Learning Human-to-Robot Handovers from Point Clouds
Supplementary Material

Sammy Christen[1,2*] Wei Yang[2] Claudia Pérez-D'Arpino[2]
Otmar Hilliges[1] Dieter Fox[2,3] Yu-Wei Chao[2]

[1]ETH Zurich [2]NVIDIA [3]University of Washington
{sammy.christen, otmar.hilliges}@inf.ethz.ch {weiy, claudiap, dieterf, ychao}@nvidia.com

The supplementary material of this work contains **a video** and this document. See the table of contents below for an overview. The video and code are available at https://handover-sim2real.github.io/.

Contents

A Method Details **1**
A.1. Task and Method 1
A.2 Loss Functions 2
A.3. Training Details 2

B Implementation Details **3**
B.1. Network architecture 3
B.2. Training Information 3

C Additional Simulation Evaluation **3**

D Limitations and Future Work **4**

E Sim-to-Real Transfer **6**
E.1. System Setup 6
E.2. Pilot Study 7
E.3. User Evaluation 7

A. Method Details

A.1. Task and Method

Action Space Actions are defined as transformations of the end-effector pose. In particular, an action contains the relative transformation of the 3D translation and 3D rotation of the end-effector. More formally, given the current gripper pose $\mathcal{T}_t \in \mathbb{SE}(3)$ at time t, the policy $\boldsymbol{\pi}$ predicts an action $\mathbf{a}_t = \mathcal{T}_{\boldsymbol{\pi}}$ that indicates the relative transformation of the end-effector. The next end-effector pose it then computed by transforming the current pose $\mathcal{T}_{t+1} = \mathcal{T}_t \mathcal{T}_{\boldsymbol{\pi}}$. We then use inverse kinematics to compute the next robot pose that is sent to the PD-controllers of the physics engine.

*This work was done during an internship at NVIDIA.

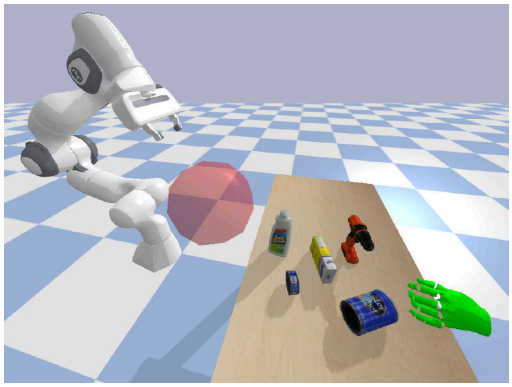

Figure 1. The simulation environment from HandoverSim. The red sphere indicates the goal region. Image source: Chao *et al*. [2].

Goal Space Goals are defined in a similar way. They correspond to the relative transformation between the current 6DoF end-effector pose $\mathcal{T}_t$ and the pre-grasp pose $\mathcal{T}_{\text{pre}}$, i.e., $\mathbf{g}_t = \mathcal{T}_{\text{pre}} \mathcal{T}_t^{-1}$. Note that in contrast to related work [18], we define the goal as the pre-grasp pose from where the object can be grasped via a forward grasping motion (see main paper Sec. 4.1) instead of the actual grasp pose.

Reward Function We use a sparse reward function. The reward is 1 if a task has been successfully completed, i.e., the robot has successfully taken over the object from the human and moved it to a predefined goal region (see red sphere in Fig. 1) without dropping or collision with the human hand. The reward is 0 in all other cases. If the robot collides with the human hand or the object is dropped, the episode terminates early and the reward stays 0.

Grasping Phase The grasping phase comprises the forward motion from the pre-grasp pose to the grasp pose, the closing of the gripper, and the retraction of the object (see Fig. 2). The forward motion moves the end-effector 8 cm into the z-direction of the gripper. Then, the gripper is closed to grasp the object. Finally, a retraction trajectory is executed in open-loop fashion to bring the object

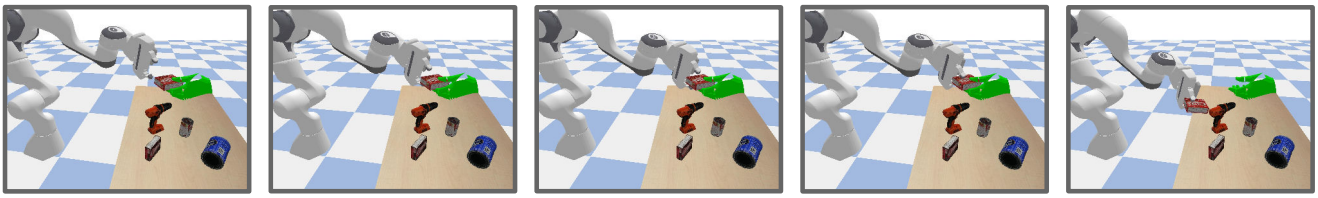

Figure 2. The grasping phase starts with a forward grasping motion that moves the end-effector from a pre-grasp pose to a grasp pose, then grasps the object, and finally moves it to a pre-defined goal region.

into a goal location. We compute the trajectory by linearly interpolating the path between the end-effector and the end-effector goal location, and then transform the trajectory into robot poses using inverse kinematics. Note that the grasping phase is non-learning based and purely based on heuristics and open-loop control.

Perception Module To transform the segmented object point cloud $\mathbf{p}_o$ and hand point cloud $\mathbf{p}_h$ into a single point cloud $\mathbf{p}$ of constant size, we down- or up-sample both point clouds. Since there are usually more points contained in the segmented object point cloud (e.g., due to the hand being occluded), we use a ratio of 87.5% to 12.5% for the object and hand point clouds when sampling the single point cloud $\mathbf{p}$. We add two one-hot encoded vectors to the point cloud $\mathbf{p}$ to indicate which points are from the object or hand point clouds, respectively. We keep track of the latest available point cloud $\mathbf{p}_{t-1}$. If there is no point cloud available at time t, we use the latest available point cloud $\mathbf{p}_t = \mathbf{p}_{t-1}$.

A.2. Loss Functions

We provide a more detailed description of the loss functions used to train our method. We mostly follow the description of [18].

Policy Loss As stated in the main paper (Eq. 3), our loss function for the policy is defined as:

$$L(\theta) = \lambda L_{\text{BC}} + (1-\lambda) L_{\text{DDPG}} + L_{\text{AUX}}. \tag{1}$$

We first introduce the point matching loss function [**?**]:

$$L_{\text{POSE}}(\mathcal{T}_1, \mathcal{T}_2) = \frac{1}{|X_g|} \sum_{x \in X_g} ||\mathcal{T}_1(x) - \mathcal{T}_2(x)||_1, \tag{2}$$

where X_g is a set of pre-defined points on the end effector. The loss computes the L1 norm between of these points after applying pose transformations $\mathcal{T}_1$ and $\mathcal{T}_2$ to the end-effector. The behavior cloning loss is defined as :

$$L_{\text{BC}}(\mathbf{a}_*, \mathbf{a}) = L_{\text{POSE}}(\mathbf{a}_*, \mathbf{a}). \tag{3}$$

The loss computes the L1 norm between these points after applying the relative transformation $\mathbf{a}$ predicted by the policy and the relative transformation $\mathbf{a}_*$ of the expert to the end effector. The auxiliary loss is defined similarly:

$$L_{\text{AUX}}(\mathbf{g}_*, \mathbf{g}) = L_{\text{POSE}}(\mathbf{g}_*, \mathbf{g}), \tag{4}$$

where $\mathbf{g}$ is an additional output of the policy that predicts the pre-grasp pose and $\mathbf{g}_*$ indicates the pre-grasp pose of the expert.

Critic Loss The critic loss is defined as:

$$L(\phi) = L_{\text{BE}} + L_{\text{AUX}}, \tag{5}$$

where L_{AUX} is identical to Eq. 4 and L_{BE} is the Bellman equation defined in Eq. 1 of the main paper.

