Robotic Learning From Advisory and Adversarial Interactions Using a Soft Wrist

Masashi Hamaya, Kazutoshi Tanaka, Yoshiya Shibata, Felix von Drigalski, Chisato Nakashima, and Yoshihisa Ijiri

Abstract—In this letter, we developed a novel learning framework from physical human-robot interactions. Owing to human domain knowledge, such interactions can be useful for facilitating learning. However, applying numerous interactions for training data might place a burden on human users, particularly in real-world applications. To address this problem, we propose formulating this as a model-based reinforcement learning problem to reduce errors during training and increase robustness. Our key idea is to develop 1) an advisory and adversarial interaction strategy and 2) a human-robot interaction model to predict each behavior. In the advisory and adversarial interactions, a human guides and disturbs the robot when it moves in the wrong and correct directions, respectively. Meanwhile, the robot tries to achieve its goal in conjunction with predicting the human’s behaviors using the interaction model. To verify the proposed method, we conducted peg-in-hole experiments in a simulation and real-robot environment with human participants and a robot, which has an underactuated soft wrist module. The experimental results showed that our proposed method had smaller position errors during training and a higher number of successes than the baselines without any interactions and with random interactions.

Index Terms—Physical human-robot interaction, reinforcement learning for robotic control, soft robot applications.

I. INTRODUCTION

Learning technologies have demonstrated significant potential in various robotic manipulation applications such as, but not limited to, autonomous assembles [1] and collaborations with humans [2]. Owing to more expressive neural network architectures, recent developments have allowed robots to complete more complex tasks that could not be solved using manually designed controllers.

Meanwhile, two serious problems remain when applying them to a real robot environment: learning inefficiencies and vulnerability amidst unknown environments. First, as the tasks or learning architecture become more complex, the required amount of training data increases. Furthermore, real-world environments always introduce model uncertainty and disturbances, to which learning or learned policies are frequently vulnerable [3]. Such complex learning settings and vulnerable learned policies might increase the risk of malfunction leading to catastrophic failures, such as generating large forces that damage the environment. For this reason, we need to explore a simpler way of efficiently obtaining such robust policies.

To satisfy both robustness and efficiency, adversarial learning techniques can be adopted, which provides disturbances that prevent an agent from completing tasks [3]. The agent can learn a policy to overcome these disturbances, resulting in enhanced robustness. Although many approaches have been proposed, applying them to real-robot environments is challenging as additional engineering specifications are required [4].

In contrast to previous approaches, we aim to exploit the human domain knowledge for adversarial learning. Relying on knowledge, a complex architecture to learn adversarial policies will not be required. Duan et al. also proposed the use of human interactions for adversarial learning during a simulation [5]. However, for more practical applications in real robot environments, the remaining requirement should be addressed. Duan et al. addressed only one-step adversarial interactions for each episode [5], while we tackle multi-step interactions for various real-world applications. Under such conditions including
multi-step interactions, we need to find more sample-efficient methods because the dynamics will be more complex.

To deal with these problems, we propose a novel learning framework from human interactions that is more applicable to real-world settings (see Fig. 1). To alleviate human effort during interactions, we formulated this problem as a model-based reinforcement learning task. Our key idea is to develop 1) an advisory and adversarial interaction strategy with the aim of reducing errors during training and increasing robustness and 2) a human-robot interaction model to predict human and robot behaviors. First, the human provides advisory interactions when the robot moves in a wrong direction. This help the robot to reduce the errors during the task. Then, the human provides adversarial interactions when the robot moves in the correct direction resulting in obtaining robustness. The robot learns a policy to achieve its goal using the model while minimizing and maximizing human efforts in the advisory and adversarial interactions, respectively. However, different contexts might exist for the advisory and adversarial interactions resulting in catastrophic forgetting. To share the experiences of both interactions effectively, we also apply progressive neural networks [6]. These networks can leverage prior knowledge through lateral connections. Finally, based on the interaction model, we employed a model-based reinforcement learning algorithm using an ensemble of deep neural network models [7].

