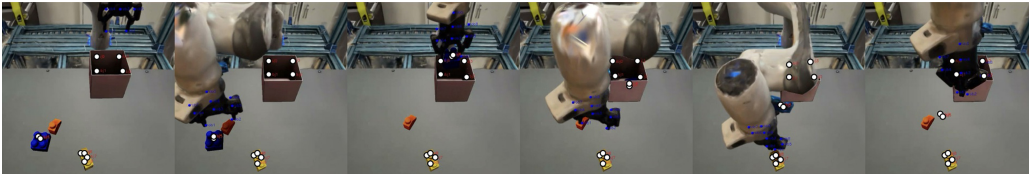
(a) Pick-and-Place



(b) Stacking



(c) Insertion



(d) Sequential Manipulation

Figure 13: Representative keypoint tracking visualizations of Point Policy [12] during evaluation rollouts across the four H2RBench tasks. All shown rollouts are failure cases. White markers denote task-relevant object keypoints, while blue markers denote robot keypoints.

Figure 13 shows representative tracked keypoints during evaluation rollouts. We observe two common failure modes. First, DIFT can occasionally produce incorrect correspondences due to appearance changes, lighting variations, occlusions, or visually ambiguous object regions. Since these correspondences determine the initial points used by Co-Tracker, errors at this stage can cause the tracker to follow semantically incorrect regions throughout the trajectory. Second, Co-Tracker can struggle when tracked points become partially or fully occluded during task execution. This issue frequently arises in scenarios involving object handoffs, container interactions, or sequential pick-and-place operations, where task-relevant objects may become temporarily or permanently invisible to the camera. In these cases, tracked points can drift, become unstable, or disappear altogether.

These tracking failures are particularly problematic in long-horizon pick-and-place tasks. After an object is placed into a container, it often becomes fully occluded from the camera’s view while Co-Tracker continues tracking its associated keypoints. Since the object is no longer visible, the tracker may drift to unrelated image regions and produce meaningless trajectories. These erroneous trajectories remain part of the policy state representation, introducing noise that accumulates over time. As more objects are manipulated and become occluded, the state representation becomes increasingly unreliable, making it harder for the policy to reason about the remaining objects and execute subsequent placements. This failure mode helps explain the near-zero success rates observed when placing later objects, even when the first placement is completed successfully.

Although the policy is trained on demonstrations that naturally contain imperfect correspondences and tracking noise, its performance still depends heavily on the quality of the resulting visual state representation. Errors from either DIFT or Co-Tracker directly affect the point trajectories provided to the policy and can therefore propagate to downstream decisions. This sensitivity is especially evident in tasks requiring precise object contact, accurate placement, or fine-grained spatial reasoning, where even small perception errors can lead to cascading failures.

Overall, these results suggest that Point Policy is highly dependent on the reliability of the underlying correspondence and tracking modules. Improving correspondence quality, handling object occlusions more effectively, and developing more robust long-horizon tracking methods remain important directions for future work.

C.1.4 GHOST Failure Cases

As shown in the main benchmark results, GHOST [15] performs competitively across a range of manipulation tasks and demonstrates strong task-level reasoning capabilities. Nevertheless, failures still occur, particularly in tasks requiring precise object contact, alignment, or placement. To better understand these limitations, we analyze two common sources of error in the GHOST pipeline.

First, many failures appear to originate from limitations in the low-level goal-conditioned policy rather than the high-level goal prediction module. During evaluation, we frequently observe cases where the predicted high-level subgoals are semantically correct and correspond to the appropriate object interaction strategy. However, the low-level policy occasionally predicts actions that follow the intended subgoal but are not sufficiently precise to successfully complete the manipulation, resulting in small spatial deviations in the executed end-effector motion. While such deviations are often inconsequential for coarse manipulation tasks, they become increasingly problematic in tasks requiring precise grasping, insertion, alignment, or object placement. As shown in Figure 14, in those failed rollouts, the robot often approaches the correct target and follows the predicted high-level goal, yet ultimately fails because the end-effector misses the intended contact location by a small margin. These observations suggest that the primary bottleneck of GHOST in challenging manipulation scenarios is the precision of the low-level goal-conditioned policy rather than high-level subgoal prediction.