Grasp Prediction Loss The loss for the grasp prediction network is defined as:

$$L(\zeta) = L_{\text{CE}}(\boldsymbol{\sigma}_\zeta(\psi(\mathbf{p})), \mathbf{y}), \tag{6}$$

where L_{CE} is a binary cross-entropy loss between the output predictions of the model $\boldsymbol{\sigma}_\zeta(\psi(\mathbf{p}))$ and the binary labels $\mathbf{y}$. The labels indicate whether a pre-grasp pose will lead to a successful grasp or not.

A.3. Training Details

Training Techniques We apply a variety of different techniques to make the policies more robust. To this end, during the sequential phase, we alternate between initializing the robot in a home position and random poses (that have the object and hand in its view). As proposed in [18], we occasionally compute optimal actions using DAGGER [14] during RL exploration and use them to supervise the policy's actions. Additionally, we start episodes of the RL agent with a random amount of initial actions proposed by the expert to further guide the training process. In the fine-tuning phase, we drop most of these techniques and start all the episodes from home position. We do not use DAGGER anymore and rollouts from the RL agent are not started with actions proposed by the expert.

Expert Demonstrations In the pretraining stage, we use motion and grasp planning [17]. We plan trajectories until the pre-grasp pose and then use the forward grasping motion (cf. main paper Sec. 4.1) to grasp the object. The ACRONYM dataset [5] is a large dataset for robot grasp planning based on physics simulation. It is used for grasp selection of the expert planner. However, because it does not consider the human hand, we first prefilter suitable grasps offline. To this end, we parse grasps from ACRONYM and combine them with the handover poses extracted from DexYCB [3], i.e., the pose at the last frame of the sequences where the hand and object are not moving. We check for collisions between the hand and the gripper using a mesh collider. We then filter out sequences where the robot and hand collide. During rollouts of expert trajectories, we frequently add random perturbations to the robot end-effector and replan the trajectory from the current pose.

Hindsight goals In the pretraining stage, the expert planner [17] provides goal labels for both expert demonstrations and RL rollouts. In the finetuning stage, we cannot rely on the goal selection from the planner anymore. We therefore employ the hindsight scheme from [18], which was used for training with novel objects, and utilize it for labeling sequences where the human and robot move simultaneously. In particular, if an episode is successful, we can use the pre-grasp pose from this episode as a label for supervision of the goal-prediction task.

Grasp Prediction Data Collection To collect samples for training the grasp prediction network, we utilize pre-grasps generated from ACRONYM [5]. We initialize the hand and object in the final pose of the handover trajectories from HandoverSim [2]. We then use inverse kinematics to compute a robot pose that matches the end-effector pre-grasp pose. We then rollout the forward grasping motion (cf. main paper Sec. 4.1) and check for grasp stability. The label is 1 if the grasp is successful, and 0 otherwise (e.g., hand collision or object drop). For each pre-grasp pose, we collect two more samples by adding small random noise (translation and rotation) to the end-effector pose. This is done to balance the positive and negative labels in the dataset [11], since the pre-grasp poses from ACRONYM will have a higher chance of leading to a stable grasp.

B. Implementation Details

B.1. Network architecture

We use PointNet++ [12] as backbone for the grasp prediction network, the actor network and the critic network. We use separate backbones for each network. The networks are fully-connected MLPs with three-layers and 256 neurons per layer. The critic network has two heads as output, one for predicting the Q-value and one for the auxiliary goal prediction. Similarly, the actor has one head for the goal prediction and one for the action prediction. The critic network takes as input the concatenation of a point cloud and an action (see Fig. 3 in the main paper) . During critic training, transitions from the replay buffer are used as actions and point clouds. On the other hand, during actor training, the actor's action predictions are used together with the point cloud from the replay buffer. The output of the grasp prediction network is a scalar that indicates a probability. We use Adam [9] as optimizer and ReLU activation functions during training of all networks. The learning rate is decreased over the course of training.

Training Parameters	Value
Num. iterations pretraining	10000
Num. iterations finetuning	5000
Buffer size pretraining	1e6
Buffer size finetuning	4e5
Parallel workers	3
Simulation timestep	1e-3s
Simulation steps per action	150
Network layers	3
Hidden size	256
Activation functions	ReLU
Optimizer	Adam [9]

Table 1. Overview of the most important parameters.

B.2. Training Information

We use TD3 [7] as our learning algorithm. To ensure a fair comparison with GA-DDPG, we use their implementation of TD3 and only make minimal changes to learning parameters. We therefore refer the reader to [18] for an exact description of all the parameters used and report only crucial or changed parameters in Tab. 1. In every iteration, we use three parallel workers to rollout episodes. Thereafter, we update our networks using 20 optimization steps. We run the pretraining for 10k iterations and the finetuning for 5k iterations. The grasp prediction network is trained for 1k iterations.

We train our method on a single Nvidia V100 32GB. Training the full method takes around 72-96 hours, about 36-48 hours for pretraining and 36-48 hours for finetuning. The grasp prediction network is trained offline and training takes roughly 1 hour.

C. Additional Simulation Evaluation

HandoverSim Benchmark For completeness, we report the results of the remaining settings from HandoverSim [2]. Namely, we add the settings "S1 - Unseen Subjects" in Tab. 3, "S2 - Unseen Handedness" in Tab. 4, and "S3 - Unseen Objects" in Tab. 5. Overall, we observe that the results are consistent with the main paper. In general, the

	success (%)	failure (%)		
		contact	drop	timeout
w/ Kinect noise	59.49	13.19	19.68	7.64
Ours	**75.23**	**9.26**	**13.43**	**2.08**

Table 2. We analyze the effect of simulated Kinect noise [8] on our model. Results are averaged over 3 random seeds.

main baseline GA-DDPG [18] struggles in the simultaneous setting. Our method has significantly better performance in terms of overall success rates, while retaining a slightly slower mean accumulated time for successful handovers. This is because GA-DDPG often goes for a grasp in the most direct path, whereas our approach searches for a safe pre-grasp pose, from where the object can be grasped. For a qualitative demonstration of this behavior, we refer to the supplementary video. We also compare our final model with the pretrained versions (*Ours w/o ft.*). The results further indicate that finetuning helps improve the model, especially in the simultaneous setting, e.g., the success rate in Tab. 3 improves from 62.78% to 73.33% with the finetuned model.

Notably, results on S2 and S3 suggest that our method can generalize well to unseen subjects and unseen objects. This result is important because in unstructured real world environments, neither objects nor subjects have been encountered during training.

Robustness Analysis We evaluate in simulation how *noisy observations* affect our pipeline by (1) adding simulated Kinect noise to depth images [8] and (2) testing with imperfect hand segmentation. For (2), we divide the hand into 6 different parts (fingers and palm) and re-label a subset of parts as object in the segmentation mask. We vary the mislabeling ratio ("0/6" no parts and "6/6" all parts) and sample randomly which parts will be mislabeled for a given episode. As expected, performance degrades with increasing noise in depth (*e.g.*, a 59.49% success rate in Tab. 2) and increasing mislabeling ratio (*e.g.*, decreasing success rate and increasing hand collisions in Fig. 3).

D. Limitations and Future Work

We will now discuss failure cases of our method and exciting directions for future work. Some failures occur with smaller objects, where the human hand often encloses large parts of the object. For the robot to find grasps where the gripper does not touch the hand at all in such cases is extremely difficult, especially when only having access to point cloud input. The grasp prediction task that decides when to switch from approaching to grasping is quite challenging, because a small change in end-effector pose can already cause a handover to fail. We sometimes find that the grasp prediction triggers the grasp too early or in an instance where the object will eventually drop. Since the grasp prediction network is trained offline, it may be improved by finetuning in online fashion with experiences from policy rollouts. Furthermore, we investigate *sensor-challenging* objects in real world transfers. Our depth sensor is vulnerable to transparent or dark objects, which may lead to failures of the policy (Fig. 4). Improving the vision pipeline to detect such objects reliably [21] could be a viable direction.