Our contributions can be summarized as follows:

- We propose a novel learning framework from human interactions to reduce errors during training and increase robustness.
- We develop advisory and adversarial interaction strategies, and a human-robot interaction model to predict their behaviors.

To verify the proposed method, we conducted a simulation and a real robot experiment with human participants. Although our method has the potential to apply various robotic tasks, we focus on peg-in-hole tasks, which are difficult owing to contact-richness and low tolerance between the peg and hole. For the experiment, we exploited a robot with a soft wrist [8] which can safely make contact with the environment. The experimental results showed that our proposed method demonstrates smaller position errors in the training environment and a larger number of successes in unknown environments when compared to a method without human interactions and with random ones.

The remainder of this letter is organized as follows. Section II presents previous works related to our research. We introduce our proposed method in Section III. Sections IV and V describe the simulation and real robot experiments, whereas Section VI discusses our results and Section VII provides some concluding remarks.

II. RELATED WORKS

In this section, we introduce related studies mainly on, 1) adversarial learning, 2) learning from human-robot interactions, and 3) soft robots.

A. Adversarial Learning

Many researchers have investigated adversarial learning approaches [3] because deep reinforcement learning is typically vulnerable to adversarial perturbations, which causes a malfunction in the learning processes [9]. Minimax approaches are often used for adversarial learning [10], [11], Pint et al. formulated a robust adversarial reinforcement learning method that learned protagonist and antagonist policies [10]. Other approaches have also been proposed to provide perturbations with states based on gradients of the cost function [12], [13]. Although these approaches have demonstrated robustness, most of them have only been evaluated through simulations. Pint et al. conducted adversarial learning in real-world environments where two robots provided perturbations to disturb their grasping [4]. However, this method requires a higher engineering cost to design the planning for both robots. In this study, we aim to exploit human interactions for easier adversarial learning in real-world applications.

B. Learning From Human Interactions

Exploiting human domain knowledge is useful for designing reward functions or policies. Recent studies have used human feedback and corrections to learn reward functions or policy parameters [14]–[17]. Physical human-robot interactions have also been adopted to reward learning [18].

Duan et al. developed a robot learning framework through human-robot interactions, where humans disturb the robot’s grasping [5]. Although this method demonstrated robust grasping strategies, it was not evaluated in a real robot environment, probably owing to the following problems. First, because the method applies self-supervised learning, a large number of interactions place a heavy burden on users. Second, Duan et al. addressed only one-step adversarial interaction for grasping [5], which is difficult to apply to other tasks requiring multi-step (continuous) interactions such as assembly tasks. Our method can handle continuous advisory and adversarial interactions using model-based reinforcement learning while maintaining the sample efficiency.

C. Soft Robots

Soft robots, which include compliant components such as springs and silicon, have recently attracted significant attention [19], [20] because the softness allows contact rich interactions without high-precision or -frequency force sensors and controllers. These robots have been applied to assembly [21], [22], grasping as well as pre-grasping tasks [23]–[25]. As soft robots are also suitable for physical human-robot interactions, learning approaches for the interactions have been proposed for assistive robots [26]–[28]. We used a robot with an underactuated soft wrist module composed of springs [8], [29]. The soft wrist is useful for our problem setting, as we tackle contact-rich tasks and physical interactions simultaneously. Our method could also be applied to existing soft robots.

