GenH2R: Learning Generalizable Human-to-Robot Handover via Scalable Simulation, Demonstration, and Imitation

Zifan Wang*[1,3] Junyu Chen*[1,3] Ziqing Chen[1] Pengwei Xie[1] Rui Chen[1] Li Yi†[1,2,3]
[1]Tsinghua University [2]Shanghai Artificial Intelligence Laboratory [3]Shanghai Qi Zhi Institute
https://GenH2R.github.io

Abstract

This paper presents GenH2R, a framework for learning generalizable vision-based human-to-robot (H2R) handover skills. The goal is to equip robots with the ability to reliably receive objects with unseen geometry handed over by humans in various complex trajectories. We acquire such generalizability by learning H2R handover at scale with a comprehensive solution including procedural simulation assets creation, automated demonstration generation, and effective imitation learning. We leverage large-scale 3D model repositories, dexterous grasp generation methods, and curve-based 3D animation to create an H2R handover simulation environment named GenH2R-Sim, surpassing the number of scenes in existing simulators by three orders of magnitude. We further introduce a distillation-friendly demonstration generation method that automatically generates a million high-quality demonstrations suitable for learning. Finally, we present a 4D imitation learning method augmented by a future forecasting objective to distill demonstrations into a visuo-motor handover policy. Experimental evaluations in both simulators and the real world demonstrate significant improvements (at least +10% success rate) over baselines in all cases.

1. Introduction

The embodied AI research community has long been driven by the goal of empowering robots to interact and collaborate with humans. A crucial aspect of this pursuit is equipping robots with the capability to reliably receive arbitrarily moving objects of varying geometry handed over by humans, based on dynamic visual observations. This human-to-robot (H2R) handover ability allows robots to seamlessly collaborate with humans across a wide range of tasks, including cooking, room tidying, and furniture assembly.

However, compared to learning human-free robot manipulation skills, the progress in scalably learning H2R handover that can generalize to various objects and versatile human behaviors has lagged due to its unique challenges. Training robots to interact with humans in real-world scenarios entails increased risks and expenses, rendering it inherently non-scalable. Therefore, it is demanded to simulate human behaviors and train robots in simulated environments prior to real-world deployment. However, creating a substantial number of assets for humans handing over objects poses a significant challenge. In a recent study [9] that employed motion capturing to drive virtual humans in a simulator, only 1000 unique human hand motion trajectories were provided for handing over 20 objects. Limited object geometry and human motion assets can hardly capture the complexities of the real world. Besides, the challenge extends to the demonstration side. The success of large language model [6, 31, 49] has suggested a recipe for scaling up learning through modeling large-scale training

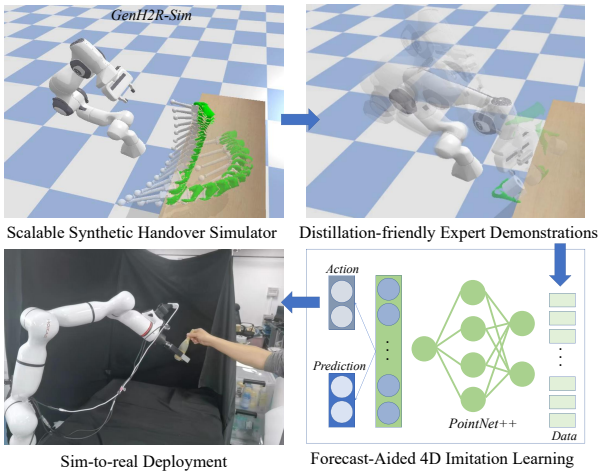

Figure 1. **The overview of GenH2R.** We introduce a framework for learning generalizable vision-based human-to-robot handover via scalable synthetic simulation, distillation-friendly expert demonstration generation, and a forecast-aided 4D imitation learning method. Our models demonstrate strong generalization capabilities to real datasets and can be deployed to a real robot.

*Equal contribution with the order determined by rolling dice.
†Corresponding author.

data. Nevertheless, collecting robot demonstrations receiving objects from real humans is very costly and unscalable. How to scale up the number of demonstrations while ensuring effective learning poses additional challenges.

In this work, we aim to learn generalizable H2R handover at scale by tackling the above challenges. We present a comprehensive solution that scales up both the assets and demonstrations and effectively learns a closed-loop visuomotor policy through a novel imitation learning algorithm.

Specifically, to scale up geometry and motion assets depicting humans handing over various objects, we leverage large-scale 3D model repositories [7, 15], dexterous grasp generation methods [44], and curve-based 3D animation. This enables us to procedurally generate millions of handover scenes, forming an environment named GenH2R-Sim to support generalizable H2R handover learning. GenH2R-Sim surpasses HandoverSim [9], an existing H2R simulator, in both scene quantity (by three orders of magnitude) and unique object involvement (by two orders of magnitude). In addition, scenes in GenH2R-Sim go beyond a straightforward giving and then receiving and cover cases when humans might keep transforming the object in a large range during the entire H2R handover process. This allows for studying complex behaviors such as humans hesitating before handing over.

To scale up robot demonstrations, we draw inspiration from the Task and Motion Planning (TAMP) [21] literature and propose to automatically generate demonstrations with grasp and motion planning using privileged human motion and object state information. There are some straightforward ways to achieve this goal, such as using the privileged human handover destination information to plan a smooth demonstration. However, the problem is more challenging than it seems since the generated demonstrations need to be suitable for distilling into a visuomotor policy. We identify the vision-action correlation between visual observations and planned actions as the crucial factor influencing distillability and point out that due to the constraints of robot arm morphology one can easily generate observation-irrelevant actions and thus harm distillation. To tackle this challenge, we present a distillation-friendly demonstration generation method that sparsely samples handover animations for landmark states and periodically replans grasp and motion based on privileged future landmarks.

Finally, to distill the above demonstrations into a visuomotor policy, we utilize point cloud input for its richer geometric information and smaller sim-vs-real gap compared to images. We propose a 4D imitation learning method that factors the sequential point cloud observations into geometry and motion parts, facilitating policy learning by better revealing the current scene state. Furthermore, the imitation objective is augmented by a forecasting objective which predicts the future motion of the handover object. Since our demonstrating actions are generated based on future landmarks, the forecasting objective can help further exploit the vision-action correlation.

We evaluate our learned policy in simulators (HandoverSim and our own GenH2R-Sim) and the real world. Remarkably, without any mocap assets or real-world demonstrations, our method achieves significantly better performance compared to baselines across all settings (at least **+10%** success rate). Our experiments highlight that the scaling-up efforts bring substantial improvement in policy generalizability to novel geometry and complex motion. Furthermore, these efforts greatly facilitate skill transfer to real robotic systems.

In summary, the key contribution of this paper is a novel framework scaling up the learning of H2R handover with the following three components: i) a simulation environment named GenH2R-Sim consists of millions of human handover animations for generalizable H2R handover learning, ii) an empirically validated automatic robot demonstration generation pipeline for vision-based closed-loop control, iii) a forecast-aided 4D imitation learning method effective in distilling the large-scale demonstrations.

2. Related Work

2.1. Human-to-Robot Handovers

Recently, significant progress in human-robot handovers [11, 32, 35] has been observed, driven by the increasing popularity of human-robot interaction [1, 37] and the emergence of extensive datasets [5, 8, 17, 24, 27, 47] capturing hand-object interactions. Some traditional methods [2, 4] require 3D object shape models and struggle to handle unseen objects. One possible way for handover is to consider grasping and dynamic motion planning [18, 29, 46, 48]. However, these methods often exhibit constrained motions and perform poorly on large-scale datasets. HandoverSim [9], a physics-simulated environment, introduced a new simulation benchmark for human-to-robot object handovers. Leveraging DexYCB [8], a dataset of human grasping objects and performing handover attempts, this environment allows training learning-based handover policies such as [10]. However, it lacks large-scale and diverse handover scenes, which limits its effectiveness for generalizable handovers. Building on this, we propose GenH2R-Sim , aiming to benchmark generalizable handover.

2.2. Scaling Up Robot Demonstrations

In the realm of robot learning, scaling up data collection to encompass a diverse range of manipulation skills has spurred extensive research. Approaches include leveraging large language models [23], enhancing hardware capabilities [38], utilizing non-robotics datasets [22], and

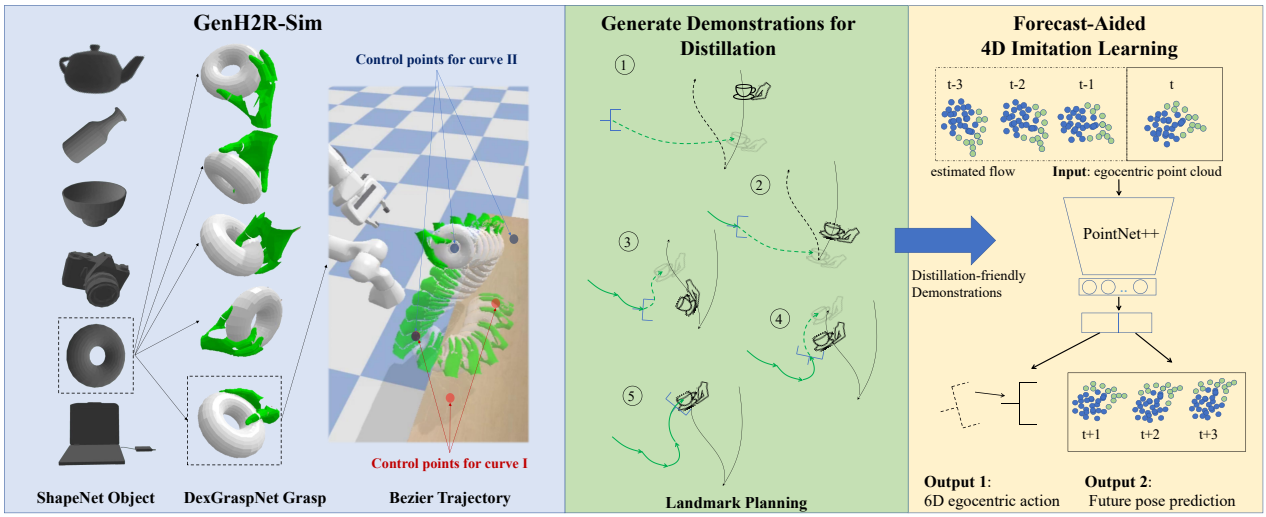

Figure 2. **The overview of our framework.** First, we propose a new simulation environment named GenH2R-Sim, featuring large-scale synthetic datasets with diversity in object geometry, grasp poses, and complex trajectories. Second, other than destination planning (move straight toward the final position) and dense planning (replan at each step), we propose a distillation-friendly demonstration generation method—landmark planning, predicting landmarks on the trajectory (as indicated by the dashed object above) and replanning based on those landmarks. Thirdly, our Forecast-aided 4D Imitation Learning leverages past flow information, and the forecasting objective enhances the exploitation of vision-action correlation.