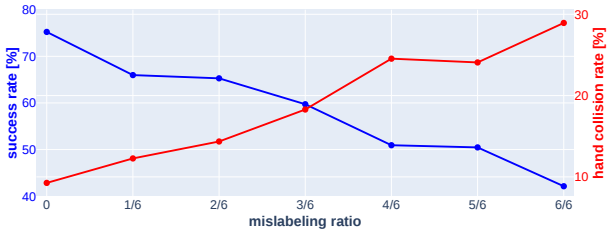

Figure 3. **Segmentation Mislabeling.** Our method's success (blue) and hand collision rate (red) under increasing degree of mislabeling hand as object segments.

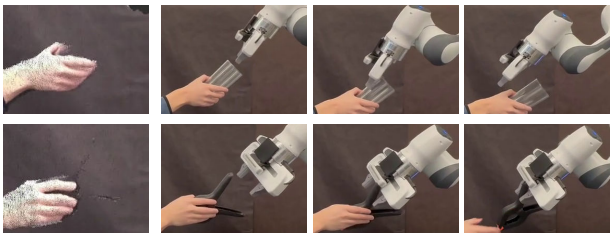

Figure 4. **Real World Failures**. Left: Missing/sparse point cloud of transparent/dark objects in real world perception. Right: Handover policy behavior.

We find that most human trajectories in HandoverSim have roughly the same length. A future direction can therefore include exploring a wider variety of human behaviors. For example, in a realistic, interactive setting the human may be constantly moving, and the robot should only take objects from the human once it wants to hand them over. Anticipating the intent and future states of the human could provide a more natural system. We also noticed that humans start adapting to the robot once they learn how it behaves. Therefore, introducing a multi-agent training scheme where both the simulated human and the robot are trained jointly [10] is interesting. Another direction could include making the RL exploration more efficient, as it currently still requires long training times, e.g., by leveraging state-agnostic priors [1] or using an intrinsic reward to incentivize contacts [16]. Lastly, our framework can potentially serve as a real world application framework to evaluate vision pipelines, e.g., for testing hand and object pose estimation estimation pipelines [15, 22] or segmentation models [6].

S1: Unseen Subjects									
		success (%)	mean accum time (s)			failure (%)			
			exec	plan	total	contact	drop	timeout	total
Sequential	OMG Planner [17] †	62.78	8.012	1.355	9.366	33.33	2.22	1.67	37.22
	Yang et al. [20] †	62.78	4.719	0.039	4.758	14.44	7.78	15.00	37.22
	GA-DDPG [18]	55.00	**6.791**	**0.136**	**6.927**	8.89	15.00	21.11	45.00
	Ours w/o ft.	68.15	7.151	0.164	7.314	6.85	12.96	12.04	31.85
	Ours	**75.00**	7.108	0.159	7.267	**5.00**	**12.59**	**7.41**	**25.00**
Simult.	GA-DDPG [18]	33.33	**4.261**	**0.132**	**4.393**	15.56	21.67	29.44	66.67
	Ours w/o ft.	62.78	5.695	0.164	5.859	5.93	17.59	13.70	37.22
	Ours	**73.33**	5.633	0.158	5.791	**5.56**	**15.37**	**5.74**	**26.67**

Table 3. **Unseen Subjects** Comparison of our method against various baselines from the HandoverSim benchmark [2] in the setting "S1: Unseen Subjects". The results of our method are averaged over 3 random seeds. †: both methods [17, 20] are evaluated with ground-truth states in [2] and thus are not directly comparable with ours.

S2: Unseen Handedness									
		success (%)	mean accum time (s)			failure (%)			
			exec	plan	total	contact	drop	timeout	total
Sequential	OMG Planner [17] †	62.78	8.275	1.481	9.755	30.56	3.89	2.78	37.22
	Yang et al. [20] †	62.50	4.808	0.034	4.843	16.11	10.56	10.83	37.50
	GA-DDPG [18]	55.00	7.145	**0.129**	7.274	**8.61**	17.78	18.61	45.00
	Ours w/o ft.	71.76	**7.045**	0.140	**7.185**	8.80	14.72	4.72	28.24
	Ours	**72.96**	7.101	0.144	7.245	11.29	**12.69**	**3.05**	**27.04**
Simult.	GA-DDPG [18]	28.33	**4.747**	**0.133**	**4.881**	9.17	34.44	28.06	71.67
	Ours w/o ft.	64.81	5.638	0.144	5.783	**8.24**	21.02	5.93	35.19
	Ours	**71.11**	5.771	0.150	5.921	10.00	**15.37**	**3.61**	**28.89**

Table 4. **Unseen Handedness.** Comparison of our method against various baselines from the HandoverSim benchmark [2] in the setting "S2: Unseen Handedness". The results of our method are averaged over 3 random seeds. †: both methods [17, 20] are evaluated with ground-truth states in [2] and are not directly comparable with ours.

S3: Unseen Objects									
		success (%)	mean accum time (s)			failure (%)			
			exec	plan	total	contact	drop	timeout	total
Sequential	OMG Planner [17] †	69.00	8.478	1.588	10.066	23.00	4.00	4.00	31.00
	Yang et al. [20] †	62.00	4.805	0.031	4.837	18.00	9.00	11.00	38.00
	GA-DDPG [18]	50.00	**7.305**	**0.135**	**7.440**	**5.00**	23.00	22.00	50.00
	Ours w/o ft.	76.33	7.410	0.151	7.565	9.33	10.67	**3.67**	23.67
	Ours	**79.67**	7.499	0.156	7.656	6.33	**10.33**	**3.67**	**20.33**
Simult.	GA-DDPG [18]	33.00	**4.948**	**0.123**	**5.071**	10.00	33.00	24.00	67.00
	Ours w/o ft.	72.00	6.242	0.168	6.410	7.33	13.67	7.00	28.00
	Ours	**75.67**	6.153	0.160	6.314	**5.00**	**13.33**	**6.00**	**24.33**

Table 5. **Unseen Object Evaluation.** Comparison of our method against baselines from the HandoverSim benchmark [2] in the setting "S3: Unseen Objects". The results of our method are averaged over 3 random seeds. †: both methods [17, 20] are evaluated with ground-truth states in [2] and are not directly comparable with ours.