A second source of error stems from imperfect 3D hand pose estimation in human demonstrations. Similar to prior H2R methods that rely on vision-based hand tracking, GHOST depends on estimated 3D hand states to infer human-object interactions and extract task-relevant supervision. In practice, 3D hand pose estimation is inherently noisy and can become unreliable under occlusion, motion blur,

self-contact, or challenging viewpoints, leading to inaccurate estimates of hand configuration and object interaction. Since the high-level supervision used by GHOST is derived directly from these estimated hand trajectories, errors in hand pose estimation can propagate into the extracted 3D goals and subgoals used for training. Consequently, inaccuracies in hand tracking may degrade the quality of supervision available to the high-level goal prediction module and contribute to downstream task failures.

In addition, GHOST extracts subgoals using heuristic rules based on estimated gripper open and close events. While this strategy is effective across all tasks evaluated in H2RBench, such heuristics may not always capture the underlying task structure and remain sensitive to errors in hand-state estimation. This limitation is particularly relevant for tasks involving prolonged object contact or continuous manipulation, where meaningful task transitions may not coincide with discrete grasping events. Consequently, inaccuracies in hand-state estimation or subgoal extraction can occasionally lead to suboptimal supervision and contribute to downstream execution failures.

Therefore, these observations suggest that GHOST’s remaining failures are primarily associated with limitations in the low-level goal-conditioned policy and imperfect 3D hand pose estimation. Addressing these limitations through more accurate low-level goal-conditioned policies, improved 3D hand pose estimation methods, and more robust goal and subgoal extraction techniques represents a promising direction for future work.



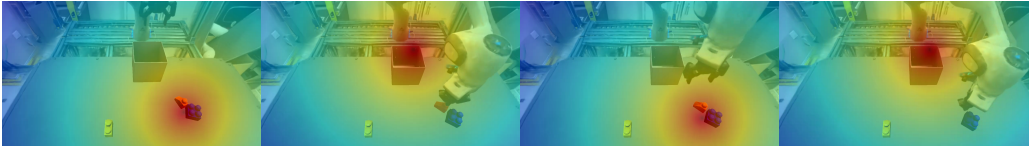
(a) Pick-and-Place



(b) Stacking



(c) Insertion



(d) Sequential Manipulation

Figure 14: Representative high-level prediction visualizations of GHOST [15] during evaluation rollouts across the four H2RBench tasks. The heatmaps are generated by projecting the high-level 3D goal prediction onto the image plane. All shown rollouts correspond to failure cases. Additional rollout videos are available on the project website.

D Real-World Experiments

To evaluate whether H2RBench simulation performance is predictive of real-world deployment, we benchmark all evaluated H2R methods on the physical robot under both robot-only and maximum-human-demonstration training settings. In addition to the main-paper results, we provide further analysis of Sim-vs.-Real correspondence from three complementary perspectives: overall benchmark-level correlation, per-task consistency, and sensitivity to individual methods and tasks.

D.1 Overall Sim-vs.-Real Performance

Table 12 summarizes overall Sim-vs.-Real fidelity across all four tasks and four evaluated H2R methods. Pearson r and Spearman ρ are computed over all method-task pairs, while MMRV is computed per task and averaged across tasks.

Overall, simulation performance remains strongly predictive of real-world performance under both training settings. In the robot-only setting, we observe near-perfect linear correlation ($r = 0.971$) together with strong rank consistency ($\rho = 0.940$). Under maximum human supervision, correlation remains high ($r = 0.894$; $\rho = 0.851$), although with slightly larger variation in relative method ordering.

Figure 15 visualizes this relationship across all method-task pairs. Across both settings, higher simulated success rates generally correspond to higher real-world success rates, indicating that H2RBench captures the relative performance trends of H2R methods reliably before deployment on physical hardware.