employing trial-and-error explorations [20]. As depicted in [23], one of the most critical challenges is scaling up robot-complete data. A popular line of research scales up demonstration generation via Task and Motion Planning [12, 21, 30]. These works usually focus on fairly static scenes without active motion while our method extends to dynamic H2R handover cases by considering how to interpret human behavior and generate demonstrations easy to be distilled by closed-loop visuo-motor policy.

2.3. Offline Learning from Demonstrations

Imitation Learning (IL) represents a methodology for training embodied agents in manipulation tasks by utilizing expert demonstrations. The commonly used Behavior Cloning (BC) [33] strategy directly trains the policy to imitate expert actions in a supervised learning manner. Despite its simplicity, this approach has demonstrated remarkable effectiveness in robotic manipulation [3, 19, 28, 50] especially when combined with a substantial number of high-quality demonstrations [14, 25]. Inspired by these works, we adopt an imitation learning paradigm, focusing on how to leverage spatial-temporal perception and future forecasting to better consume our distillation-friendly demonstrations.

3. Method

3.1. Overview

For the generalizable H2R handover task, we introduce GenH2R, a framework designed to learn control policies, specifically 6D control actions for the robot gripper, using segmented point cloud data captured from an egocentric camera. We describe our method for synthesizing human handover animations in Section 3.2, generating expert demonstrations in Section 3.3, and distilling demonstrations to 4D vision-based neural networks by imitation learning in Section 3.4, as the pipeline depicted in Figure 2.

3.2. GenH2R-Sim

The size and quality of human-object datasets in simulators play a crucial role in generating high-quality handover demonstrations and training reliable policies for handover scenarios. The recent handover simulator, HandoverSim [9], utilizes the DexYCB [8] dataset, which captures real-world human grasping objects in a limited manner, comprising only 1000 scenes with 20 distinct objects. In the real world, scenarios can be more complex and may involve intricate trajectories and poses beyond those in DexYCB.

To address these limitations, we introduce a new environment, GenH2R-Sim, to overcome these deficiencies and facilitate generalizable handovers. To diversify geometry and motion assets depicting humans handing over various objects, we focus on two primary aspects: the hand grasping pose and the hand-object moving trajectory within a scene.

In aspects of grasping poses, DexGraspNet [43] has made significant contributions by employing optimization techniques to generate a substantial dataset of human hand grasp poses. We utilize this method to generate approximately 1,000,000 grasp poses for 3,266 different objects sourced from Shapenet [7]. These objects span a wide range of categories, from larger items like computers to smaller

ones like mobile phones, covering most sizes and shapes encountered in real-life handovers.

In aspects of hand-object moving trajectories, we propose to use Bézier curves, which are one class of smooth curves determined by several control points, to generate complex yet smooth-transiting motion trajectories. We use multiple Bézier curves to model different stages of the motion, and link the ends of these curves to create a seamless track. We can generate scenes matching various scenarios of different complexity in the real world by adjusting the distribution of control points of the trajectory and the speed of the human hand. To enhance the trajectory's realism, we incorporate consistent object rotations, which also enhances the importance of choosing the appropriate grasp for the robotic arm. Since we can always attach a new segment of motion at the end of the current motion and the duration is much longer than DexYCB scenes, the destination of the hand-object is not a significant factor, so we just randomly select a point within the reach of the robotic arm.

We do not guarantee that every item in the dataset we generate perfectly mimics the human-like characteristics of real-world data, but our approach ensures a significantly higher degree of domain randomization and provides greater diversity in terms of geometry and motion. Given the challenges in scaling up real-world motion capture datasets, we opt for a large-scale synthetic dataset for our handover simulations. Our key insight is that for both demonstrations and policy learning, having a substantial amount of synthetic data is more beneficial than relying on a small-scale real-world dataset.

GenH2R-Sim follows the setup of HandoverSim, which consists of a Panda 7DoF robotic arm with a gripper and a wrist-mounted RGB-D camera, and a simulated human hand. Just like HandoverSim, we switch from the pre-handover kinematic phase to the handover dynamic phase when the object has been in contact with the gripper. HandoverSim is not adaptive to the robot's action and just loads and replays every frame of the data. To align with the real-world handover process more naturally in GenH2R-Sim, the simulated hand will stop from moving and wait for handover when the robot arm is close to the object.

3.3. Generating Demonstrations for Distillation

In this section, we address a key question in learning visuo-motor policy: how to efficiently generate robot demonstrations that incorporate paired vision-action data from successful task experiences. While distilling successful demonstrations into a single policy has proven effective for open-loop control tasks, the challenge lies in closed-loop visuo-motor control, where the quality of demonstrations becomes crucial for learning. Merely ensuring success is no longer sufficient. We present two examples of demonstration generation with different grasp and motion planning strategies as shown in Figure 3. In the first example, a foresighted planner generates smooth, short demonstrations based on the privileged destination end state of a human handover animation. Though efficient, the planned path does not align actions with the dynamic visual observations during the handover. Distilling such demonstrations requires accurately forecasting the end state of the human trajectory, which can be extremely challenging in complex handover cases. The second example involves a shortsighted planner that independently replans grasp and motion at each time step using privileged hand and object states. Due to robot morphology constraints and the multi-solution nature of common robot planners, smooth visual observations may correspond to unsmooth and multi-modal robot trajectories, increasing the difficulty of distillation. We emphasize the importance of distillability as a quality factor for handover demonstrations. An effective demonstration generation method must consider the vision-action correlation by jointly incorporating robot morphology and dynamic vision during grasp and motion planning.

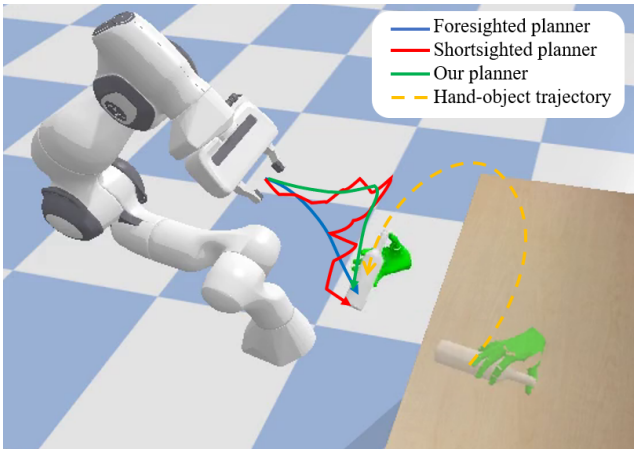

Figure 3. **Different demonstration generation methods for dynamic handover.** The orange curve shows the hand-object trajectory. The blue, red, and green curves show the example trajectories generated by the foresighted planner, the shortsighted planner, and our planner, respectively.

Along this line, we base our method on the foresighted and shortsighted planner mentioned above to combine the advantages of both sides while encouraging the demonstration distillability. We first improve the shortsighted planner so that sequentially smooth visual observations result in smooth grasp and motion plans. Then we improve the handover efficiency by looking toward the future while guaranteeing the vision-action correlation.

To be specific, we build our method based on the OMG planner [40] for grasp and motion planning. This planner optimizes the grasp and motion path by considering the object's 6D pose and a set of candidate grasp poses. To support this optimization, we provide privileged knowledge that in-

cludes the object's 6D pose, candidate grasps generated through physics simulation [15], and human hand poses for filtering out invalid grasps. However, independently calling the OMG planner for each time step may result in unsmooth trajectories, as it is designed for static scenarios. To address this, we sequentially plan the grasp and motion based on the privileged knowledge by: 1) sorting grasps based on their pose distance to the robot end effector and attempting inverse kinematics (IK) starting from the nearest grasp until success; 2) initializing IK based on the robot arm pose from the previous time step; 3) invoking the OMG planner only when IK can be successfully solved. By prioritizing closer grasps, we encourage the object to remain within the field of view of a wrist camera, reducing visually irrelevant actions when the object is not visible. Additionally, enforcing IK smoothness improves the overall trajectory smoothness. As a result, the enhanced vision-action correlation dramatically improves the demonstration quality.

