Appendix

A Implementation Details

This section describes the network architectures of each model used in the main paper.

Co-policy: Our co-policy model π_{co} consists of an actor and a critic network. Both networks are multilayer perceptrons with 4 fully-connected layers of size 128 each. The output layer of the critic is (128, 1). The output feature of the actor network is further processed by a fully-connected layer into a normal distribution with the mean and standard deviation for each action dimension. The output action is then branched into the human action a^H and robot action a^R. These actions will be sampled from the distribution during training and the one with maximum probability will be selected during testing.

Strategy recognition network: The strategy recognition network ψ is a multilayer perceptrons with 2 fully-connected layers with the output size of 2 (The strategy code in our experiments is 2 float numbers in the range $(-1.0, 1.0)$). To regularize the output strategy prediction, we use a *tanh* function to process the output prediction.

Discriminator network: The discriminator networks D is a multilayer perceptrons with 3 fully-connected layers $(64, 64, 1)$, followed by a *tanh* function to regularize the output within $(-1.0, 1.0)$.

B Training Details

B.1 GAIL

The standard GAIL objective detailed in Equation 5 typically uses a sigmoid cross-entropy loss function. However, we empirically find this loss will cause the vanishing of gradients because of the sigmoid function, especially in the high-dimensional manipulation tasks (HR-Handover and HR-SeqManip). In this work, we use the loss function proposed by least-squares GAN (LSGAN)[50] for the high-dimensional manipulation tasks. The training objective of the discriminator D is:

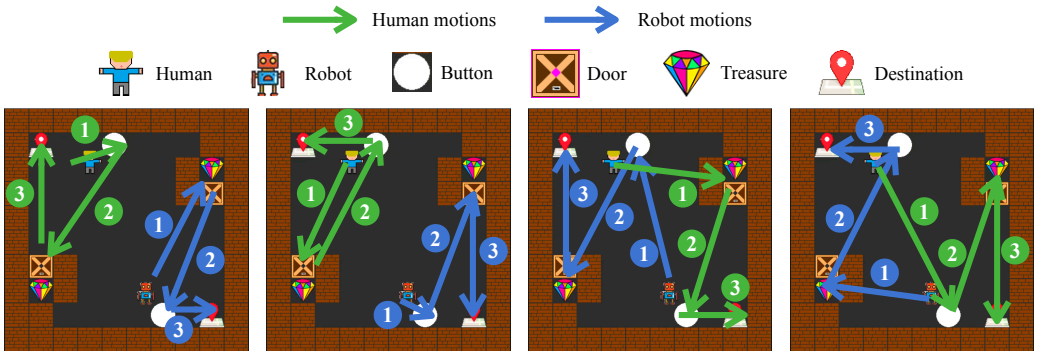
$$\min_D L(D) = \mathbb{E}_{\boldsymbol{x}\sim\rho(\pi_{E_1},\pi_{E_2})}[(D(\boldsymbol{x})-1)^2] + \mathbb{E}_{\boldsymbol{y}\sim\rho(\pi_{co})}[(D(\boldsymbol{y})+1)^2], \tag{5}$$

During the training, the strategy code z is randomly sampled. The original implementation of infoGAIL[25] uses a uniformly random sampling over code space. However, we empirically found that random sampling might cause unstable results between different training seeds. Therefore, we use a grid-based sampling on our strategy code. The code space $(-1.0, 1.0) \times (-1.0, 1.0)$ is first discretized into 25 grid points. During the training, each grid point will be selected in a cycle order and a uniformly sampled noise ($[-0.2, 0.2]$ for each code dimension in our experiments) will be added to the selected grid point. This setting could make sure that the replay buffer covers most of the strategies of the learned policy and improve the stability of the training results across different training seeds.

In the final learning objective of Co-GAIL (Equation 4), two hyperparameters λ_1 and λ_2 are introduced. For all three experiments, we used $\lambda_1 = \lambda_2 = 0.1$. We found that large λ_1 and λ_2 would cause unstable training procedures of the GAIL algorithm, where the objectives that measure the reconstruction error of the strategy code and human actions in Equation 4 dominate the loss function and cause the model to neglect the first GAIL learning objective. We empirically found $\lambda_1 = \lambda_2 = 0.1$ achieve the most stable learning performance in all three task environments.

B.2 PPO

The co-policy model is trained by the PPO[46]. For all three experiments, we use the learning rate $3e-4$ with a linear decay based on the percentage of the completed training epochs over the total number of episodes. The size of the replay buffer is 6000 for the first two experiments (2D-Fetch-Quest, HR-Handover) and 10000 for the HR-SeqManip. The rest of the hyperparameters are the same as the default PPO algorithm[46].