Table 12: Overall Sim-vs.-Real fidelity metrics across four tasks and four H2R methods. Pearson r and Spearman ρ are computed over all method-task pairs ($N = 16$). MMRV is computed per task using normalized success rates in $[0, 1]$ and averaged across tasks. Lower MMRV indicates stronger agreement in relative method ordering.

Condition	Pearson $r \uparrow$	Spearman $\rho \uparrow$	MMRV $\downarrow$
Robot Only	0.971	0.940	0.014
Robot + Max Human	0.894	0.851	0.060

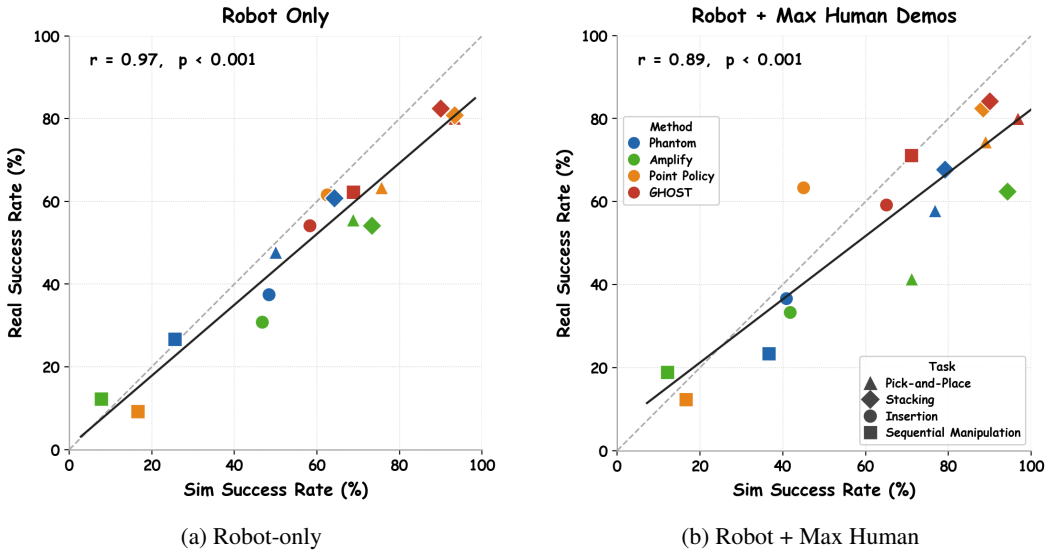


Figure 15: Sim-vs.-Real performance correlation under robot-only (left) and robot + maximum human demonstration (right) settings. Each point corresponds to one method-task pair. Simulated performance is broadly predictive of real-world performance across both training conditions.

D.2 Per-Task Sim vs. Real Correlation

Table 13: Per-task Mean Maximum Rank Violation (MMRV) under robot-only and maximum-human-demonstration settings.

Task	Robot Only ↓	Robot + Max Human ↓
Pick-and-Place	0.000	0.000
Stacking	0.042	0.171
Insertion	0.000	0.037
Sequential Manipulation	0.015	0.033
Mean	0.014	0.060