Our approach modifies the OMG planner for dynamic grasp and motion planning. However, densely replanning at each time step leads to inefficient and non-smooth zigzag demonstrations, which does not align with how humans receive objects. Humans anticipate dynamic scene changes before taking action. On the other hand, a highly foresighted planner that directly plans grasp and motion based on the end state of a human handover animation can disrupt the vision-action correlation. To strike a balance between these extremes, we propose an algorithm that sparsely samples handover animations for landmark states and periodically replans grasp and motion based on future landmarks. The key idea is to select landmarks strategically so that the planner only considers visually foreseeable futures. Specifically, let $\xi = (\mathcal{T}_0, \mathcal{T}_1, \cdots, \mathcal{T}_{T-1})$ represent an object trajectory, where $\mathcal{T}_t \in \mathbb{SE}(3)$ denotes the object pose in the t-th frame within the world coordinate system. Based on all the object trajectories in the training set, we train an object pose forecasting network which consumes past and current object poses $(\mathcal{T}_0, \mathcal{T}_1, \cdots, \mathcal{T}_t)$ for each time step t within each trajectory and forecasts the object poses $(\mathcal{T}_{t+1}, \mathcal{T}_{t+2}, \cdots, \mathcal{T}_{t+N})$ in future N steps. By thresholding the forecasting error corresponding to each time step, we identify a set of endpoints where past observations cannot forecast the future very well and partition the complete trajectory ξ to several segments using endpoints $0 = l_0 < l_1 < \cdots < l_k = T$. Within each segment, we assume the ability to predict the future object pose based on historical information. We then denote $P \in \mathbb{N}$ as the hyperparameter determining the replaning period. For each planning frame $t = 0, P, 2P, \cdots$, suppose the next endpoint is l_{i+1}, *i.e.*, $l_i \leq t < l_{i+1}$. Then we will plan based on the object pose at frame $\hat{t} = \min(t + P, l_{i+1})$, which serves as a landmark. Note here planning is based on the future states but avoids bypassing the sharply transitioning points where human motion becomes unpredictable. Also worth mentioning, densely planning is a special case of our method, and landmark planning is a full version.

3.4. Forecast-Aided 4D Imitation Learning

Traditional methods for human-to-robot handover face challenges in gaining insights into dynamic scene perception. Approaches based on motion planning [41] often emphasize robot morphology and lack dynamic vision perception. They struggle to capture long-horizon information, mainly focusing on the current frame and failing to predict the future. Reinforcement Learning methods [10, 42], while powerful, require extensive training and may train unstably across different scenarios. To enhance the vision-action correlation and establish an efficient training paradigm, we introduce our forecast-aided 4D imitation learning approach.

In robot perception, the 4D point cloud serves as the common representation. In the t-th frame, we can define $M_t^i \in \mathbb{SE}(3)$ as the relative object pose between the current frame and the i-th frame in the egocentric view. While frame stacking is a straightforward approach, it struggles to capture both motion and geometry effectively. Inspired by recent 4D learning methods [13, 39], we employ the Iterative Closest Point (ICP) registration algorithm [36] to efficiently compute transformation matrices $\{\hat{M}_t^{t-1}, \hat{M}_t^{t-2}, \cdots, \hat{M}_t^{t-L_1}\}$ between the point cloud in the t-th frame and the point clouds in previous L_1 frames. Applying these transformation matrices to a specific point in the current frame yields its rough coordinates in previous frames. Then we incorporate this flow feature into 3D PointNet++ [34] to encode a global spatial-temporal feature and use Multilayer Perceptron (MLP) to decode it into a 6D egocentric action. The loss function, denoted as $\mathcal{L}_{action}$, is computed as the L1 loss for aligning 3D points on the robot gripper as defined in [26]. We believe some sophisticated 4D backbones [16, 45] are suitable for 4D understanding, but they are often not suitable for robotic tasks that require a fast reference speed. Our method strikes a balance between effectiveness and simplicity.

To enhance the responsiveness of our policy to human motion and extend the vision horizon into the future, we introduce an auxiliary task to predict the future motion $\{M_t^{t+1}, M_t^{t+2}, \cdots, M_t^{t+L_2}\}$ of objects in the next L_2 frames. Using the ground truth object poses from trajectories, we compute the motion prediction loss for the t-th frame:

$$\mathcal{L}_{pred} = \sum_{i=t+1}^{t+L_2} \|\hat{M}_t^i - M_t^i\| \tag{1}$$

In contrast to reinforcement learning, our imitation learning method requires only a few hours of training and achieves great generalizability through large-scale, high-

quality demonstrations. We acquire vision-action pairs and ground truth object states from demonstrations, and then supervise our policy using the loss function $\mathcal{L} = \mathcal{L}_{action} + \lambda\mathcal{L}_{pred}$, where λ serves as a weighting hyper-parameter to balance the losses. This efficient distillation paradigm empowers our policy to naturally approach objects with a forecasting intention and to effectively generalize to a wide range of unseen objects and motions.

4. Experiments

Dataset (1) HandoverSim [9] contains 1000 real-world H2R handover scenes and 20 objects from DexYCB [8]. We evaluate on the "s0" setup which contains 720 training and 144 testing scenes. Each handover motion has a duration of 3 seconds. Following the evaluation of HandoverSim2real [10], we consider "Sequential" and "Simultaneous" settings. In "s0 (Sequential)", the robot is allowed to move when the hand reaches the handover location and remains static. In "s0 (Simultaneous)", the robot is allowed to move from the beginning of the episode. (2) GenH2R-Sim contains 1,000,000 complex synthetic H2R handover scenes and 3266 objects. We evaluate the "t0" setup which contains 1,000,000 training and 3260 testing scenes. Each handover motion has a duration of 8s and will stop when the robot gripper is close to the object. To introduce more real-world handover scenes into GenH2R-Sim for evaluation, we extract and clip the handover point cloud sequence from HOI4D [27], a real-world mocap dataset. This additional setup is referred to as "t1", which only contains 1000 testing scenes for evaluation.

Metrics We adhere to the HandoverSim evaluation protocol. A successful handover involves grasping the object from the human hand and moving it to a designated location. Failure cases involve hand contact, object drop, and timeout ($T_{max} = 13s$). We report the successful rate and the execution time. Given that some policies prioritize success over speed, potentially wasting considerable human time, and others prioritize speed without considering success, we aim to evaluate both success rate and completion efficiency. To achieve this, we introduce AS (Average Success), akin to AP (Average Precision):

$$\text{AS} = \int_0^1 \text{Success}(t)\,\mathrm{d}t \tag{2}$$

where $\text{Success}(t)$ is success rate considering only successful cases within $t \cdot T_{\text{max}}$. This method can better evaluate success-time relations which is more suitable in our handover scenarios.

4.1. Evaluating on Different Benchmarks

Setup We have 2 training sets: small-scale real-world "s0" from HandoverSim and large-scale synthetic "t0" from our GenH2R-Sim. Evaluation is conducted on four testing sets: "s0 (Sequential)"/"s0 (Simultaneous)" from HandoverSim and "t0"/"t1" from our GenH2R-Sim. We conduct experiments on our forecast-aid 4D imitation learning from different demonstration strategies including destination planning, dense planning, and landmark planning. As discussed in Section 3.3, destination planning denotes the foresighted planner, dense planning denotes the improved shortsighted planner and landmark planning is our proposed method.

Baselines We compare our methods with HandoverSim2real*, the state-of-the-art method in HandoverSim. We additionally compare GA-DDPG which is designed for grasping objects, and OMG Planner.

Results on different datasets As depicted in Table 1, our method trained on "t0" outperform all methods trained on "s0" by a large margin. Compared with Handover-Sim2real trained on "s0", our landmark planning method trained on "t0" exhibits 11.34%, 16.90%, 12.26%, and 15.93% increase in the success rate across the four testing sets. Moreover, compared with our landmark planning method trained on "s0", the version trained on "t0" demonstrates notable improvements, achieving success rate increases of 8.79%, 6.48%, 11.80%, and 14.13% increase in the same testing sets. This underscores the importance of having a substantial amount of synthetic data for handover training in simulation, which is more beneficial than only relying on a small-scale real-world dataset. Our GenH2R-Sim, with its large-scale complex human hand behavior, generalizes effectively to real-world scenarios such as "s0" in DexYCB and "t1" in HOI4D.

Results for different methods We can compare our methods with the baseline HandoverSim2real within the same training set in different benchmarks. When trained on "s0", our landmark planning method demonstrates improvements of 2.55%, 10.42%, 0.46%, and 1.8% (13.43%, 53.48%, 1.07%, and 23.60% in our reproduced version) across the 4 test sets. Similarly, When trained on "t0", our landmark planning method gives substantial improvements of 20.78%, 23.15%, 7.72%, and 21.23% (23.02%, 46.76%, 8.12%, and 34.98% in our reproduced version). The last 3 benchmarks ("s0"(simultaneous), "t0", and "t1") closely resemble real-world scenarios. They greatly demonstrate the effectiveness of our pipeline from distillation-friendly demonstrations to forecast-aided 4D imitation learning, which is capable of handling dynamic robot perception in complex handover scenarios. We also show visualizations on different methods in Figure 4 (a)(b).