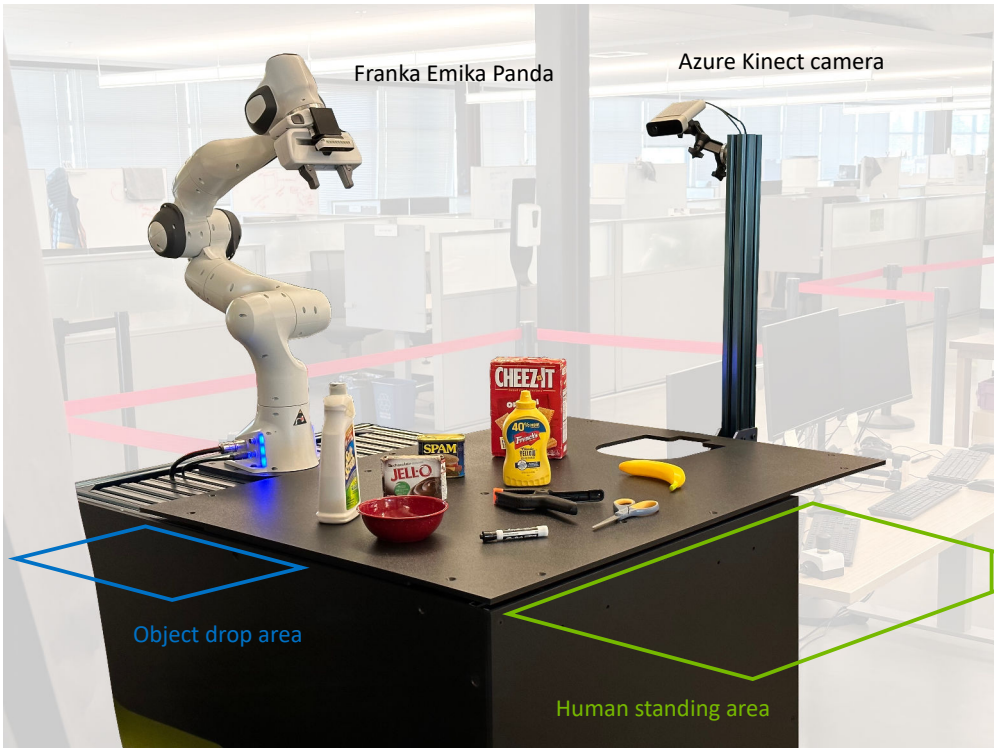

Figure 5. The setup of our real-world handover system. A Franka Emika Panda robot and an Azure Kinect camera are rigidly mounted on a table. The human participant will stand across the table (in the green area), pick up objects, and attempt handovers to the robot. The robot will drop the object in a designated area (blue) after retrieving it from the handover.

E. Sim-to-Real Transfer

E.1. System Setup

Fig. 5 shows the setup of our real-world handover system. The setup consists of a Franka Emika Panda robot and a Azure Kinect camera, both rigidly mounted to a table. The Azure Kinect is mounted externally to the robot with the extrinsics calibrated, and is perceiving the scenes from a third-person view with an RGB-D stream. The objects for handover are initially placed on the table. During handovers, the human participant will stand on the opposite side of the table (in the green area), pick up the objects, and attempt handovers to the robot. If the robot successfully retrieves the object, it will move the end effector to a drop-off area (blue) and drop the object into a box.

Since our policy expects a segmented point cloud at the input, we follow the perception pipeline used in [18, 20] to generate segmented point clouds for the hand and object. The Azure Kinect is launched to provide a continuous stream of RGB images and point clouds. We first use Azure Kinect's Body Tracking SDK to track the 3D location of the wrist joint of the handover hand. At each time frame, we crop a sub-point cloud around the tracked joint location which includes points on both the hand and the held object. We additionally run a 2D hand segmentation model on the RGB image and use it to label the hand points in the cropped point cloud. We treat all the points not labeled as hand as the object. Since our policies are trained for wrist camera views and we use an external camera in the real-world system, we need to additionally compensate for the view point change. We transform the segmented point cloud from the external camera's frame to the wrist camera's frame using the calibrated robot-camera extrinsics and forward kinematics. This way we can simulate the segmented point cloud input which the policy observes during the training in simulation. Note that this perception pipeline can induce sim-to-real gaps from several sources: (1) noises in the point clouds from real cameras, (2) noises from body tracking and hand segmentation errors, (3) the change in view points (i.e., from the wrist to external camera), and (4) unseen human behavior. To make our transfer policy more robust to diversity in human behvaior, we include human-object trajectories generated with D-Grasp [4] during training.

Compared to GA-DDPG [18], we adapt the control flow of the policy to explicitly incorporate the pre-grasp mechanism in our method. To control the motion of Franka, we follow the pipeline used in [18, 20]. Given a target end effector pose at a new time step, we use Riemannian Motion Policies (RMPs) [13] to generate a smooth trajectory for

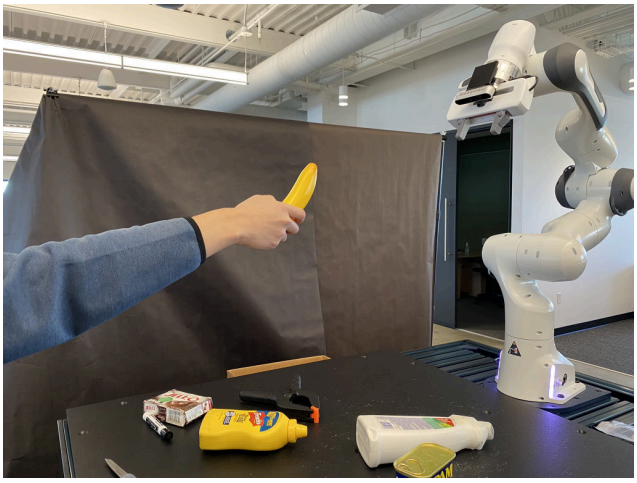

Figure 6. We conducted a pilot study by controlling the handover poses from the human subject.

the robot arm. We use libfranka to control the Franka arm to follow the trajectory. The robot will start moving only when a segmented point cloud is perceived. Once it decides to grasp, we will execute a predefined motion where the robot closes the gripper, lifts the end effector, moves to the drop-off area, and opens the gripper. The robot will return to a standby pose and remain in that state if no segmented point cloud is perceived or after it drops off the object.

E.2. Pilot Study

The goal of the pilot study is to provide a standardized benchmarking of the sim-to-real transfer. We instruct the participated subjects to follow a set of pre-determined handover poses when performing the handovers. We keep the instructed handover poses fixed for different methods to ensure a fair comparison.

Evaluation Protocol First, we select the following 10 objects from the YCB-Video dataset [19]:

- 011_banana
- 037_scissors
- 006_mustard_bottle
- 024_bowl
- 040_large_marker
- 003_cracker_box
- 052_extra_large_clamp
- 008_pudding_box
- 010_potted_meat_can
- 021_bleach_cleanser

For each object, we select 3 handover poses separately for the right and left hand, totaling 60 handover poses for both hands. The set of handover poses is selected to represent the handover task at different levels of difficulty: for each hand-object combination, we select one common handover pose ("pose 1"), one handover pose with the object held horizontally ("pose 2"), and one handover pose with severe hand occlusion by holding the object from the top ("pose 3"). Fig. 6 illustrates the setting where a subject holds an object in a controlled handover pose in front of the robot. Figs. 7 and 8 show the selected handover poses for the right and left hand respectively.

For each subject, we iterate through the 60 handover poses and evaluate each pose once. A handover is considered failed if (1) the robot pinches (or is about to pinch) the subject's hand (in which case the subject may evade the grasp), (2) the robot drops the object during the handover, or (3) the robot has reached an irrecoverable state (e.g., a locked arm due to joint limits). A handover is successful if the robot retrieves the object from the subject's hand and successfully move it to the drop-off area without incurring any failures. We evaluate the same handover poses on two methods: GA-DDPG [18] and ours. Therefore, each subject will perform 120 handover trials in total.

Results We conduct our pilot study with two subjects. Tabs. 6 and 7 present the results of subject 1 and 2 respectively. For each pose, we report whether the handover is successful ("succ."), and if so, the completion time of the handover ("time"), defined as the time span from the robot starts moving to the moment where the gripper lifts the object. For each object, we also report the mean success rate and mean completion time (see the "mean" column). For each subject, we present the results separately for the right hand (top), left hand (middle), and overall (bottom).