(a) Strategy 1 (b) Strategy 2 (c) Strategy 3 (d) Strategy 4

Figure 5: **2D-Fetch-Quest.** In the first row, the meaning of each icon in the game is illustrated. The goal is both human and robot fetch a treasure and reach the closest destination. There are four strategies to tackle the game. In strategy 1, the human will first help the robot to get the treasure. In the second strategy, the human will first get its treasure. In strategies 3 and 4, the human and robot will switch their role and move to the other side of the map to complete the collaboration.

C Experimental Details

C.1 2D-Fetch-Quest

Task details. This environment is adapted from the Fetch-Quest collaborative game in Super Mario Party. The human and robot agents need to work together to fetch the treasure and reach the destination. In Figure 5, we illustrate the meaning of each icon in the first row. At the beginning of the game, two treasures are locked in the rooms located in two corners of the map. For each agent, the only way to fetch a treasure is to let its collaborator press the button beside the room for it. After the collaborator press the button, the agent could enter the room and get the treasure. To win the game, both agents should get a treasure and reach the closest destination. Therefore, the agent who has already get the treasure will help the other agent to fetch its treasure. There are four types of strategies to tackle this game. As is shown in Figure 5, strategy 1 and 2 is different in the order of who gets the treasure first. In strategies 3 and 4, the human and robot will switch their roles and go to the other side of the map to complete the collaboration.

State and action space. The continuous action space of both human and robot agents is a 2D translation $(\Delta x, \Delta y)$ in the map, where $\Delta x \in [-1.0, 1.0], \Delta y \in [-1.0, 1.0]$. The size of the map is 8.0×8.0. The input state of the co-policy model consists of the 2D locations of the human and robot agent, the positions of the two treasures in the map, and two binary values that indicate whether each door of the room is opened or not.

Demonstrations. The human-human collaboration demonstrations are collected by two users with two joysticks on an Xbox gaming controller. During training, 60 human-human demonstrations are given. The average step size of the given human-human demonstration is 145. No external reward signals are given during the training.

Training and evaluation details. For each method, the training takes 1000 episodes and each episode contains 6000 environment steps. The checkpoint is saved every 30 training episodes. Each method is trained with three different seeds: 300, 400, 500 in our experiments. The maximum of the mean success rate among all checkpoints across three seeds and its 95% confidence interval are reported for both replay evaluation and interpolation in Table. 1 and Table. 2 for each method. During the replay evaluation, 60 testing demonstrations (not included in the training set) are used to test each method. During the interpolation, 100 uniformly sampled strategies are used to generate different behaviors for each method.

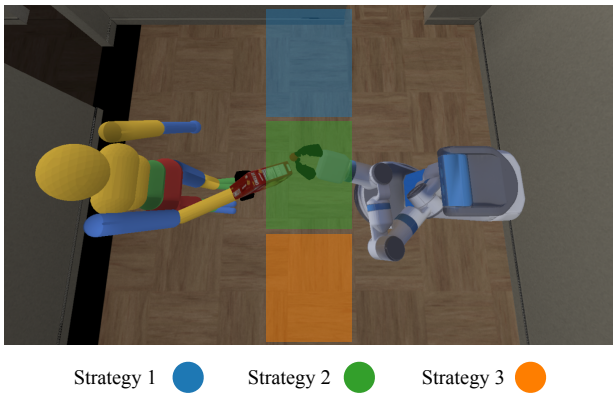


Figure 6: **HR-Handover.** We classified handover regions into three categories for post-hoc analysis. The classification is based on the relative location of where the handover happens in front of human. From left to the right, they are strategy 1, 2 and 3.

C.2 HR-Handover

Task details. In this environment, the human agent will first fetch the object in front of it and handover the object to the robot. A successful completion of the task happens only when the robot holding the object and the distance between the end effectors of the human and robot is larger than 10 centimeter. The diversity of the collaborative behaviors in this task is the position of the object when handover happens. As shown in Figure 6, the handover can happen in a wide range of locations. We post-hoc categorize these handover locations to left, right, or center only for analysis purposes.

State and action space. The continuous action space of both human and robot agents is the 6D pose changes $(\Delta x, \Delta y, \Delta z, \Delta R_x, \Delta R_y, \Delta R_z)$ of the end-effector relative to the previous time step and a float value $g \in [-1.0, 1.0]$ describing the gripper state (-1 refers to gripper fully closed and 1 refers to gripper fully opened), where $\Delta x, \Delta y, \Delta z \in [-1.0, 1.0]$ centimeters and $\Delta R_x, \Delta R_y, \Delta R_z \in [-1.0, 1.0]$ degrees. The environment state of this task includes the 6D pose of the end-effectors of both human and robot in their base frame, the relative position between the object to the end-effector of each agent and the orientation of the object.