*Our approach strictly adheres to the simultaneous setting defined in the paper of HandoverSim and HandoverSim2real: the robot moves from the beginning of the handover episode. However, it's noteworthy that HandoverSim2real manually makes their policy hold still in the first 1.5 seconds in the code implementation, deviating from the simultaneous setting definition. To ensure a fair comparison, we reproduce their results in the true simultaneous setting.

		s0 (Sequential)			s0 (Simultaneous)			t0			t1		
		S	T	AS	S	T	AS	S	T	AS	S	T	AS
	OMG Planner† [41]	62.50	8.31	22.5	-	-	-	-	-	-	-	-	-
train on s0	GA-DDPG [42]	50.00	**7.14**	22.5	36.81	**4.66**	23.6	23.59	7.31	10.3	46.7	**5.50**	26.9
	Handover-Sim2real [10]	75.23	7.74	30.4	68.75	6.23	35.8	29.17	6.29	15.0	52.40	7.09	23.8
	Handover-Sim2real* [10]	64.35	7.61	26.7	25.69	5.43	15.0	28.56	4.73	17.9	30.60	5.98	16.5
	Destination Planning	74.31	7.98	28.7	76.16	5.89	41.7	25.68	5.34	15.1	48.4	7.49	20.5
	Dense Planning	74.77	8.14	28.0	75.45	6.06	40.3	27.30	5.49	15.7	52.3	7.44	22.4
	Landmark Planning	77.78	8.15	29.0	79.17	6.06	42.0	29.63	5.22	17.7	54.2	7.41	23.3
train on t0	GA-DDPG [42]	54.76	7.26	24.2	44.68	5.30	26.5	24.05	4.70	15.3	25.50	5.86	14.1
	Handover-Sim2real [10]	65.97	7.18	29.5	62.50	6.04	33.5	33.71	5.91	18.4	47.10	6.35	24.1
	Handover-Sim2real* [10]	63.55	7.58	26.5	38.89	5.29	23.1	33.31	**4.64**	21.4	33.35	5.81	18.4
	Destination Planning	0.93	11.76	0.1	6.48	11.22	0.9	5.96	7.57	2.5	1.60	11.38	0.2
	Dense Planning	81.48	8.52	28.1	84.95	6.32	43.7	38.04	6.06	20.3	57.90	7.23	25.7
	Landmark Planning	**86.57**	7.62	**35.8**	**85.65**	5.38	**50.2**	**41.43**	4.97	**25.6**	**68.33**	6.14	**36.1**

Table 1. **Evaluating on different benchmarks.** We compare our method against baselines from the test set of HandoverSim [9] benchmark ("s0 (sequential)" and "s0 (simultaneous)") and our GenH2R-Sim benchmark ("t0" and "t1"). We use the best-pretrained models from the repositories of GA-DDPG [42] and Handover-Sim2real [10] for evaluation. The results for our method are averaged across 3 random seeds. Note that S means success rate(%). T means time(s). AS means average success(%) as defined in Equation 2. †: This method [41] is evaluated with ground-truth states and cannot handle dynamic handover like "s0 (Simultaneous)", "t0" and "t1".*: We reproduce the results of HandoverSim2real in the true simultaneous setting as detailed in Section 4.1 to make a fair comparison.

Results for different Demonstrations Trained on "s0" which consists of relatively simple trajectories, demonstrations based on destination planning can offer a rudimentary cue for downstream visuo-motor policy. However, when trained on "t0" this strategy may lose focus on the object, leading to a failure in maintaining vision-action correlation and providing minimal gains for vision-friendly learning. There is a significant 73.38% / 69.68% decrease in success rate in the "s0" setting. Additionally, distillation from landmark planning slightly outperforms dense planning in success rate and completes the handover process more quickly in all benchmarks. While dense planning can sustain the success rate to some extent, it slows down the agent and may result in unnatural approaches to objects. To jointly consider the time efficiency and the success rate, we compare the Average Success in methods distilled from these two strategies and find that landmark planning is a more efficient and generalizable approach. For instance, when trained on "t0", landmark planning exhibits significant improvements of 7.7%, 6.5%, 5.3%, and 10.4% across the four testing sets.

4.2. Evaluating on different Dataset Scales

We have proved the crucial role of large-scale datasets in handover generalization in Section 4.1. We can also reveal it by scaling down the usage of "t0" in GenH2R-Sim which contains 1,000,000 training scenes. With 10% data utilization, we observe a 5.93% drop in the success rate on the unseen "t1" test set. This result proves the significance of the dataset scale in our imitation learning method. Thanks to our large-scale data and efficient demonstration generation pipeline, concerns about limited datasets hindering generalization are alleviated.

Methods	S	T	AS
w/o Flow	31.66	**4.64**	20.2
w/o Prediction	39.18	5.03	24.1
w/o Flow & Prediction	37.04	4.88	23.1
Ours	**41.43**	4.97	**25.6**

Table 2. **Ablations on different modules.** "w/o Flow" means do not use flow information in the input. "w/o Prediction" means do not add prediction loss in the output.

4.3. Ablation Study

As shown in Table 2, we prove the effectiveness of our well-designed 4D imitation learning method. The absence of flow information results in a 9.77% decrease (predicting without past information adversely affects the model performance). The absence of the prediction task leads to a 2.25% decrease, and the absence of both components results in a 4.39% decrease. The results demonstrate the model obtains improved performance in leveraging flow information, particularly when tasked with predicting the future object pose. More ablations about our demonstration generation and imitation learning are detailed in the supplementary material.

4.4. Real World Experiments

Sim-to-Real Transfer In addition to simulation, we deploy the models trained in GenH2R-Sim on a real robotic platform. Using point cloud input from the wrist-mounted camera, we employ the output 6D egocentric action to update the end effector's target position. A user study compares our method against Handover-Sim2real [10]. The supplementary material provides further details.

User Study We recruited 6 users to compare our method (based on landmark planning) and Handover-Sim2real

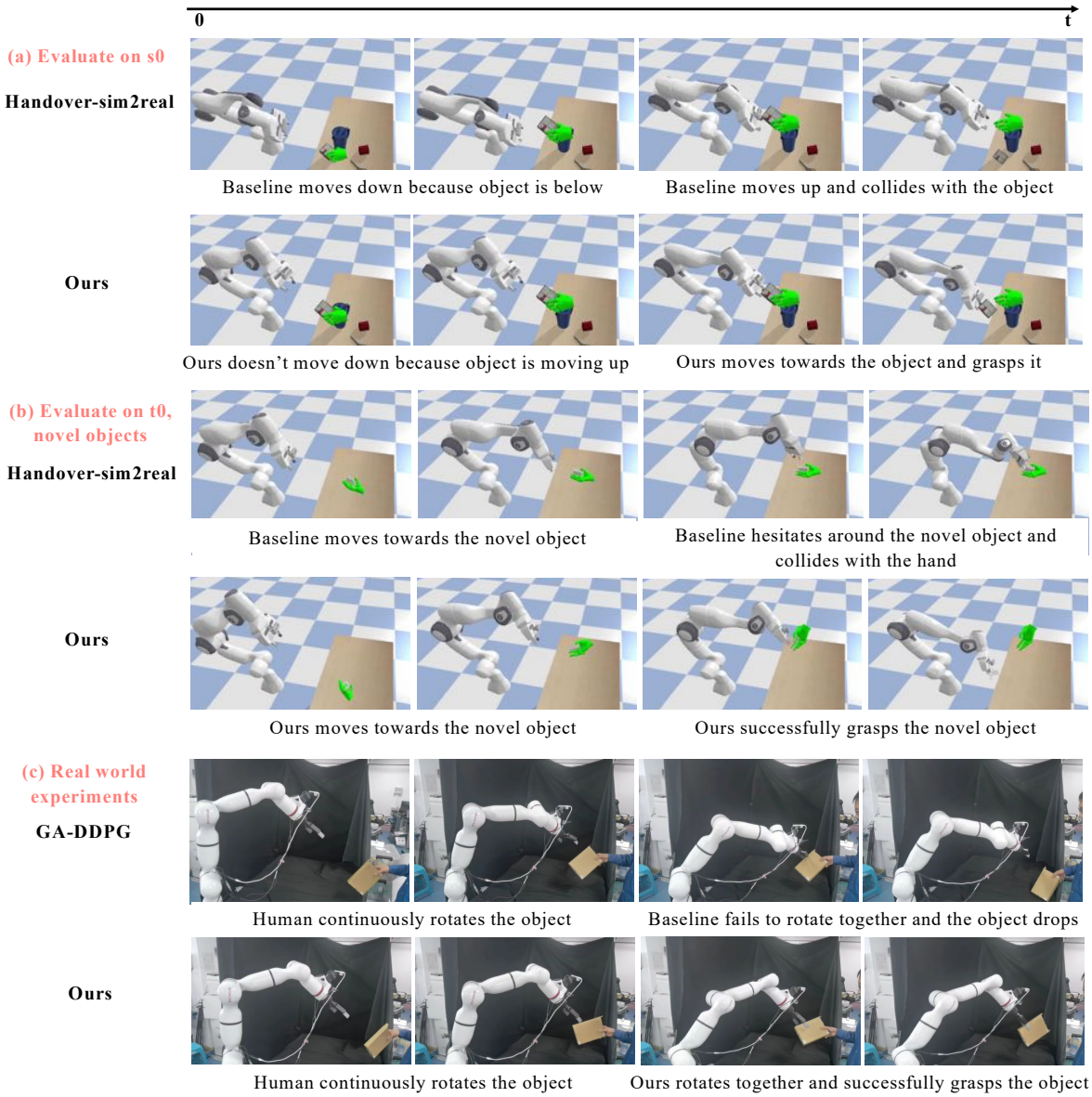

Figure 4. **Qualitative results**. We in detail compare different methods in simulators and deploy them in the real-world platform.

Methods	Simple Setting	Complex Setting
Handover-Sim2real	56.7%	33.3%
Ours	90.0%	70.0%

Table 3. **Sim-to-Real Experiments.** We report the success rate of our method and HandoverSim2real in 2 different settings.

across 5 objects in 2 different settings. In the simple setting, users hand each object to the gripper without quick movements. In the complex setting, users execute a relatively long and quick trajectory. The results are reported in Table 3. We observe that our model gets better performance in completing the handover process across various objects and scenarios. Figure 4(c) shows examples of the real-world handover trials.