III. PROPOSED METHOD

In this section, we introduce our proposed method. The human-robot interaction model for predicting human and robot behaviors is first developed, after which the dual interaction strategy for advisory and adversarial interactions. Finally, we apply a model-based reinforcement learning approach.
A. Problem Formulation

We formulated this task as model-based reinforcement learning. We consider a setting in which human and robot physically interact with each other. Given action commands, the robot attempts to achieve the desired task while the human provides advisory and adversarial external forces to guide or disturb the learning. We consider a Markov decision process (MDP) in this setting and then formulate the interaction model following a previous study [28]. In this study, we assume that the human state is observable using additional sensors such as tactile sensors and the human’s policy can be fixed during learning episodes by prior instructions. The robot’s dynamics can be written as 

\[ s_{t+1}^R = f^R(s_t^R, a_t, s_t^H), \]

where \( s^R \in \mathbb{R}^{d_R} \) is the robot’s state (e.g., the pose of the end effector, joint angle, or velocity of the robot arm), \( s^H \in \mathbb{R}^{d_H} \) is the state of the human (e.g., external force provided by the human), and \( a \in \mathbb{R}^{d_a} \) is the robot’s action (e.g., the velocity or torque commands). Because the human state is also affected by the robot’s state and action, the human’s dynamics model can also be expressed as follows:

\[ s_{t+1}^H = f^H(s_t^H, s_t^R, a_t). \]

By integrating these models into a single model, the human-robot interaction dynamics model can be expressed as 

\[ s_{t+1} = f(s_t, a_t), \]

where \( s = [s^R, s^H] \).

The interaction model is approximated by a function \( f_\theta \) parameterized by \( \theta \). We learn the model using deep neural networks given the collected data \( D = \{s_n, a_n, s_{n+1}\}_{n=1}^N \) as the dynamics of a human-robot interaction typically becomes complex. The task is specified by a reward function \( r(s, a) \). Our objective is to find optimal robot actions that maximize the expected returns.

B. Advisory and Adversarial Interaction Strategy

Based on the human-robot interaction model, the robot learns policies (models) while the human provides physical interactions. First, the human guides the robot with advisory interactions to reduce errors during training and then disturbs it with the adversarial interactions to increase robustness. We prepared a reward function, which is given as follows:

\[ r(s, a) = r^R(s^R, a) + r^H(s^H), \]

where \( r^R \) is the reward for the robot (e.g., distance between the current position and goal, and the action cost) and \( r^H \) is the reward for the human (e.g., a penalty or bonus for human interactions).

Advisory interaction: First, the human attempts to guide the robot to help it learn. The human is instructed to provide corrective forces when the robot moves in the correct direction. When the robot deviates further from the goal, the corrective force becomes stronger. The human reward can be written as \( r^H = \alpha |s^H|^2 \), where \( \alpha \) is a weight parameter. This term indicates that the robot receives penalties from the human. Because the model is learned from the interactions, the robot can reach the goal even with adversarial interactions from the human.

Adversarial interaction: After the advisory interactions, the human then attempts to disturb the robot to make the learning more robust. The human is instructed to provide adversarial forces when the robot moves in the correct direction. When the robot approaches the goal, the adversarial forces become stronger. The human reward can be written as \( r^H = \alpha |s^H|^2 \), where \( \alpha \) is a weight parameter. This term indicates that the robot receives a bonus from the human. Because the model is learned from the human interactions, the robot can reach the goal even with adversarial interactions from the human.

C. Model-Based Reinforcement Learning

We employ a state-of-the-art model-based reinforcement learning method [7]. This method applies an ensemble of deep neural network models to deal with model uncertainty resulting in better sample efficiency than those of the previous methods. Based on the ensemble model, model predictive control (MPC) was used to obtain the optimal sequences of actions. Given the current state \( s_t \), a predictive horizon \( T \), and a sequence of actions \( a_{t:t+T} = \{a_t, \ldots, a_{t+T}\} \), we predict future outcomes \( s_{t:t+T} \) by using deterministic models \( f_\theta \) (we found that the deterministic models worked better than the probabilistic ones in our simulation and real-robot setting). We started by generating \( P \) particles for action sequences from uniform random distribution. For each particle, the future state is predicted by \( f_{\theta(p,t)} \), where \( b \in \{1, \ldots, B\} \) is the index of \( B \) bootstrap models in the ensemble. We uniformly re-sample the bootstrap on each timestep following a previous study [7].