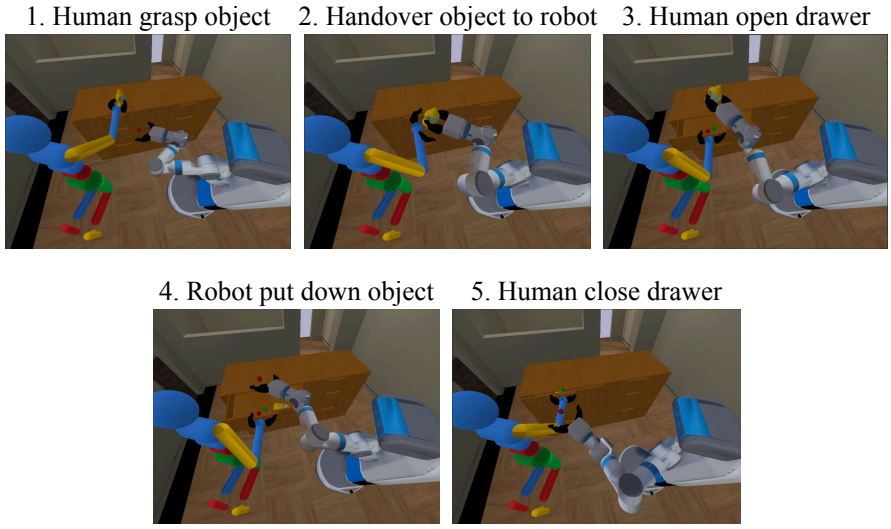
Demonstrations. The human-human collaboration demonstrations are collected by two users with the phone teleportation system RoboTurk[47, 48]. Without haptic feedback, it is hard for the human agent to identify whether the robot has held the object firmly or not. Therefore, we simplify the control system that after the robot holds the object, the gripper of the human agent will automatically open. During training, 160 human-human demonstrations with an average step size 211 is used to train the model.

Training and evaluation details. For each method, the training takes 300 episodes with 6000 environment steps each episode. The checkpoints are saved every 10 training episodes. The evaluation settings are the same as the 2D-Fetch-Quest. To visualize the mapping between the latent space to different strategies (Figure. 4), we classified the strategies into three categories based on the position where handover happens from the top-down view (Figure. 6).

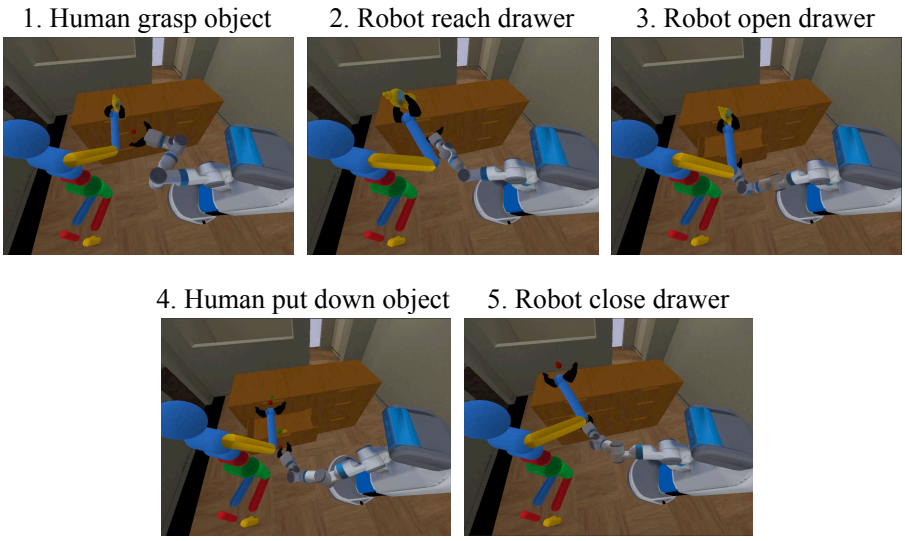
Additional results. Here we show an additional interpolation result on the handover positions of each method in Figure. 8. Our proposed method Co-GAIL successfully covers the data distribution while the baseline methods focus on a partial region.

C.3 HR-SeqManip

In this environment, the human agent will first fetch the object on the cabinet. The goal is to put the object into the drawer. As is shown in Figure. 7, there are two strategies to complete the task. In strategy 1, the human will first handover the object to the robot and then open the drawer. After the robot putting the object into the drawer, the human will close the drawer to complete the task. In strategy 2, the robot will open the drawer for the human to put the object into.