5. Conclusion

In this work, we present a novel framework GenH2R for scaling up the learning of human-to-robot handover. We introduce a new simulator GenH2R-Sim and generate a million human handover animations to facilitate generalizable H2R handover learning. We then propose a distillation-friendly demonstration generation method that automatically produces a million high-quality demonstrations suitable for learning. We further introduce a forecast-aided 4D imitation learning method for effective demonstration distillation. Our experiments demonstrate that scaling-up efforts result in substantial improvement of generalizability to novel geometry and complex motion, both in the simulator and the real world.

References

[1] Christoph Bartneck, Tony Belpaeme, Friederike Eyssel, Takayuki Kanda, Merel Keijsers, and Selma Šabanović. *Human-robot interaction: An introduction*. Cambridge University Press, 2020. 2

[2] Antonio Bicchi and Vijay Kumar. Robotic grasping and contact: A review. In *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065)*, pages 348–353. IEEE, 2000. 2

[3] Aude Billard, Sylvain Calinon, Ruediger Dillmann, and Stefan Schaal. Survey: Robot programming by demonstration. Technical report, Springrer, 2008. 3

[4] Jeannette Bohg, Antonio Morales, Tamim Asfour, and Danica Kragic. Data-driven grasp synthesis—a survey. *IEEE Transactions on robotics*, 30(2):289–309, 2013. 2

[5] Samarth Brahmbhatt, Chengcheng Tang, Christopher D Twigg, Charles C Kemp, and James Hays. Contactpose: A dataset of grasps with object contact and hand pose. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 361–378. Springer, 2020. 2

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1

[7] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2, 3

[8] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9044–9053, 2021. 2, 3, 6, 14, 17

[9] 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)*, pages 6941–6947. IEEE, 2022. 1, 2, 3, 6, 7, 12, 14, 16, 17

[10] Sammy Christen, Wei Yang, Claudia Pérez-D'Arpino, Otmar Hilliges, Dieter Fox, and Yu-Wei Chao. Learning human-to-robot handovers from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9654–9664, 2023. 2, 5, 6, 7, 13, 14, 16, 17

[11] Gianluca Corsini, Martin Jacquet, Hemjyoti Das, Amr Afifi, Daniel Sidobre, and Antonio Franchi. Nonlinear model predictive control for human-robot handover with application to the aerial case. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7597–7604. IEEE, 2022. 2

[12] Murtaza Dalal, Ajay Mandlekar, Caelan Garrett, Ankur Handa, Ruslan Salakhutdinov, and Dieter Fox. Imitating task and motion planning with visuomotor transformers. *arXiv preprint arXiv:2305.16309*, 2023. 3

[13] Yuhao Dong, Zhuoyang Zhang, Yunze Liu, and Li Yi. Nsm4d: Neural scene model based online 4d point cloud sequence understanding. *arXiv preprint arXiv:2310.08326*, 2023. 5

[14] Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021. 3

[15] Clemens Eppner, Arsalan Mousavian, and Dieter Fox. Acronym: A large-scale grasp dataset based on simulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6222–6227. IEEE, 2021. 2, 5

[16] Hehe Fan, Yi Yang, and Mohan Kankanhalli. Point 4d transformer networks for spatio-temporal modeling in point cloud videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14204–14213, 2021. 5

[17] Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J Black, and Otmar Hilliges. Arctic: A dataset for dexterous bimanual hand-object manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12943–12954, 2023. 2

[18] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 2023. 2

[19] Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. In *Conference on robot learning*, pages 357–368. PMLR, 2017. 3

[20] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020. 3

[21] Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 4:265–293, 2021. 2, 3

[22] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. 2

[23] Huy Ha, Pete Florence, and Shuran Song. Scaling up and distilling down: Language-guided robot skill acquisition. *arXiv preprint arXiv:2307.14535*, 2023. 2, 3

[24] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11807–11816, 2019. 2

[25] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022. 3

[26] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018. 5, 14

[27] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21013–21022, 2022. 2, 6

[28] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021. 3

[29] Naresh Marturi, Marek Kopicki, Alireza Rastegarpanah, Vijaykumar Rajasekaran, Maxime Adjigble, Rustam Stolkin, Aleš Leonardis, and Yasemin Bekiroglu. Dynamic grasp and trajectory planning for moving objects. *Autonomous Robots*, 43:1241–1256, 2019. 2

[30] Michael James McDonald and Dylan Hadfield-Menell. Guided imitation of task and motion planning. In *Conference on Robot Learning*, pages 630–640. PMLR, 2022. 3

[31] OpenAI. Gpt-4 technical report, 2023. 1

[32] Valerio Ortenzi, Akansel Cosgun, Tommaso Pardi, Wesley P Chan, Elizabeth Croft, and Dana Kulić. Object handovers: a review for robotics. *IEEE Transactions on Robotics*, 37(6): 1855–1873, 2021. 2

[33] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988. 3

[34] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 5, 14

[35] Patrick Rosenberger, Akansel Cosgun, Rhys Newbury, Jun Kwan, Valerio Ortenzi, Peter Corke, and Manfred Grafinger. Object-independent human-to-robot handovers using real time robotic vision. *IEEE Robotics and Automation Letters*, 6(1):17–23, 2020. 2

[36] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *Proceedings third international conference on 3-D digital imaging and modeling*, pages 145–152. IEEE, 2001. 5, 14

[37] Thomas B Sheridan. Human–robot interaction: status and challenges. *Human factors*, 58(4):525–532, 2016. 2

[38] Shuran Song, Andy Zeng, Johnny Lee, and Thomas Funkhouser. Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations. *IEEE Robotics and Automation Letters*, 5(3):4978–4985, 2020. 2

[39] Zachary Teed and Jia Deng. Raft-3d: Scene flow using rigid-motion embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8375–8384, 2021. 5

[40] Lirui Wang, Yu Xiang, and Dieter Fox. Manipulation trajectory optimization with online grasp synthesis and selection. In *Robotics: Science and Systems (RSS)*, 2020. 4

[41] Lirui Wang, Yu Xiang, and Dieter Fox. Manipulation trajectory optimization with online grasp synthesis and selection. In *Proceedings of Robotics: Science and Systems (RSS)*, 2020. 5, 7, 13, 14

[42] Lirui Wang, Yu Xiang, Wei Yang, Arsalan Mousavian, and Dieter Fox. Goal-auxiliary actor-critic for 6d robotic grasping with point clouds. In *Conference on Robot Learning*, pages 70–80. PMLR, 2022. 5, 7, 14, 16, 17

[43] Ruicheng Wang, Jialiang Zhang, Jiayi Chen, Yinzhen Xu, Puhao Li, Tengyu Liu, and He Wang. Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. *arXiv preprint arXiv:2210.02697*, 2022. 3

[44] Ruicheng Wang, Jialiang Zhang, Jiayi Chen, Yinzhen Xu, Puhao Li, Tengyu Liu, and He Wang. Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11359–11366. IEEE, 2023. 2

[45] Hao Wen, Yunze Liu, Jingwei Huang, Bo Duan, and Li Yi. Point primitive transformer for long-term 4d point cloud video understanding. In *European Conference on Computer Vision*, pages 19–35. Springer, 2022. 5

[46] Wei Yang, Chris Paxton, Arsalan Mousavian, Yu-Wei Chao, Maya Cakmak, and Dieter Fox. Reactive human-to-robot handovers of arbitrary objects. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3118–3124. IEEE, 2021. 2

[47] Ruolin Ye, Wenqiang Xu, Zhendong Xue, Tutian Tang, Yanfeng Wang, and Cewu Lu. H2o: A benchmark for visual human-human object handover analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15762–15771, 2021. 2

[48] Gu Zhang, Hao-Shu Fang, Hongjie Fang, and Cewu Lu. Flexible handover with real-time robust dynamic grasp trajectory generation. *arXiv preprint arXiv:2308.15622*, 2023. 2

[49] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 1

[50] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on*

Robotics and Automation (ICRA), pages 5628–5635. IEEE, 2018. 3

GenH2R: Learning Generalizable Human-to-Robot Handover via Scalable Simulation, Demonstration, and Imitation Supplementary Material

The supplementary material offers additional details on various aspects of the method and experiments. Refer to the table of contents below for an overview. Section A provides additional details and clarifications on our methods. Sections B and C present supplementary experiments on baselines, along with additional quantitative and qualitative results in the simulation and real-world scenarios, respectively. Section D discusses the limitations of our work and explores potential research directions for future human-to-robot handovers and human-robot interactions.

Contents

A. More Method Details

A.1. GenH2R-Sim

In this section, we provide details on the generation of hand-object moving trajectories and our simulator GenH2R-Sim.

A.1.1 Hand-Object Moving Trajectory Generation

Note: The unit for all positions in the following paragraphs is meters.

In HandoverSim [9] and GenH2R-Sim, the robot arm's base is located at $(0.61, -0.50, 0.875)$, and the center of the table surface is at $(0.61, 0.28, H = 0.92)$. To synthesize a hand-object moving trajectory, we start by generating the object's trajectory.

For the translation part, we uniformly sample a starting point from the starting region $[0.3, 0.9] \times [0, 0.2] \times [H + 0.1, H + 0.3]$. Subsequently, we sample several endpoints from the activity region $[0.3, 0.9] \times [-0.3, 0.1] \times [H + 0.1, H + 0.7]$ and employ Bézier curves to connect the starting point and the endpoints. For each Bézier curve, a key point is sampled from a Gaussian distribution centered at the midpoint with a standard variation of 0.2, and the translation speed along this curve is uniformly sampled from $[0.1, 0.2]\ \mathrm{m\,s^{-1}}$.