For online planning using MPC, we applied a cross-entropy method (CEM), which was used in [7]. With the CEM, the candidates of the sequence of actions are sampled, and the distributions are then iteratively updated based on the expected return. We repeat this procedure with a pre-defined number of iterations.

Here, this procedure can be written as follows:

\[ a_{t:t+T}^p \sim \mathcal{N}(\mu_{t:t+T}^m, \Sigma_{t:t+T}^m), \]

\[ a_{\text{elites}} = \text{sort} \left[ a_{t:t+T}^p \right] [-J:], \]
We defined the robot’s action \( \alpha \) as only 3D lateral actions (x, z, and pitch) for the robot, and the human state \( \beta^H \) as a 1D action (x-axis).

Implementation details: We implemented our method based on a pytorch implementation for a model-based reinforcement learning algorithm by [33]. We designed an ensemble of \( B = 3 \) three-layer neural network models with ReLU activations except for the last layer with tanh activations. The number of neurons in each layer was 100 for the advisory interactions, and 100 neurons were then added for the adversarial interactions. The control and sample frequency was 10 Hz. For training, we used an Adam optimizer with a learning rate of 1e-5 and smooth-11 loss function. The model was updated at each timestep using a stochastic gradient method with a batch composed of 100 uniform and randomly selected data from \( D \). We could perform the update in real-time as we calculate only one gradient step on each timestep with low control frequency. The number of particles \( P \) used to sample the candidate actions was 300, the predictive horizon \( T \) was 3, the number of iterations \( M \) was 10, and the number of elites \( J \) was 50.

The reward function is given in Eq. (1). We designed the robot reward function to express the distance between the peg and hole.

\[
r^{R}(s_t, a_t) = -e^{xyz}_w e^{xyz} - W_z a_z^2, \tag{7}
\]

where \( e^{xyz} \) was the position error between the hole and the tip of the peg along the x-, y-, and z-axes given as \( e^{xyz} = s^{R}_{xyz} - s_{hole} \) (\( s_{hole} \) was the position of the hole); \( W_e \) was a weight matrix set to diag(1.0, 1.0, 10.0); and \( W_z \) was a weight parameter for the actions for the z-axis set to 0.001 if \( ||e^{xyz}|| < 0.007 \) and 1.0 otherwise. Owing to this action term, the robot could not apply an excessively large force to the environment. We also set the weight parameters for the human rewards \( \alpha_i \) and \( \alpha_e \) to 0.01.

Evaluation protocol: For training, we ran 14 episodes (seven advisory and adversarial interaction episodes) under each condition with five different random seeds. We defined the number of episodes empirically such that the participants did not become tired from a large number of interactions while the robot completed the tasks. The participants conducted the learning sessions twice on each random seed, and we then used the learned model for the test, which showed higher returns on the final episode. One episode included 15-s excursions corresponding to 150 data points.

We then tested the five models learned under each condition. During the test, the participants did not provide any interactions. We compared our methods with baselines, which did not use any interactions. For the baseline conditions, we learned only the robot models composed of \( \beta^R \) and \( \alpha \) without any interactions and with uniform random interactions to observe whether the participants provided informative interactions. We set the frequency for the random interactions to 1 Hz as the previous study estimated human control frequency in a similar task [34]. The learning setting was same as these conditions. The baseline has 200 neurons per layer from the beginning of learning for the same total number of neurons.