We observe a gap in the performance between the three selected handover poses. Both methods have a lower performance on "pose 2" and "pose 3" compared to "pose 1". For example, for subject 1, the overall success rate is $2/20$ for "pose 2" versus $16/20$ for "pose 1" for GA-DDPG [18], and $9/20$ for "pose 2" versus $19/20$ for "pose 1" for ours. This demonstrates the increase in challenges when the handover is conducted in uncommon poses ("pose 2"), where the robot needs to rotate the end effector, or when the human hand is blocking the robot's closest grasp point ("pose 3"), where the robot needs to diverge to avoid hand collision. However, our method is able to handle these cases better since the model is trained with diverse handover poses and supervision on hand collision avoidance (e.g., for "pose 3" on subject 1, an overall success rate of $13/20$ for ours versus $3/20$ for GA-DDPG [18]). In addition, our method achieves a higher overall success rate on both subjects (i.e., $41/60$ versus $21/60$ on subject 1 and $41/60$ versus $33/60$ on subject 2), demonstrating its efficacy over GA-DDPG [18]. However, our method still fails for 19 trials on each subject. This can be attributed to either the sim-to-real gaps discussed in Appendix E.1, an inherent failure of the policy on handling these cases, or an interplay of both.

E.3. User Evaluation

In contrast to the standardized evaluation in the pilot study, the goal of our user evaluation is to collect feedback from lay users with their own handover preferences.

Figure 7. The instructed handover poses for the **right hand** in the pilot study. We adopt 10 objects from the YCB-Video dataset [19] and pre-select 3 handover poses per object with varying handover difficulties, totaling 30 handover poses for the right hand.

Figure 8. The instructed handover poses for the **left hand** in the pilot study. Similar to the right hand poses in Fig. 7, we adopt the same 10 objects from the YCB-Video dataset [19] and pre-select 3 handover poses per object, totaling 30 handover poses for the left hand.

	subject 1: right hand															
	GA-DDPG [18]								Ours							
	pose 1		pose 2		pose 3		mean		pose 1		pose 2		pose 3		mean	
	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time
011_banana	✓	11.313	✓	10.790	×	–	2 / 3	**11.051**	✓	8.889	✓	18.535	✓	9.292	**3 / 3**	12.239
037_scissors	✓	9.571	×	–	×	–	1 / 3	9.571	✓	8.863	×	–	✓	10.216	**2 / 3**	**9.539**
006_mustard_bottle	×	–	×	–	×	–	0 / 3	–	✓	9.310	×	–	✓	10.792	**2 / 3**	10.051
024_bowl	✓	9.719	×	–	✓	10.760	**2 / 3**	**10.239**	✓	10.634	×	–	✓	13.333	**2 / 3**	11.983
040_large_marker	×	–	×	–	×	–	0 / 3	–	✓	9.605	×	–	×	–	**1 / 3**	9.605
003_cracker_box	✓	9.284	×	–	✓	10.440	**2 / 3**	**9.862**	✓	16.782	×	–	×	–	1 / 3	16.782
052_extra_large_clamp	×	–	×	–	×	–	0 / 3	–	✓	10.228	✓	19.855	✓	10.796	**3 / 3**	13.626
008_pudding_box	✓	9.565	×	–	✓	9.095	2 / 3	**9.330**	✓	10.583	✓	11.833	✓	11.387	**3 / 3**	11.267
010_potted_meat_can	✓	9.770	×	–	×	–	1 / 3	**9.770**	✓	13.500	×	–	✓	10.908	**2 / 3**	12.204
021_bleach_cleanser	✓	9.709	✓	12.367	×	–	2 / 3	**11.038**	✓	11.016	✓	11.572	✓	11.539	**3 / 3**	11.376
total	7 / 10	9.847	2 / 10	11.579	3 / 10	10.098	12 / 30	**10.199**	10 / 10	10.941	4 / 10	15.449	8 / 10	11.033	**22 / 30**	11.794

	subject 1: left hand															
	GA-DDPG [18]								Ours							
	pose 1		pose 2		pose 3		mean		pose 1		pose 2		pose 3		mean	
	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time
011_banana	✓	9.790	×	–	×	–	1 / 3	9.790	✓	9.180	✓	9.192	✓	8.931	**3 / 3**	**9.101**
037_scissors	✓	9.549	×	–	×	–	1 / 3	**9.549**	✓	8.973	✓	9.485	✓	10.254	**3 / 3**	9.571
006_mustard_bottle	✓	10.135	×	–	×	–	**1 / 3**	10.135	✓	8.499	×	–	×	–	**1 / 3**	**8.499**
024_bowl	✓	10.062	×	–	×	–	1 / 3	**10.062**	✓	14.572	×	–	✓	10.016	**2 / 3**	12.294
040_large_marker	×	–	×	–	×	–	0 / 3	–	✓	15.736	✓	14.601	✓	10.626	**3 / 3**	13.654
003_cracker_box	✓	10.231	×	–	×	–	**1 / 3**	**10.231**	✓	14.836	×	–	×	–	**1 / 3**	14.836
052_extra_large_clamp	✓	8.513	×	–	×	–	**1 / 3**	**8.513**	✓	10.186	×	–	×	–	**1 / 3**	10.186
008_pudding_box	✓	10.851	×	–	×	–	1 / 3	**10.851**	✓	17.965	✓	13.015	✓	20.966	**3 / 3**	17.315
010_potted_meat_can	✓	9.810	×	–	×	–	1 / 3	9.810	×	–	×	–	×	–	0 / 3	–
021_bleach_cleanser	✓	24.872	×	–	×	–	1 / 3	24.872	✓	15.028	✓	9.378	×	–	**2 / 3**	**12.203**
total	9 / 10	11.535	0 / 10	–	0 / 10	–	9 / 30	**11.535**	9 / 10	12.775	5 / 10	11.134	5 / 10	12.159	**19 / 30**	12.181

	subject 1: overall															
	GA-DDPG [18]								Ours							
	pose 1		pose 2		pose 3		mean		pose 1		pose 2		pose 3		mean	
	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time
011_banana	2 / 2	10.551	1 / 2	10.282	0 / 2	–	3 / 6	**10.631**	2 / 2	9.034	2 / 2	13.864	2 / 2	9.111	**6 / 6**	10.670
037_scissors	2 / 2	9.560	0 / 2	–	0 / 2	–	2 / 6	9.560	2 / 2	8.918	1 / 2	10.292	2 / 2	10.235	**5 / 6**	**9.558**
006_mustard_bottle	1 / 2	10.064	0 / 2	–	0 / 2	–	1 / 6	10.135	2 / 2	8.904	0 / 2	–	1 / 2	10.038	**3 / 6**	**9.534**
024_bowl	2 / 2	9.890	0 / 2	–	1 / 2	10.076	3 / 6	**10.180**	2 / 2	12.603	0 / 2	–	2 / 2	11.675	**4 / 6**	12.139
040_large_marker	0 / 2	–	0 / 2	–	0 / 2	–	0 / 6	–	2 / 2	12.670	1 / 2	14.268	1 / 2	10.481	**4 / 6**	12.642
003_cracker_box	2 / 2	9.757	0 / 2	–	1 / 2	10.021	**3 / 6**	**9.985**	2 / 2	15.809	0 / 2	–	0 / 2	–	**3 / 6**	15.809
052_extra_large_clamp	1 / 2	8.874	0 / 2	–	0 / 2	–	1 / 6	**8.513**	2 / 2	10.207	1 / 2	15.824	1 / 2	16.119	**4 / 6**	12.766
008_pudding_box	2 / 2	10.208	0 / 2	–	1 / 2	9.579	3 / 6	**9.837**	2 / 2	14.274	2 / 2	12.424	2 / 2	16.176	**6 / 6**	14.291
010_potted_meat_can	2 / 2	9.790	0 / 2	–	0 / 2	–	**2 / 6**	**9.790**	1 / 2	14.458	0 / 2	–	1 / 2	12.596	**2 / 6**	12.204
021_bleach_cleanser	2 / 2	17.291	1 / 2	11.867	0 / 2	–	3 / 6	15.650	2 / 2	13.022	2 / 2	10.475	1 / 2	16.574	**5 / 6**	**11.707**
total	16 / 20	10.797	2 / 20	11.579	3 / 20	10.098	21 / 60	**10.771**	19 / 20	11.810	9 / 20	13.052	13 / 20	11.466	**41 / 60**	11.973

Table 6. The pilot study results of **subject 1**. We present the results separately for the right-hand handovers (top), the left-hand handovers (middle), and overall (bottom). We report both the success rate ("succ.") and the completion time ("time"). Our method outperforms GA-DDPG [18] in the success rate.