For the rotation part, we uniformly sample a rotation $R \in \mathrm{SO}(3)$ as the starting object orientation. When the object travels along a Bézier curve, we rotate the object about a random rotation axis with an angular speed uniformly sampled from $[0.5, 1]\ \mathrm{rad\,s^{-1}}$.

After generating the object trajectory $\xi = (\mathcal{T}_0, \mathcal{T}_1, \cdots, \mathcal{T}_{T-1})$, we correspondingly generate the hand pose trajectory $\varsigma = \xi \circ T^{\text{hand}}_{\text{object}}$, where $T^{\text{hand}}_{\text{object}}$ is the hand pose in the object reference frame.

A.1.2 More Details about GenH2R-Sim

In real-world handovers, in order to enhance stability, our motion typically stops when another person's hand is close to the object. We incorporate this characteristic into our simulator, making it reactive to the robot arm's motion. To be specific, suppose $p \in \mathbb{R}^3$ is the current position of the gripper's tip, and $Q \subset \mathbb{R}^3$ is the current object point cloud. If $\min_{q \in Q} \|p - q\| \leq 0.1$, the hand and object will stop moving, awaiting the robot arm to grasp the object. We believe that this modification makes the cooperative handover process more realistic, transforming it from a simple chase-and-grasp game into a more authentic interaction.

It is important to note that we apply this modification exclusively in our simulation environment, GenH2R-Sim, for the benchmarks "t0" and "t1". We refrain from modifying it in HandoverSim [9] for the benchmark "s0" to ensure a fair comparison with the exact results obtained by baseline methods.

A.2. Generating Demonstrations for Distillation

A.2.1 Clarification of Different Methods

We would like to provide further clarification regarding the terms used to describe various demonstration generation methods, as outlined in Table 3 of the manuscript. These terms include **destination planning**, **dense planning**, and **landmark planning**.

Destination planning refers to the foresighted planner discussed in Section 3.3, which plans directly to the object's destination at the beginning. While the generated demonstrations exhibit smoothness, they lack vision-action correlation in complex scenarios and are not distillation-friendly.

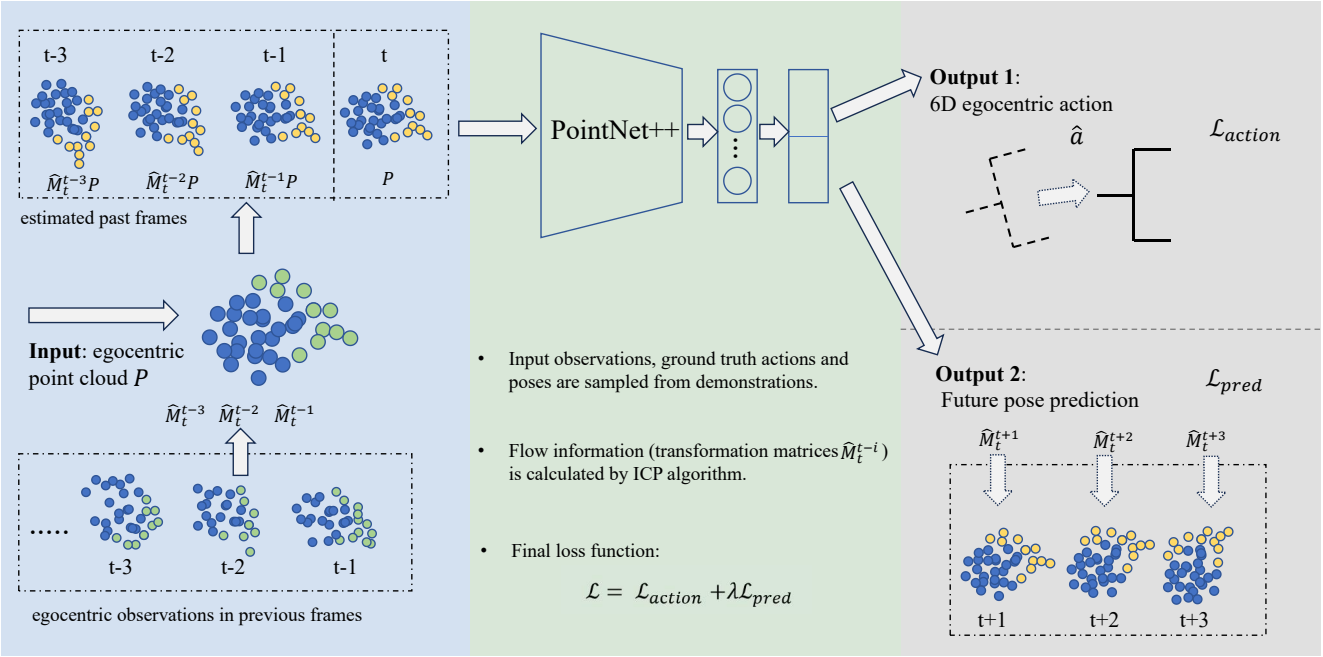

Figure 5. **Forecast-Aided 4D Imitation Learning Pipeline.** The network receives egocentric point cloud input and produces egocentric 6D actions as output. For each input point, we compute its past coordinates using flow information obtained through the Iterative Closest Point algorithm. Subsequently, we employ PointNet++ to encode the processed point cloud into a low-dimensional global feature. The policy head decodes this feature into a 6D egocentric action, serving as the primary policy output. Simultaneously, the prediction head decodes the feature into future pose transformations, contributing to the auxiliary loss.

Dense planning represents the special case of our method, where the planner plans at each time step based on the current object position. Although the generated demonstrations ensure a strong vision-action correlation, they suffer from the zigzag issue, resulting in slower performance.

Landmark planning denotes our method, where the planner periodically plans based on the object's position at future landmarks.

A.2.2 Trajectory Resampling Strategy

In this section, we introduce another resampling strategy designed to enhance the vision-action correlation in expert demonstrations.

Upon successfully solving the Inverse Kinematics (IK) to achieve the 6D target grasping pose, the subsequent task involves generating a trajectory from the initial 7D joint configuration C to the destination 7D joint configuration D. Through optimization, OMG-Planner [41] produces a trajectory $C_0 = C, C_1, \cdots, C_{N-1}, C_N = D$ with fixed time steps N. However, a challenge arises as the step size of the expert action is influenced by the distance between the initial end effector pose and the target grasping pose. Specifically, if the target grasping pose is far away from the initial end effector pose, the expert will move faster; conversely, if the target grasping pose is close to the initial end effector pose, the expert will move slower. This variability in step size can potentially confuse the vision-based model, which lacks awareness of the initial end effector pose and struggles to discern the expert's speed.

To address this issue, we conduct a resampling process to refine the obtained trajectory. Let $s_i = \sum_{j=1}^{i} \|C_j - C_{j-1}\|$ represent the accumulated step length, and L denote the hyperparameter controlling the desired step length. The resampled trajectory, denoted as $C'_0 = C_0, C'_1, \cdots, C'_{M-1}, C'_M = C_N$, where $(M-1)L < s_N \leq ML$, and for each $1 \leq i \leq M-1$, suppose $s_j \leq iL \leq s_{j+1}$, then

$$C'_i = \frac{s_{j+1} - iL}{s_{j+1} - s_j} C_j + \frac{iL - s_j}{s_{j+1} - s_j} C_{j+1}. \tag{3}$$

The resampling ensures that, regardless of the proximity between the initial end effector pose and the target grasping pose, the resulting trajectory maintains a consistent step length. This characteristic makes the expert demonstrations more conducive to distillation for vision-based models.

A.3. Forecast-Aided 4D Imitation Learning

Figure 5 provides an overview of our Forecast-Aided 4D Imitation Learning process. Similar to HandoverSim2real [10], We initiate the pipeline by obtaining an egocentric hand and object point cloud from the simulator, together with the one-hot encoding for hand/object labels. We

		s0 (Sequential)			s0 (Simultaneous)			t0			t1		
		S	T	AS	S	T	AS	S	T	AS	S	T	AS
	OMG Planner† [41]	62.50	8.31	22.5	-	-	-	-	-	-	-	-	-
train on s0	GA-DDPG [42]	50.00	**7.14**	22.5	36.81	**4.66**	23.6	23.59	7.31	10.3	46.7	**5.50**	26.9
	Handover-Sim2real [10]	75.23	7.74	30.4	68.75	6.23	35.8	29.17	6.29	15.0	52.40	7.09	23.8
	Handover-Sim2real* [10]	64.35	7.61	26.7	25.69	5.43	15.0	28.56	4.73	17.9	30.60	5.98	16.5
	Destination Planning	74.31	7.98	28.7	76.16	5.89	41.7	25.68	5.34	15.1	48.4	7.49	20.5
	Dense Planning	74.77	8.14	28.0	75.45	6.06	40.3	27.30	5.49	15.7	52.3	7.44	22.4
	Landmark Planning	77.78	8.15	29.0	79.17	6.06	42.0	29.63	5.22	17.7	54.2	7.41	23.3
train on t0	GA-DDPG [42]	54.76	7.26	24.2	44.68	5.30	26.5	24.05	4.70	15.3	25.50	5.86	14.1
	Handover-Sim2real [10]	65.97	7.18	29.5	62.50	6.04	33.5	33.71	5.91	18.4	47.10	6.35	24.1
	Handover-Sim2real* [10]	63.55	7.58	26.5	38.89	5.29	23.1	33.31	**4.64**	21.4	33.35	5.81	18.4
	Destination Planning	0.93	11.76	0.1	6.48	11.22	0.9	5.96	7.57	2.5	1.60	11.38	0.2
	Dense Planning	81.48	8.52	28.1	84.95	6.32	43.7	38.04	6.06	20.3	57.90	7.23	25.7
	Landmark Planning	**86.57**	7.62	**35.8**	**85.65**	5.38	**50.2**	**41.43**	4.97	**25.6**	**68.33**	6.14	**36.1**

Table 4. **Evaluating on different benchmarks.** We compare our method against baselines from the test set of HandoverSim [9] benchmark ("s0 (sequential)" and "s0 (simultaneous)") and our GenH2R-Sim benchmark ("t0" and "t1"). We use the best-pretrained models from the repositories of GA-DDPG [42] and Handover-Sim2real [10] for evaluation. The results for our method are averaged across 3 random seeds. Note that S means success rate(%). T means time(s). AS means average success(%) as defined in Equation 2. †: This method [41] is evaluated with ground-truth states and cannot handle dynamic handover like "s0 (Simultaneous)", "t0" and "t1".*: We reproduce the results of HandoverSim2real in the true simultaneous setting as detailed in Section 4.1 to make a fair comparison.

augment the feature of each point with its 3D coordinates in the past n time steps, computed from the estimated flow information and the current 3D coordinates. As a result, each point has a feature vector of length $3 + 2 + 3 \cdot n$.