Fig. 2. Simulation platform. We implemented a simulation emulating a physically soft robot [8] using PyBullet. The participants provided external forces with the end-effector using a gamepad.

\[
\mu_{t+1}^{m+1} = \beta \cdot \text{mean} (\{ \alpha_{\text{elites}} \}) + (1 - \beta) \mu_t^{m+1}, \tag{5}
\]

\[
\sum_{t+1}^{m+1} = \beta \cdot \text{var} (\{ \alpha_{\text{elites}} \}) + (1 - \beta) \sum_{t+1}^{m+1}, \tag{6}
\]

where we select the top-\( J \) highest action sequences (\( \alpha_{\text{elites}} \)) with respect to the returns; \( p \in P \) and \( m \in M \) are the numbers of particles and iterations, respectively; and \( \beta \) is a coefficient of smoothing filters introduced in [31]. The optimal actions \( \alpha^{opt} \) can be obtained from

\[
p^{*} = \arg\max_{p^*} \sum_{t=0}^{T} r(p^*_t, \alpha^*_t). \]

IV. SIMULATION

To verify our method, we first conducted simulation experiments with real human participants. We aimed to confirm whether our method could learn the robust model with smaller errors from advisory and adversarial interactions. To conduct the simulation and real-robot experiments described in the next section, we obtained approval from the research ethics committee of OMRON Corp. and followed its guidelines. Six people participated in this simulation, and before commencing the experiment, each of them was instructed on how to interact with the robot. They were afforded one practice session to do the simulation.

A. Simulation Setup

Simulation platform: We applied peg-in-hole tasks because they are difficult to achieve owing to complex dynamics from contact-richness as well as human-robot interactions. We developed a simulation that emulates a soft robot previously proposed in [8], [29] using the PyBullet environment [32]. Similarly, we deployed virtual 6 DoF compliance components between the end-effector and the tip of the arm, as shown in Fig. 2. We implemented the compliance virtually using PD controllers for the six axes such that the end-effector could return to the equilibrium length. The frequency of the simulation step was set to 500 Hz. The participants provided external forces with a gamepad while watching the robot’s behavior displayed on a monitor.

Dynamics expression: We defined the robot’s state \( s^R \) as a 9D pose: 3D position and 6D orientations (with sin. and cos. expressions of the roll, pitch, and yaw orientations) of the tip of the peg under the assumption that it was fixed to the end-effector.

We defined the human’s state \( s^H \) as the external force provided by the gamepad and applied a smoothing filter. For simplicity and to avoid computational instability in this simulation, we defined the robot’s action \( \alpha \) as only 3D lateral actions (x, z, and pitch) for the robot, and the human state \( \beta^H \) as a 1D action (x-axis).

We defined the robot’s state \( s^R \) as a 9D pose: 3D position and 6D orientations (with sin. and cos. expressions of the roll, pitch, and yaw orientations) of the tip of the peg under the assumption that it was fixed to the end-effector.
Fig. 3. Learning curves in simulation. (a) Mean absolute x, y, and z position errors during the episodes. Our method showed smaller errors than the baselines at the end of the advisory and adversarial interactions. (b) Cumulative force of participants. The force decreased for the advisory interactions and increased for the adversarial interactions.

We also manually designed simple interaction strategies that imitated the human behaviors described in Section III-B to compare them with the real-human interactions. We call them “heuristic interactions” expressed as follows:

Advisory:

$$s_t^H = \begin{cases} \text{sgn}(-e_{xt})(1 - \exp(-|e_{xt}|/l_i)) & \text{if } |e_{xt}| - |e_{xt-1}| > 0 \\ 0 & \text{otherwise} \end{cases}$$

(8)

Adversarial:

$$s_t^H = \begin{cases} \text{sgn}(-e_{xt})(\exp(-|e_{xt}|/l_e)) & \text{if } |e_{xt}| - |e_{xt-1}| < 0 \\ 0 & \text{otherwise} \end{cases}$$

(9)

where $e_{xt}$ is an x-position error, and $l_i$ and $l_e$ are scales set to 0.04 and 0.1, respectively. We applied the heuristic interactions to the proposed model with the progressive networks.