Therefore, we do not constrain the users on how they hand over objects. Instead, we let them carry out in ways which they feel most comfortable (Fig. 9). Rather than objective metrics, we collect subjective feedback from the users via a questionnaire.

Evaluation Protocol We adopt the same 10 objects from the pilot study, and ask each user to hand over each object once with their right hand. We instruct the users to hand over objects "in any way they like". We compare the two methods (i.e., GA-DDPG [18] and ours) by repeating the same process, i.e., we instruct the user to hand over the 10 objects to one system first, followed by to the other system. We counterbalance the order of the two systems throughout the user evaluation to avoid bias. During their experiments, the users are asked to fill out a questionnaire with Likert-scale and open-ended questions to provide feedback after they interact with each system.

Results We conduct our user evaluation with 6 users (Fig. 9). The evaluation results are presented in Fig. 10 for GA-DDPG [18] and Fig. 11 for our method. Each fig-

	subject 2: right hand															
	GA-DDPG [18]								Ours							
	pose 1		pose 2		pose 3		mean		pose 1		pose 2		pose 3		mean	
	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time
011_banana	✓	13.599	✓	15.559	✓	9.333	**3 / 3**	12.830	✓	11.995	×	–	✓	8.046	2 / 3	**10.021**
037_scissors	✓	10.609	×	–	✓	9.769	**2 / 3**	10.189	✓	8.343	×	–	✓	8.583	**2 / 3**	**8.463**
006_mustard_bottle	✓	9.070	×	–	×	–	1 / 3	9.070	✓	8.787	×	–	✓	9.186	**2 / 3**	**8.987**
024_bowl	×	–	×	–	✓	9.085	**1 / 3**	**9.085**	×	–	×	–	✓	9.506	**1 / 3**	9.506
040_large_marker	✓	10.762	×	–	✓	8.748	**2 / 3**	**9.755**	✓	9.035	×	–	✓	17.609	**2 / 3**	13.322
003_cracker_box	×	–	×	–	×	–	0 / 3	–	✓	11.629	×	–	✓	11.563	**2 / 3**	11.596
052_extra_large_clamp	✓	9.319	×	–	✓	9.156	**2 / 3**	**9.237**	✓	9.788	✓	17.289	×	–	**2 / 3**	13.539
008_pudding_box	✓	10.402	×	–	✓	8.838	**2 / 3**	**9.620**	✓	8.803	×	–	✓	11.061	**2 / 3**	9.932
010_potted_meat_can	×	–	×	–	✓	12.886	1 / 3	12.886	✓	8.540	×	–	✓	8.610	**2 / 3**	**8.575**
021_bleach_cleanser	✓	10.571	✓	10.223	×	–	**2 / 3**	10.397	✓	8.241	×	–	✓	11.706	**2 / 3**	**9.974**
total	7 / 10	10.619	2 / 10	12.891	7 / 10	9.688	16 / 30	10.495	9 / 10	9.462	1 / 10	17.289	9 / 10	10.652	**19 / 30**	**10.438**

	subject 2: left hand															
	GA-DDPG [18]								Ours							
	pose 1		pose 2		pose 3		mean		pose 1		pose 2		pose 3		mean	
	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time
011_banana	✓	8.709	✓	9.186	✓	8.570	**3 / 3**	**8.821**	✓	9.096	✓	10.447	✓	7.965	**3 / 3**	9.169
037_scissors	✓	9.849	×	–	×	–	1 / 3	9.849	✓	8.587	✓	10.570	✓	8.370	**3 / 3**	**9.176**
006_mustard_bottle	×	–	×	–	✓	11.332	1 / 3	11.332	✓	8.078	×	–	✓	9.703	**2 / 3**	**8.891**
024_bowl	✓	9.183	×	–	✓	9.214	**2 / 3**	**9.199**	×	–	✓	9.958	✓	17.879	**2 / 3**	13.918
040_large_marker	✓	10.651	×	–	✓	9.881	2 / 3	**10.266**	✓	9.804	✓	14.292	✓	9.466	**3 / 3**	11.187
003_cracker_box	×	–	×	–	×	–	**0 / 3**	–	×	–	×	–	×	–	**0 / 3**	–
052_extra_large_clamp	✓	19.748	✓	9.566	✓	10.204	**3 / 3**	13.173	✓	19.253	✓	9.632	✓	9.552	**3 / 3**	**12.813**
008_pudding_box	✓	9.794	×	–	✓	9.236	**2 / 3**	9.515	✓	8.532	×	–	✓	8.590	**2 / 3**	**8.561**
010_potted_meat_can	✓	9.353	×	–	✓	9.240	**2 / 3**	9.296	✓	8.344	×	–	✓	9.277	**2 / 3**	**8.810**
021_bleach_cleanser	✓	10.140	×	–	×	–	1 / 3	10.140	✓	9.301	×	–	✓	10.288	**2 / 3**	**9.795**
total	8 / 10	10.928	2 / 10	9.376	7 / 10	9.668	17 / 30	**10.227**	8 / 10	10.125	5 / 10	10.980	9 / 10	10.121	**22 / 30**	10.318

	subject 2: overall															
	GA-DDPG [18]								Ours							
	pose 1		pose 2		pose 3		mean		pose 1		pose 2		pose 3		mean	
	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time
011_banana	2 / 2	11.154	2 / 2	12.372	2 / 2	8.951	**6 / 6**	10.826	2 / 2	10.546	1 / 2	10.408	2 / 2	8.006	5 / 6	**9.510**
037_scissors	2 / 2	10.229	0 / 2	–	1 / 2	9.243	3 / 6	10.076	2 / 2	8.465	1 / 2	9.638	2 / 2	8.477	**5 / 6**	**8.891**
006_mustard_bottle	1 / 2	9.476	0 / 2	–	1 / 2	10.479	2 / 6	10.201	2 / 2	8.433	0 / 2	–	2 / 2	9.445	**4 / 6**	**8.939**
024_bowl	1 / 2	11.101	0 / 2	–	2 / 2	9.149	**3 / 6**	**9.161**	0 / 2	–	1 / 2	9.720	2 / 2	13.692	**3 / 6**	12.447
040_large_marker	2 / 2	10.707	0 / 2	–	2 / 2	9.314	4 / 6	**10.010**	2 / 2	9.419	1 / 2	15.882	2 / 2	13.537	**5 / 6**	12.041
003_cracker_box	0 / 2	–	0 / 2	–	0 / 2	–	0 / 6	–	1 / 2	11.925	0 / 2	–	1 / 2	12.989	**2 / 6**	11.596
052_extra_large_clamp	2 / 2	14.533	1 / 2	9.562	2 / 2	9.680	**5 / 6**	**11.599**	2 / 2	14.521	2 / 2	13.460	1 / 2	36.778	**5 / 6**	13.103
008_pudding_box	2 / 2	10.098	0 / 2	–	2 / 2	9.037	**4 / 6**	9.567	2 / 2	8.668	0 / 2	–	2 / 2	9.825	**4 / 6**	**9.247**
010_potted_meat_can	1 / 2	9.919	0 / 2	–	2 / 2	11.063	3 / 6	10.493	2 / 2	8.442	0 / 2	–	2 / 2	8.943	**4 / 6**	**8.693**
021_bleach_cleanser	2 / 2	10.355	1 / 2	12.233	0 / 2	–	3 / 6	10.311	2 / 2	8.771	0 / 2	–	2 / 2	10.997	**4 / 6**	**9.884**
total	15 / 20	10.784	4 / 20	11.134	14 / 20	9.678	33 / 60	**10.357**	17 / 20	9.774	6 / 20	12.031	18 / 20	10.387	**41 / 60**	10.373

Table 7. The pilot study results of **subject 2**. We present the results separately for the right-hand handovers (top), the left-hand handovers (middle), and overall (bottom). We report both the success rate ("succ.") and the completion time ("time"). Our method outperforms GA-DDPG [18] in the success rate.

ure shows the user's ranking with the statements queried in the questionnaire. For each statement, a user can rank their agreement level with one of the five options: "Strongly disagree" (1), "Disagree" (2), "Neither agree or disagree" (3), "Agree" (4), and "Strongly agree" (5) (see the color codes in Figs. 10 and 11). The length of each color bar denotes the count of the users. For each method, the statements are further grouped into two subfigures, where a higher agreement score indicates a better performance (top), and a lower agreement score indicates a better performance (bottom).