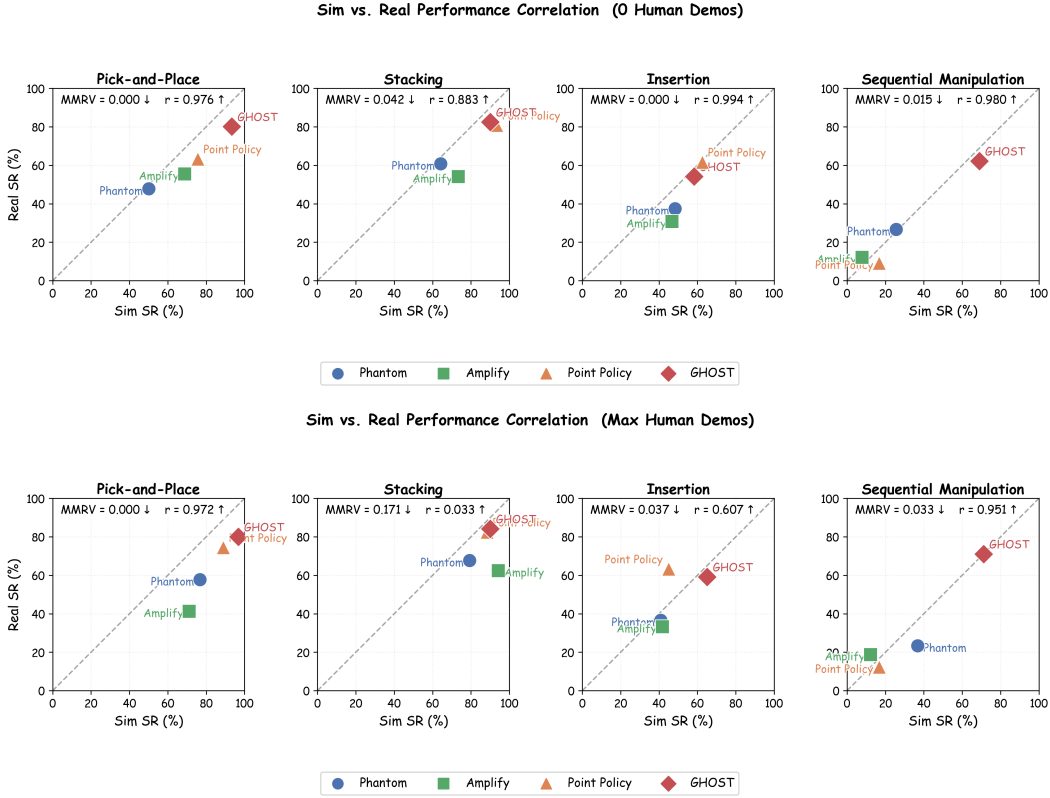


Figure 16: Per-task Sim-vs.-Real performance correlation across H2R methods. Each panel shows the relationship between simulated and real-world success rates for one task. The top row corresponds to robot-only training (0 human demos), and the bottom row corresponds to training with the maximum number of human demonstrations. Each point denotes one method, with Pearson r and MMRV reported per task.

To better understand where Sim-vs.-Real discrepancies arise, we further analyze performance correlation at the task level.

Table 13 reports per-task Mean Maximum Rank Violation (MMRV), while Figure 16 visualizes the corresponding Sim-vs.-Real success-rate relationship across methods.

We find that Sim-vs.-Real agreement is strongest for pick-and-place, which achieves zero MMRV under both training settings. Sequential manipulation and insertion also maintain relatively low ranking violation, indicating that method ordering is largely preserved between simulation and real-world evaluation.

The largest deviation occurs in the stacking task, particularly under maximum human supervision. Compared with other tasks, stacking requires more precise object alignment and stable contact interaction, making success rates more sensitive to small differences in object geometry, contact dynamics, and execution behavior between simulation and the physical robot.

We additionally observe a larger Sim-vs.-Real performance shift for AMPLIFY than for other methods. Because AMPLIFY predicts dense motion tracks as an intermediate representation, its performance depends strongly on the temporal smoothness and consistency of robot trajectories. Compared with planner-generated trajectories in simulation, real-world teleoperated demonstrations contain greater motion variability, which can affect motion-track prediction quality and lead to reduced downstream policy performance.

Importantly, these performance shifts do not always result in changes in relative method ordering. In several cases, method rankings remain stable despite moderate changes in absolute success rate. This suggests that remaining Sim-vs.-Real gaps are often method-dependent and primarily reflect differences in representation robustness to real-world execution variation rather than limitations of the benchmark itself.