We then introduce the specific method for computing flow information. Given the end effector pose, we convert the point cloud from the egocentric frame to the static world frame and store it in a buffer. In each time step, we retrieve the point clouds of several past time steps and leverage the Iterative Closest Point (ICP) registration algorithm [36] to estimate transformation matrices between the current point cloud and past point clouds in the world frame. While these transformations may be slightly imprecise due to the incomplete point cloud input, they can provide sufficient flow information for each point. Finally, the flow information is converted back to the current egocentric frame, which serves as an important part of our feature representation.

Then we feed the point cloud with processed features into PointNet++ [34] to obtain a global low-dimensional representation. This representation is then decoded by two heads: the policy head and the prediction head. The policy head decodes it into a 6D egocentric action, and $\mathcal{L}_{action}$ is computed following the approach defined in [26]. Simultaneously, the prediction head decodes the representation into transformations between the current and future object poses, with $\mathcal{L}_{pred}$ computed as a motion prediction loss.

Similar to Handover-Sim2real [10], our policy only outputs 6D egocentric actions in a closed loop. For decisions on whether to grasp the object and place it in the target location, we adopt a heuristic method akin to GA-DDPG [42]. Specifically, if the number of points in the gripper's vicinity exceeds a predefined threshold, the robot attempts to grasp the object and retract to the target location in an open-loop fashion, foregoing the execution of the egocentric actions predicted by the policy network.

B. Simulation Experiments Details

B.1. Discussions on the Simultaneous Setting

It's essential to clarify that in handoverSim [9] and Handover-Sim2real [10], the simultaneous setting (also referred to as "w/o hold") implies that the robot is allowed to move from the beginning of the episode. **We adhere to this definition in our GenH2R-Sim and our methods**. In the settings "s0 (Simultaneous)", "t0", and "t1", the robot initiates movement immediately upon detecting the object.

However, we observed that in the code of Handover-Sim2real, a parameter named "TIME_WAIT" is used to specify the time to wait before executing the actual action in different settings. In "s0 (Sequential)", "TIME_WAIT" is set to 3s (matching the 3-second duration of DexYCB [8] handover motion), but in "s0 (Simultaneous)", "TIME_WAIT" is set to 1.5s, which implies a 1.5s wait for the simultaneous setting. We believe this value should be 0s for the simultaneous setting.

The author adjusted this parameter after observing that humans typically move faster than the robot. This modification aims to prevent collisions, especially when the robot approaches the human while the human is also approaching the object. The change is implemented to reduce the number of failures caused by attempting to grasp while the human is still in motion or before the human completes pick-

ing up the object from the table.

We believe that improving performance makes sense, but we consider the true simultaneous setting to be closer to real-world scenarios. It's crucial not to make the person wait for an extended period during a short-term handover process, so we avoid adjusting the waiting time manually.

For a fair comparison, we reproduce the results in the true simultaneous setting.

As highlighted in the blue rows of Table 4, our method demonstrates significant improvements. When trained on "s0", our method achieves improvements of 13.43%, 53.48%, 1.07%, and 23.6% in the successful rate of "s0 (Sequential)", "s0 (Simultaneous)", "t0", and "t1" settings, respectively. When trained on "t0", our method achieves improvements of 23.02%, 46.76%, 8.12%, and 34.98% across the four test sets. Notably, our method excels in the simultaneous setting when hands are in motion. This highlights that our distillation-friendly demonstrations can better extract valuable insights from more complex scenarios and showcase enhanced generalizability compared to the baseline.

B.2. Baseline Experiments

When training Handover-Sim2real, we follow the two-stage teacher-student training approach outlined in the paper. In the pretraining stage, the duration of hand and object movement is clipped to 3 seconds, and the expert waits for 3 seconds before planning to grasp the static object. In the fine-tuning stage, the entire movement is used. In the original codebase, the robot waits 1.5 seconds. In our reproduced version, the robot does not need to wait, moving directly with the dynamic hand and object.

B.3. More Ablations

We conducted additional ablations for our method, as shown in Table 5. All these methods are trained in "t0" and tested in "t0".

In Section 3.4 of the manuscript, we mentioned that frame stacking is a straightforward approach but struggles to capture both motion and geometry effectively. To quantitatively demonstrate this, we compared it with our method based on forecast-aided 4D imitation learning and found a 5.26% decrease in the success rate. This highlights the effectiveness of our method in learning from 4D spatial-temporal information.

Moreover, excluding the endpoints from consideration when selecting the landmark results in a 1.70% decrease in the success rate. This indicates the importance of incorporating the endpoints when choosing the landmark state.

B.4. Evaluating on Demonstration Generation

Figure 6 compares different expert demonstration generation variants by showing their accumulated success rate w.r.t. to success time on "t0".

Methods	S	T	AS
w/o Flow	31.66	**4.64**	20.2
w/o Prediction	39.18	5.03	24.1
w/o Flow & Prediction	37.04	4.88	23.1
w/o Endpoints	39.73	**4.86**	24.9
Frame Stacking	35.17	4.71	21.5
Ours	**41.43**	4.97	**25.6**

Table 5. **Ablations on different modules.** "w/o Flow" means not using flow information in the input. "w/o Prediction" means not adding prediction loss in the final loss.

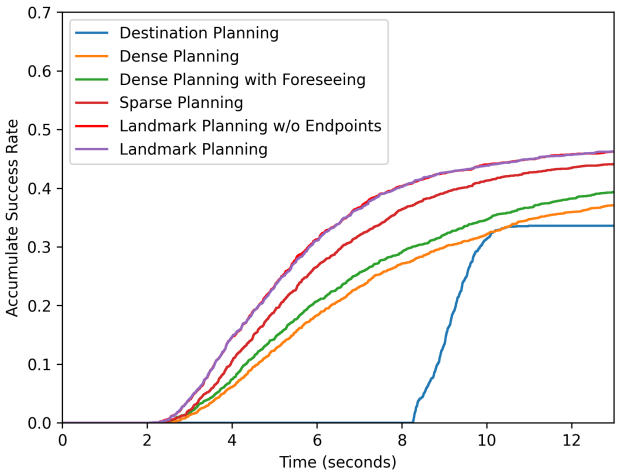

Figure 6. **Comaprison of different expert demonstrations.**

Destination Planning is suboptimal, as it plans a straight trajectory directly to the destination, which is not only slow but also nearly impossible to be distilled to vision-based agents when the object trajectory is complex.

With access to future object states and with less frequent replanning periods, Landmark Planning exhibits a larger success rate and faster success time compared with Dense Planning. To ablate the two factors, we also analyze two additional settings. In Dense Planning with Foreseeing, OMG-Planner still always replans at each step, but based on the future object states. In sparse Planning, OMG-Planner replans with the same sparsity as Landmark Planning, but it can only plan based on the current object state. We can analyze from the curves that both decreasing the replanning period and foreseeing future states can help the expert to achieve a higher success rate and lower success time.

B.5. Training Details

In our training process, we employ a single Nvidia GeForce RTX 3090 (24GB) with a batch size of 256. The training spans 80,000 iterations, with each iteration involving the random sampling of 256 observation-action pairs from

Figure 7. **Real-world Handover System Setup.** Our system consists of an xMate 3 robot, which is similar to a Franka Panda robot, and two RealSense Depth Camera D435 devices.

demonstrations. We use the Adam optimizer with a learning rate of 0.001 and weight decay of 0.0001. The entire process takes approximately 8 hours to train our method. We incorporate flow information from the last 3 time steps and calculate the prediction loss for the next 3 time steps with the weighting hyper-parameter λ as 0.1.

C. Real World Experiments Details

C.1. Setup

In our real-world handover system, illustrated in Figure 7, we utilize a ROKAE xMate 3 ER series flexible collaborative robot along with two Intel RealSense Depth Camera D435 devices.

Since the action space is 6D Cartesian, while our policies and the baselines are trained with Franka Panda robot in HandoverSim [9] and GenH2R-Sim, we can deploy them to the xMate3 robot despite their morphology difference, as they have own position controllers. Thus, our real-world experiments not only assess the policy's generalization for sim-to-real transfer but also evaluate its adaptability to different robots.

Our real-world handover system incorporates two Intel RealSense Depth Camera D435 devices to enhance the egocentric point cloud. In Figure 7, the higher camera captures the point cloud in proximity to the robot gripper but lacks visibility further ahead. Conversely, the lower camera captures the point cloud ahead of the robotic arm but misses details near the gripper. By merging the point clouds from both cameras, we achieve a comprehensive view, which is beneficial for the deployed policies.