Overall, our method receives higher agreement scores over GA-DDPG [18] for the statements "*The robot could hold the object stably once taking it over from my hand.*" (i.e., (5,4,4,4,4,3) versus (5,4,3,3,3,2)) and "*The robot was able to adapt its behavior to different ways of how I held the object for handover.*" (i.e., (5,5,5,4,4,3) versus (5,4,3,3,3,2)). This is congruent with our simulation evaluation results that our method can grasp objects more robustly by finding good pre-grasp poses around the object. This was also reflected in participants' comments. One said

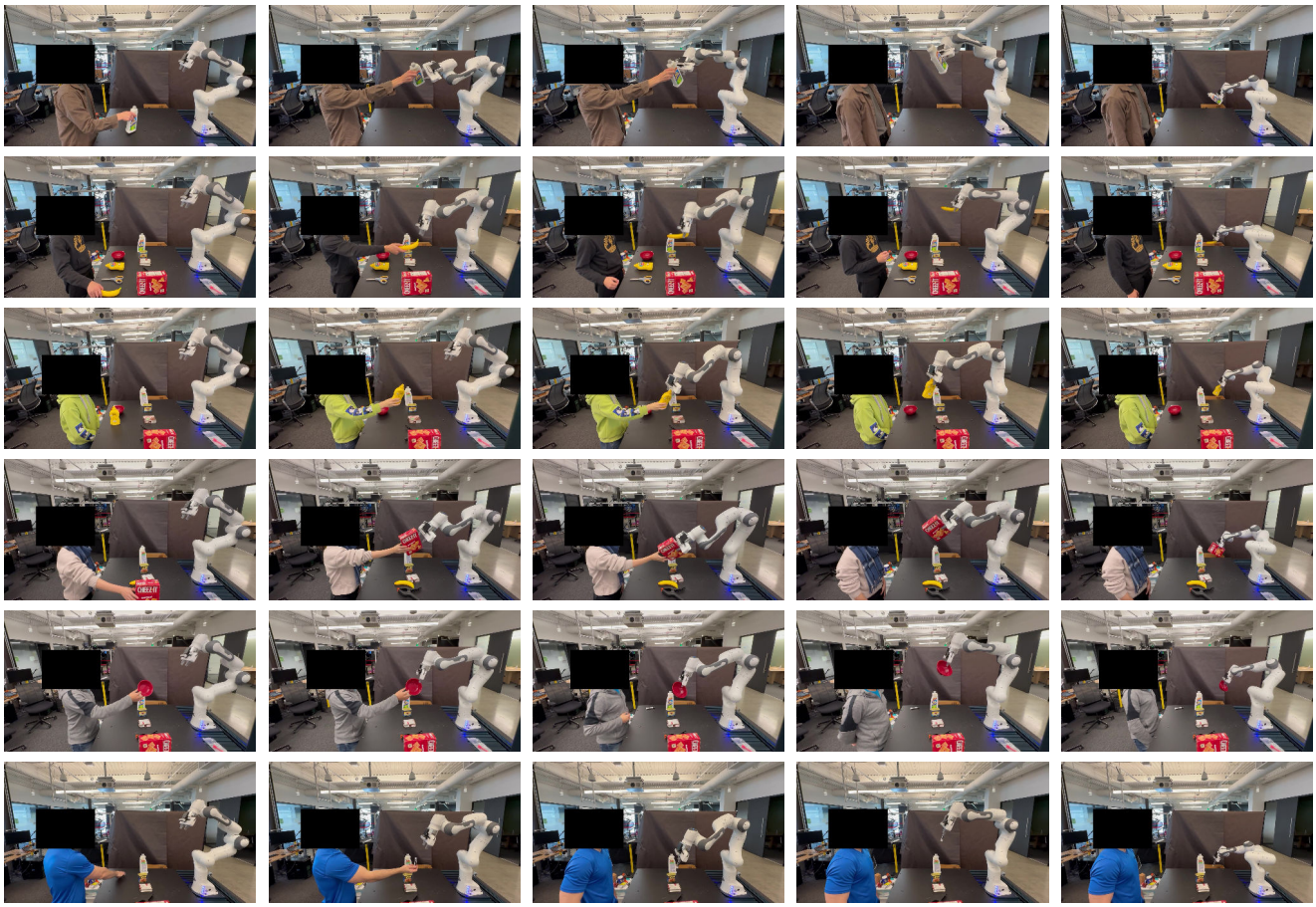

Figure 9. We conduct a user evaluation with 6 users by allowing the users to perform handovers freely. The images depict sequences (from left to right) of different users handing over a variety of objects to the robot.

our method "*tends to explore more diverse grasp*", "*was much better at aligning the grasp*" and "*adjusts behavior for different objects in different poses*" when compared with GA-DDPG. One pointed out that sometimes GA-DDPG "*grasped from the tip of the object*". The interpretability of the robot's motion was also acknowledged by their comments, *e.g.*, it "*[was] safe and interpretable at all times*" and "*felt like we understood each other*". Surprisingly, the users favor GA-DDPG [18] more when it comes to safety related metrics, e.g., for the statement "*I felt safe while the robot was moving.*" ((5,4,3,3,2,2) for ours versus (5,5,4,4,3,3) for GA-DDPG [18]) and "*The robot was likely to pinch my hand.*" ((1,2,2,3,4,4) for ours versus (1,2,2,2,2,3) for GA-DDPG [18]). This can be attributed to GA-DDPG's tendency to grasp from the grasp points closest to the robot, and hence it often keeps a safe distance from the human hand. For our method, several users felt the robot hand pushing too much during grasping. One said it was "*flexible in grasp selection, but may be too close to my finger*". Another said "*the forward movement ... put the gripper fairly close to me*". This can potentially be addressed by incorporating force feedback in the grasping motion as well as taking gripper hand distance into account during training. The majority of participants agreed that the timing of our method is more appropriate, commenting the "*handover time was pretty seamless*" and "*didn't have to wait too long*".

Although the main objective in the user study was to let users interact freely with the system in a non-standardized manner, we additionally evaluate the user study quantitatively. We report the success rate and approach time (*i.e.*, from the robot starting to move to grasp completion). Our method still compares favorably to GA-DDPG with a higher success rate (88.9% *vs.* 80.0%) and a shorter approach time (6.40 ± 2.27s *vs.* 7.48 ± 2.64s). The better timing was noted by the majority of participants, who commented that the "*handover time was pretty seamless*" and "*didn't have to wait too long*". Interestingly, we observed in our user study that natural H2R handovers are less susceptible to grasping failures, since the human partner would often help by agilely adjusting the object pose in the last mile to ensure a successful grasp.