We ran rollouts with 30 different random seeds in the environment used for training. We considered that if the positional error for the z-axis at the end of the episode was less than 0.005 m, the robot successfully completed its task. Moreover, to evaluate the robustness, we tested these models under unknown conditions by changing the end-effector mass to (0.01, 0.5, 1.0, and 2.0) kg and the friction of the hole to (0.25, 0.75, 1.0, 1.5) as the parameter difference can be considered as external forces/disturbances in the system [10]. We compared the number of successes among these methods.

B. Simulation Results

Fig. 3(a) shows the learning curves under each condition detailing how learning proceeded. The orange line represents the mean absolute x-, y-, and z-position errors during the episode for the baseline without any interactions. The yellow line represents the same position errors during the episodes with random interactions of the five seeds, and the light and dark blue lines are the errors for our proposed method with heuristic and participants’ interactions. Each participant’s mean error was calculated with the five seeds. The shaded areas show the standard deviations among the participants. The errors of the baselines exhibited a similar tendency of decreasing in the early stages of learning and then becoming constant. In the seventh episode on the advisory interactions, our method showed smaller errors than the baselines. Our method showed slightly smaller errors when trained on the heuristic interactions instead of the participants’ interactions.

Next, Fig. 3(b) shows the mean absolute external forces during the episodes to investigate the amount of forces provided by the participants and the heuristic strategy on the learning session. In the advisory interactions, the force in the seventh episode became smaller than that in the first episode. Moreover, in the adversarial interactions, the forces gradually increased as the number of episodes increased. Based on these results, our method showed smaller errors than the baselines. Our method showed slightly smaller errors when trained on the heuristic interactions instead of the participants’ interactions.

To investigate the participant’s behaviors, we plotted trajectories x- and z-position errors, and x-axis external forces of one of the participants on 1st, 7th, and 14th episode in Fig. 4. In the advisory interactions, when the position error increased towards the positive direction, the participant provided the forces toward the negative direction. Meanwhile, in the adversarial cases, when the error decreased in the negative direction, the participant provided the forces in the positive direction. This shows that the participants appropriately interacted with the robot during the learning sessions.
Next, we tested both the models learned from the baseline and the proposed method by applying them to unknown environments to determine the robustness of our approach. Figs. 5(a) and (b) show the number of successes of the five different models with 30 random seeds at the different hole frictions and end-effector masses, respectively. The orange bars are the number of successes of the baseline without any interactions, the yellow bars are the baseline with random interactions, and the light and dark blue bars are those of the proposed method with heuristic and participants’ interactions. The error bars show the standard deviations among the participants. The “default” label on the x-axis indicates the environment where the robot was trained. The proposed method demonstrated a larger number of successes compared to the baselines. Interestingly, our method showed a larger number of successes when trained on the participants’ interactions rather than heuristic ones, in most of the cases. As the results indicate, our method can obtain more robust skills, allowing it to handle unknown environments better than the baselines.

V. REAL ROBOT EXPERIMENT

To confirm the effectiveness in real-world environments, we also conducted a real-robot experiment. Our goal was to investigate whether our method can learn the peg-in-hole skills more successfully and robustly than the baseline.

A. Experimental Setup

Experimental platform: We used a Universal Robot (UR5), a soft wrist [8], and a Robotiq gripper (Hand-E) (see Figs. 6 (a) and (b)). The wrist consisted of three springs that allowed for 6D deformation. As with the simulation, we addressed a peg-in-hole task. The diameter of the peg was 9 mm, and the tolerance of the hole was 1 mm. We measured the pose of the gripper using a motion capture system (Optitrack, FLEX13) (see Fig. 6(a)). To measure the external forces from the human participant, we placed two tactile sensors (Nippon Liniax Corp., TFS12A-25) on the gripper (see Fig. 6(c)). In this study, we considered only the x- and y-axis forces and calculated the combined forces from the two sensors. The participant grasped the sensors equipped with the gripper and provided external forces in the x- and y-axes.