D.3 Ablation on Sim vs. Real Correlation

To further characterize the source of remaining Sim-vs.-Real discrepancy, we perform a sensitivity analysis by excluding individual methods and tasks from the evaluation. Table 14 summarizes Sim-vs.-Real metrics under different subsets.

Excluding AMPLIFY consistently improves Pearson correlation, Spearman correlation, and MMRV under both training settings, with the largest improvement observed under maximum human supervision. This is consistent with the per-task analysis and supports the observation that AMPLIFY exhibits a larger Sim-vs.-Real performance shift than other evaluated methods.

Excluding the stacking task also reduces MMRV, particularly in the robot+human setting. This further confirms that stacking contributes disproportionately to ranking inconsistencies across domains.

Overall, these ablation results indicate that observed Sim-vs.-Real discrepancies are concentrated in a small number of method–task interactions rather than uniformly distributed across H2RBench. Despite these localized differences, the benchmark maintains strong overall predictive value for comparative H2R evaluation before real-world deployment.

Table 14: Sensitivity analysis of Sim-vs.-Real metrics under different subsets of methods and tasks.

Condition	Subset	N	Pearson $r \uparrow$	Spearman $\rho \uparrow$	MMRV $\downarrow$
Robot Only	Full Benchmark	16	0.971	0.940	0.014
	Excluding AMPLIFY	12	0.984	0.970	0.003
	Excluding Stack Bowls	12	0.971	0.986	0.005
Robot + Max Human	Full Benchmark	16	0.894	0.851	0.060
	Excluding AMPLIFY	12	0.924	0.902	0.007
	Excluding Stack Bowls	12	0.887	0.890	0.023

D.4 Metric Calculation

To quantify Sim-vs.-Real correspondence, we report Pearson correlation, Spearman rank correlation, and Mean Maximum Rank Violation (MMRV).

Pearson Correlation. Pearson correlation measures the linear relationship between simulated and real-world success rates across method–task pairs:

$$r = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^N (y_i - \bar{y})^2}}, \quad (1)$$

where x_i and y_i denote the simulated and real-world success rates for experiment i , and $\bar{x}$ and $\bar{y}$ are the corresponding means.

Spearman Rank Correlation. Spearman correlation evaluates monotonic agreement between simulated and real-world performance rankings:

$$\rho = 1 - \frac{6 \sum_{i=1}^N d_i^2}{N(N^2 - 1)}, \quad (2)$$

where d_i is the difference between the rank of experiment i in simulation and in the real world.

For both Pearson r and Spearman ρ , we compute the correlation over all $N = 16$ method–task pairs within each evaluation condition (robot-only or robot + max human demos). We use pooled computation across tasks rather than per-task correlation because each individual task contains only four methods, which leads to unstable rank-based statistics with limited sample size. Aggregating across tasks provides a more reliable estimate of overall Sim-vs.-Real correspondence while capturing whether higher-performing policies in simulation also tend to perform better in the real world.

Mean Maximum Rank Violation (MMRV). To quantify ranking inconsistency at the task level, we additionally report Mean Maximum Rank Violation (MMRV), following SIMPLER [19]:

$$\text{MMRV} = \frac{1}{T} \sum_{t=1}^T \frac{1}{N} \sum_{i=1}^N \max_{j \neq i} \left[|r_i^{(t)} - r_j^{(t)}| \cdot \mathbf{1} \left((s_i^{(t)} < s_j^{(t)}) \neq (r_i^{(t)} < r_j^{(t)}) \right) \right] \quad (3)$$

where $s_i^{(t)}$ and $r_i^{(t)}$ denote the simulated and real-world success rates of method i on task t , respectively, normalized to $[0, 1]$. N denotes the number of methods, T the number of tasks, and $\mathbf{1}[\cdot]$ is the indicator function, which equals 1 when the relative ordering between two methods differs between simulation and real-world evaluation, and 0 otherwise.

We compute MMRV independently for each task and report the average across all T tasks. Lower MMRV indicates stronger agreement in relative method ordering between simulation and the real world.