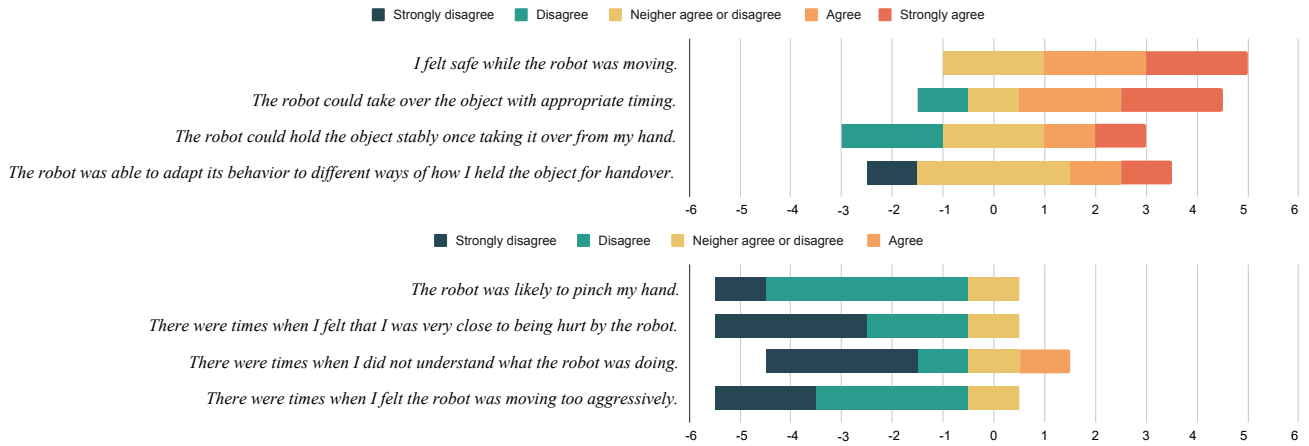

Figure 10. User's ranking with each statement for **GA-DDPG [18]** in the user evaluation. Each color denotes a different a degree of agreement. The length of the bar denotes the count of the users. For each bar, the center count of "Neither agree or disagree" is aligned with 0 in the horizontal axis. In the top figure, a higher agreement score (orange) indicates a better performance, while in the bottom figure, a lower agreement score (green) indicates a better performance.

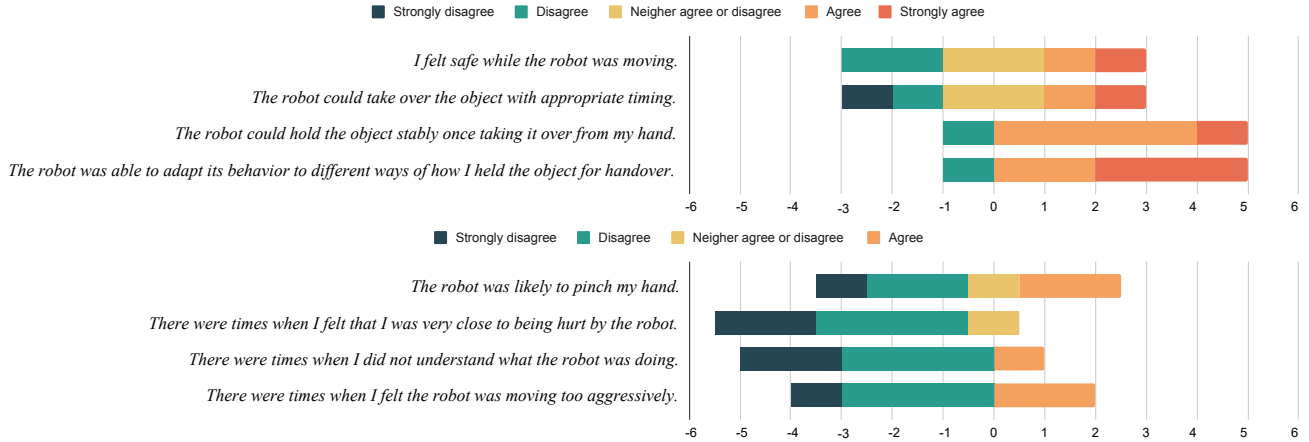

Figure 11. User's ranking with each statement for **our method** in the user evaluation. Each color denotes a different a degree of agreement. The length of the bar denotes the count of the users. For each bar, the center count of "Neither agree or disagree" is aligned with 0 in the horizontal axis. In the top figure, a higher agreement score (orange) indicates a better performance, while in the bottom figure, a lower agreement score (green) indicates a better performance.

References

[1] Marco Bagatella, Sammy Christen, and Otmar Hilliges. SFP: State-free priors for exploration in off-policy reinforcement learning. *Transactions on Machine Learning Research*, 2022. 4

[2] 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 *IEEE International Conference on Robotics and Automation (ICRA)*, 2022. 1, 3, 5

[3] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S. Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, Jan Kautz, and Dieter Fox. DexYCB: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[4] Sammy Christen, Muhammed Kocabas, Emre Aksan, Jemin Hwangbo, Jie Song, and Otmar Hilliges. D-grasp: Physically plausible dynamic grasp synthesis for hand-object interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 6

[5] Clemens Eppner, Arsalan Mousavian, and Dieter Fox. ACRONYM: A large-scale grasp dataset based on simulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021. 3

[6] Zicong Fan, Adrian Spurr, Muhammed Kocabas, Siyu Tang, Michael Black, and Otmar Hilliges. Learning to disam-

biguate strongly interacting hands via probabilistic per-pixel part segmentation. In *International Conference on 3D Vision (3DV)*, 2021. 4

[7] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning (ICML)*, 2018. 3

[8] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014. 4

[9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *International Conference on Learning Representations (ICLR)*, 2015. 3

[10] Thomas Langerak, Sammy Christen, Mert Albaba, Christoph Gebhardt, and Otmar Hilliges. Marlui: Multi-agent reinforcement learning for goal-agnostic adaptive uis. *arXiv preprint arXiv:2209.12660*, 2022. 4

[11] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-DOF GraspNet: Variational grasp generation for object manipulation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 3

[12] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 3

[13] Nathan D. Ratliff, Jan Issac, Daniel Kappler, Stan Birchfield, and Dieter Fox. Riemannian motion policies. *arXiv preprint arXiv:1801.02854*, 2018. 6

[14] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011. 2

[15] Tze Ho Elden Tse, Kwang In Kim, Ales Leonardis, and Hyung Jin Chang. Collaborative learning for hand and object reconstruction with attention-guided graph convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1664–1674, 2022. 4

[16] Nikola Vulin, Sammy Christen, Stefan Stevšić, and Otmar Hilliges. Improved learning of robot manipulation tasks via tactile intrinsic motivation. *IEEE Robotics and Automation Letters (RA-L)*, 2021. 4

[17] 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. 3, 5

[18] 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 (CoRL)*, 2021. 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13

[19] Yu Xiang, Tanner Schmidt, Venkatraman Nafurayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In *Proceedings of Robotics: Science and Systems (RSS)*, 2018. 7, 8, 9

[20] Wei Yang, Chris Paxton, Arsalan Mousavian, Yu-Wei Chao, Maya Cakmak, and Dieter Fox. Reactive human-to-robot handovers of arbitrary objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021. 5, 6

[21] Lin Yen-Chen, Pete Florence, Jonathan T. Barron, Tsung-Yi Lin, Alberto Rodriguez, and Phillip Isola. NeRF-Supervision: Learning dense object descriptors from neural radiance fields. In *IEEE Conference on Robotics and Automation (ICRA)*, 2022. 4

[22] Andrea Ziani, Zicong Fan, Muhammed Kocabas, Sammy Christen, and Otmar Hilliges. Tempclr: Reconstructing hands via time-coherent contrastive learning. In *International Conference on 3D Vision (3DV)*, 2022. 4