System dynamics: The robot’s state $s^R$ was the 6D gripper pose (3D position and 3D orientation) similar to the simulation. The human’s state $s^H$ was the combined 2D forces measured from the tactile sensors. The robot’s action $a$ was 6D Cartesian velocity commands to the tip of the arm.

Implementation details and evaluation: Most of the implementation was the same as in the simulation. During this experiment, we also trained the models using 14 episodes (seven advisory and adversarial interactions) with five different seeds, while the participant provided interactions. The start position was 15 mm away from the hole in the y-axis. In the real robot experiment, we designed the weights for the reward as $W_e = \text{diag}(20, 20, 5.0)$ with $\|e_{xy}\| < 0.002$ m set to 0.01; otherwise, it was set to 1.0. We set $\alpha_i$ and $\alpha_e$ to 0.1. The control and sample frequency was 5 Hz.

After training the models, we tested them without any interactions. We compared our method with the baseline in three different environments. We prepared three different pegs: 1) default, the same as the training, 2) plastic made from mc-nylon for the smaller friction, and 3) a sponge attached to the tip of the peg for the larger friction (see Fig. 8). We tested five learned models using 20 different random seeds. We defined the robot as successful if the error along the z-axis was smaller than 0.003 m.

B. Results

Fig. 7(a) shows learning curves under each condition to see how the learning proceeded. The orange line is the mean absolute x, y, and z position errors of the baseline during the episode among the five seeds, and the dark blue line is the error of
Fig. 7. Learning curves of the simulation: (a) Mean absolute x, y, and z position errors during the episodes. During the early learning stage of our method, the errors were smaller than those of the baseline. In the 14th episode, the variance of the proposed method was also smaller. (b) Mean absolute force provided by the participant. Similar to the simulation, the forces decreased and increased in the advisory and adversarial interactions.

Fig. 8. Number of successes on different environments (pegs). Our method showed a larger number of successes than the baseline under all conditions.

| Episode | 14 (default) | 16 | 20 | 50 | 100 | 200 |
|---------|--------------|----|----|----|-----|-----|
| Error [m]| 0.013        | 0.011 | 0.013 | 0.015 | 0.016 | 0.016 |

Table I: Convergence Check of the Baseline

our proposed method. The shaded areas show the standard deviations for the five seeds. Both conditions showed a gradual decrease in errors. During the early learning stage of our method, the errors were smaller than those of the baseline because of the participant’s corrective interactions. In the 14th episode, both conditions demonstrated reduced errors, although the variance of the proposed method was smaller than that of the baseline.

Next, Fig. 7(b) shows the mean absolute force that the participant provided during the learning sessions. Similar to the simulation, the forces decreased with the advisory interactions and increased with the adversarial interactions. The results show that our method successfully learned the peg-in-hole tasks even in a real-world environment.

Finally, we tested the learned models from the baseline and the proposed method by applying them to unknown environments. Fig. 8 shows the number of successes of the five different models with 20 random seeds for the three different pegs. The orange bars represent the number of successes of the baseline, and the dark blue bars represent those of the proposed method. We also demonstrated the results of learning from different start positions. For the default condition, the starting positions were arranged around the hole with a 15 mm distance, as shown in Fig. 8. As the results show, our proposed method demonstrated a larger number of successes than the baseline in all of the environments.

VI. DISCUSSION

The previous sections demonstrated the effectiveness of our methods in both the simulation and real robot experiments. In this section, we discuss the experimental results and limitations of our method.

To evaluate the convergence, we increased the number of episodes for the baseline without any interactions. Table I lists the mean absolute errors of the training. We confirmed that the smallest error can be seen on the 16th episode (however, that of the proposed method was smaller), and then the errors slightly increased. Next, we changed a number of episodes for our method. We asked the participant for five and 10 advisory interactions and adversarial interactions corresponding to a total of 10 and 20 episodes, respectively. The results demonstrated that the errors of the last episode learned with 10, 14, and 20 episodes were 0.009, 0.007, and 0.013 m, respectively.