The robot initiates movement upon perceiving a point cloud. In case the object is not visible during the handover process, the robot tracks the object's last known pose. The

Figure 8. **Various objects for real-world handover.** The image above displays relatively simple objects for handover, such as the can, the box, the bottle, or some square objects. In contrast, the image below showcases more challenging objects for handover, including the plastic stool, the teapot, the sticky tape, or some soft objects with diverse shapes.

baselines, GA-DDPG [42] and Handover-Sim2real [10], are treated similarly to [10]. For GA-DDPG, the pre-trained policy model is loaded, and heuristic methods determine whether to grasp. For Handover-Sim2real, both the pre-trained policy model and the pre-trained grasp prediction network are loaded.

C.2. User Study

The study involved 6 participants who compared our forecast-aided 4D imitation learning method based on landmark planning demonstrations, with two baseline methods, GA-DDPG and Handover-Sim2real, across 5 different objects in two settings. **It is important to note that** due to the release of the Handover-Sim2real source code just in November 2023, we were unable to deploy it on the real-world handover system before the manuscript deadline. These results will be included in the manuscript in subsequent revisions.

The selected 5 objects for evaluation include a mug, a bottle, a cracker box, a sticky tape, and a chips can. The

	Simple setting			Complex setting		
	GA-DDPG[42]	Handover-Sim2real[10]	Ours	GA-DDPG[42]	Handover-Sim2real[10]	Ours
1. plastic mug	5 / 6	4 / 6	6 / 6	2 / 6	2 / 6	4 / 6
2. EFES bottle	5 / 6	4 / 6	6 / 6	4 / 6	3 / 6	5 / 6
3. Cheez-It box	1 / 6	4 / 6	4 / 6	3 / 6	1 / 6	4 / 6
4. sticky tape	3 / 6	3 / 6	5 / 6	3 / 6	3 / 6	4 / 6
5. Pringle can	4 / 6	2 / 6	6 / 6	1 / 6	1 / 6	4 / 6
total	18 / 30 (60%)	17 / 30 (57%)	**27 / 30 (90%)**	13 / 30 (43%)	10 / 30 (33%)	**21 / 30 (70%)**

Table 6. **User study for sim-to-real experiments.** Each method was evaluated by six individuals for every object in both the simple and complex settings. Failure scenarios included collisions with the human hand, dropping to the table, or exceeding the time limit ($T_{\text{max}} = 13$ seconds). Our method consistently outperformed the baselines in the real-world handover system in both simple and complex settings, aligning with the results observed in the simulation experiments.

cracker box is an object shown in DexYCB [8] trajectories, while the other 4 novel objects may exhibit more diverse geometries.

In the simple setting, users hand each object to the gripper in a straightforward manner. In the complex setting, users execute a relatively long and quick trajectory, involving both translations and rotations. For each specific object, we try to ensure that each participant executed a similar trajectory in the same setting for different methods, ensuring a fair comparison.

Table 6 provides a detailed breakdown of the results presented in Table 3 of the manuscript. Our method is compared with baselines across different settings, revealing a remarkable 34% improvement in the simple setting and a substantial 40% improvement in the complex setting from Handover-Sim2real. Notably, in the simple setting, our method demonstrates great generalizability to various objects, including new objects with different geometries or similar objects of different sizes. In the complex setting, our method exhibits smooth object tracking with predictive intention, resembling a more human-like approach to grasping handed objects. Further analysis will be elaborated in our accompanying video. It is noteworthy that Handover-Sim2real exhibits a lower success rate compared with GA-DDPG. One possible explanation is that the pretrained grasp prediction network may not be as robust as heuristic methods in determining whether to grasp, which may not be able to generalize well to novel objects and potentially increase the sim-to-real gap.

C.3. Generalization Study

In addition to direct comparisons with baseline methods, we conducted numerous real-world handover experiments involving different trajectories and objects.

Figure 8 showcases two sets of objects used in our experiments. The simple set comprises regular objects similar to those used in DexYCB [8] or HandoverSim [9], which are easier to pass and grasp. The difficult set includes more challenging objects with diverse shapes and geometries. We introduced variations in human behavior, such as different grasping poses or handover trajectories.

As shown in Figure 9, our qualitative experiments reveal the robustness of our policy, crafted through extensive large-scale demonstrations and an imitation learning framework. Trained within the GenH2R-Sim environment, our policy showcases effective generalization across diverse objects and various handover scenarios.

D. Limitations and Future Work

While in this paper significant progress has been achieved in the H2R handover task, we acknowledge certain limitations that could serve as inspiration for exciting future research.

In aspects of robot morphology, we concentrate on the relatively simple 7DoF robotic arms, characterized by a confined activity region and limited motion capabilities. In contrast, robots equipped with a movable base exhibit a broader range of motion, enabling them to navigate and interact within a more extensive spatial environment, enhancing their versatility and efficacy in various human-robot interaction tasks.

In aspects of human modeling, our current focus on the object and hand poses neglects the consideration of the entire human body. In real-world scenarios, robots may need to take into account not only the hand pose and trajectories but also the motion of the entire human body for more dynamic and generalizable interactions. Extending the simulation environment to model a more complex representation of the human, including body movements, poses a challenging yet practical avenue for future work in policy learning.

In aspects of human intention, our simulator currently does not incorporate human intention. Existing simulation environments have mainly focused on physical modeling, lacking representation of human behavior. In HandoverSim [9], for instance, the human hand does not respond to the robot's actions. In GenH2R-Sim, we introduce a more interactive element, where the human hand stops moving and waits for handover when the robot arm is close to the object. However, there is room for more complex and inter-

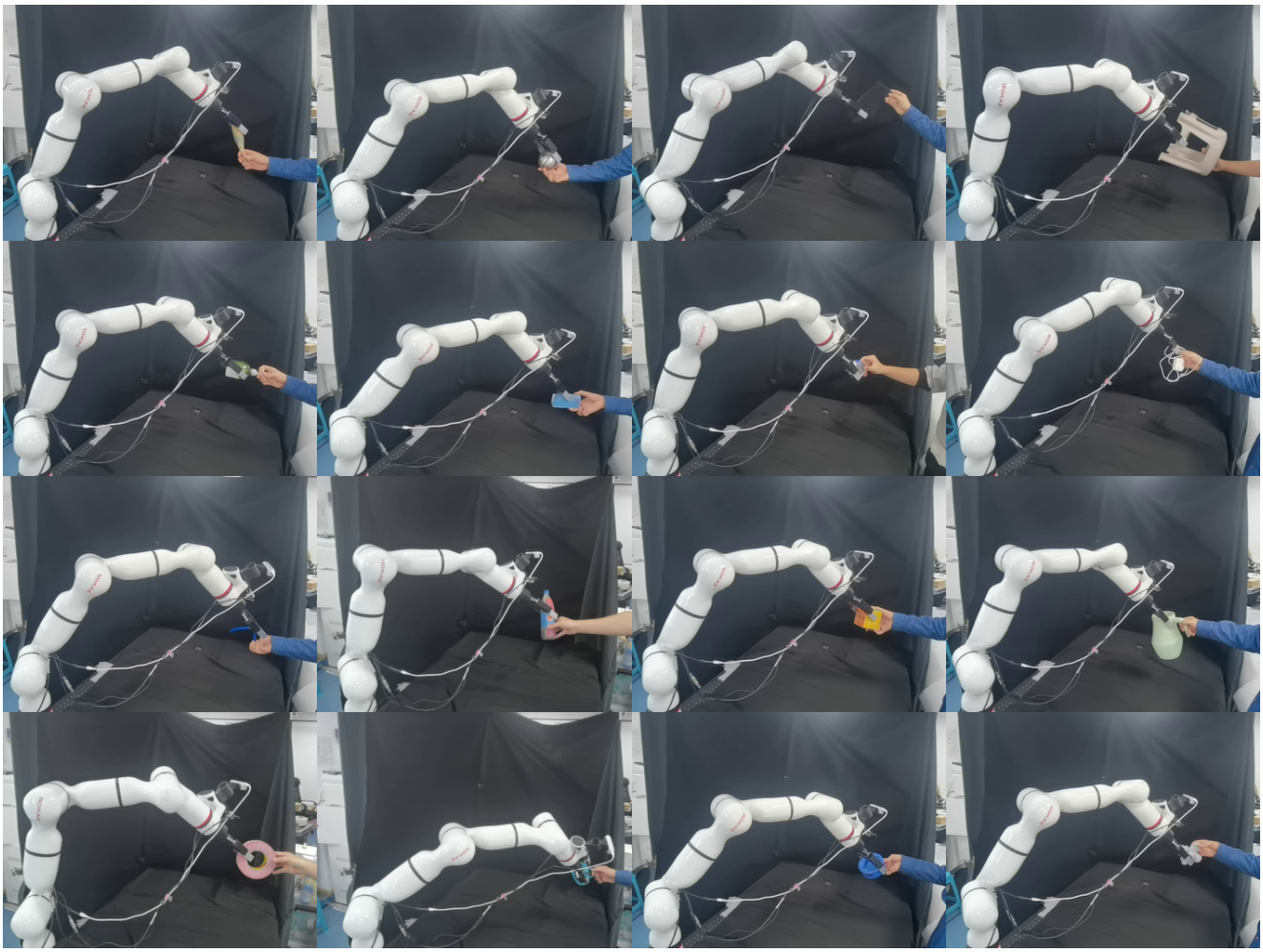

Figure 9. **Qualitative real-world results with various objects**. In the real-robot system, we qualitatively assess the generalization ability of our method by testing it with various objects.

esting modeling of human behavior. For instance, when the gripper moves rapidly toward the hand, the human may perceive danger and retract the hand. Introducing more sophisticated representations of human behavior in the simulator is crucial for a human-centric handover process.