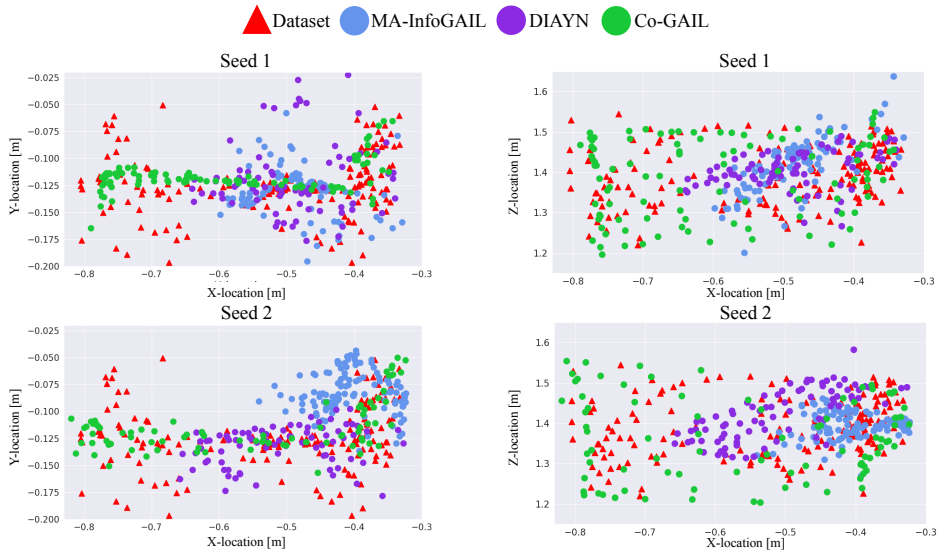
(a) strategy 1



(b) strategy 2

Figure 7: **HR-SeqManip.** Here we illustrate two types of strategies to tackle the task. In strategy 1, the human will first handover the object to the robot and then open the drawer. After the robot puts the object into the drawer, human will close it. In strategy 2, the robot will open the drawer for the human. After human place the object, the robot will close the drawer.

The action space of both human and robot agents is the same as the HR-Handover task. The environment state has an additional relative position between the handle of the drawer to the end-effector of each agent. During training, 120 human-human demonstrations with an average step size 404 is used to train the model. For each method, the training takes 1000 episodes with 10000 environment steps each episode. The checkpoints are saved every 30 training episodes. The evaluation settings are the same as the 2D-Fetch-Quest and HR-Handover.



(a) Handover positions (X-Y location)

(b) Handover positions (X-Z location)

Figure 8: **Diversity encoded in latent space.** The figure supplements Figure 4(a) by also showing the X-Z location of the handover positions. Co-GAIL successfully cover the data distribution while the baseline mathods focus on a partial region.

D Real-human Evaluation Details

Evaluation protocol. To test whether each method can handle diverse human behavior, we invite four real-human operators to conduct the real-human evaluation. We select three methods that have the best performance in the replay evaluation (*MA-GAIL*, *MA-InfoGAIL*, *Co-GAIL*) to compare in this experiment. The *BC* baseline is also included for reference. The checkpoint with the highest replay evaluation success rate of each method will be first loaded. Each human operator will perform 20 rounds of evaluation. In each round, the robot will be controlled by the models learned by each method in random order. The human operator does not know which model is it currently working with. To further make sure the comparison between the methods is fair, human operators are suggested to maintain similar motions for different trials in the same round. The final average success rate over 20 rounds of each method is reported in Table. 3.

Controller. For the 2D-Fetch-Quest, the human operator will use one joystick to control the human agents in the game. For the HR-Handover and HR-SeqManip, the human operator will use the phone teleportation system RoboTurk[47, 48] to control the 6D pose of the end-effector of the humanoid.

E Applicability to real robot

Due to COVID-19, we were not able to explore the potential of deploying our system onto a real-world robot platform. Our plan for simulation-to-real world transfer involves two steps. First, our models currently operate on low-dimensional state space including (1) robot and human proprioceptive information and (2) object pose information. In the real world, we could easily obtain robot proprioceptive information through the robot's internal APIs and match that of the simulator. We could obtain human's hand information through an off-the-shelf hand tracker [51], and the object pose through a category-agnostic 6DoF pose tracking system [52]. This allows us to match the state space between the simulated environment and the real world. Second, as indicated by prior works, there are likely mismatches between the simulated and real-world physics. We plan to address such gap in physical dynamics through domain randomization, i.e., we will expose our agent with wide ranges of possible physical dynamics settings during training. Past work [49] showed that such training strategy could largely alleviate the domain shift in dynamics, allowing the agent to adapt to varying dynamics in the physical world.