Our method showed better performances likely because the advisory interactions could avoid the larger errors in the early stage of learning and the adversarial ones could also function as exploration. A similar result was reported in a previous study [10]. It compared the proposed method learned from adversarial perturbations with the baseline. The method also showed a better performance than the baseline in the training environment. Moreover, the adversarial interactions can help the learning process because they emulate situations in which the robot had difficulty reaching the goal owing to the larger mass or friction of the environments.

However, we should improve our proposed method to make it more successful. It sometimes caused the robot to become stuck as it moved downward even if the peg was not fitted to the hole. One possible solution could be to create more lateral connections of the progressive neural networks to consider additional interaction strategies. We might be able to learn how to recover from a stuck robot through additional interactions and lateral connections.
In the future, we will investigate why our method worked better in more detail such as the differences between the participants’ and the heuristic interactions. We also plan to evaluate our method under more various environments, such as different shapes or tolerances of the peg and hole, and explore different robotic manipulation tasks such as the pouring, drawing, or pushing an object.

VII. CONCLUSION

In this study, we proposed a novel learning framework using soft robots. Our key idea was to exploit both advisory and adversarial interactions and build human-robot interaction models. To verify our method, we conducted simulations and real-robot experiments with real human participants. The results demonstrated that our method learned a task with smaller errors in training and demonstrated a larger number of successes even under unknown environments when compared to the baselines.

REFERENCES

[1] J. Xu, Z. Hou, Z. Liu, and H. Qiao, “Compare contact model-based control and contact model-free learning: A survey of robotic peg-in-hole assembly strategies,” 2019, arXiv:1904.05240.

[2] U. E. Ogenyi, J. Liu, C. Yang, Z. Ju, and H. Liu, “Physical human-robot collaboration: Robotic systems, learning methods, collaborative strategies, sensors, and actuators,” IEEE Trans. Cybern., vol. 51, no. 4, pp. 1888–1901, Apr. 2021.

[3] I. Bahl et al., “Challenges and countermeasures for adversarial attacks on deep reinforcement learning,” 2020, arXiv:2001.09684.

[4] L. Pinto, J. Davidson, and A. Gupta, “Supervision via competition: Robot adversaries for learning tasks,” in Proc. IEEE Int. Conf. Robot. Automat., 2017, pp. 1601–1608.

[5] J. Duan, Q. Wang, L. Pinto, C.-C. J. Kuo, and S. Nikolaidis, “Robot learning via human adversarial games,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2019, pp. 1056–1063.

[6] A. A. Rusu et al., “Progressive neural networks,” 2016, arXiv:1609.04671.

[7] K. Chua, R. Calandra, R. McAllister, and S. Levine, “Deep reinforcement learning in a handful of trials using probabilistic dynamics models,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 4754–4765.

[8] K. Tanaka et al., “A compact, cable-driven, activatable soft wrist with six degrees of freedom for assembly tasks,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2020, pp. 8752–8757.

[9] V. Behzadan and A. Munir, “Vulnerability of deep reinforcement learning to policy induction attacks,” in Proc. Int. Conf. Mach. Learn. Data Mining Pattern Recognit., 2017, pp. 262–275.

[10] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, “Robust adversarial reinforcement learning,” in Proc. Int. Conf. Mach. Learn., 2017, pp. 2817–2826.

[11] O. Ogunnolu, N. Gans, and T. Summers, “Minimax iterative dynamic game: Application to nonlinear robot control tasks,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2018, pp. 6919–6925.

[12] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and D. Abbeel, “Adversarial attacks on neural network policies,” 2017, arXiv:1702.02384.

[13] A. Mandlekar, Y. Zhu, A. Garg, L. Fei-Fei, and S. Savarese, “Adversarially robust policy learning: Active construction of physically-plausible perturbations,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2017, pp. 3932–3939.