2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Learning Soft Robotic Assembly Strategies from Successful and Failed Demonstrations

Masashi Hamaya[1], Felix von Drigalski[1], Takamitsu Matsubara[2], Kazutoshi Tanaka[1], Robert Lee[3], Chisato Nakashima[4], Yoshiya Shibata[4], and Yoshihisa Ijiri[1,4]

***Abstract*— Physically soft robots are promising for robotic assembly tasks as they allow stable contacts with the environment. In this study, we propose a novel learning system for soft robotic assembly strategies. We formulate this problem as a reinforcement learning task and design the reward function from human demonstrations. Our key insight is that the failed demonstrations can be used as constraints to avoid failed behaviors. To this end, we developed a teaching device with which humans can intuitively provide various demonstrations. Moreover, we leverage Physically-Consistent Gaussian Mixture Models to clearly assign Gaussian components to the successful and failed trials. We then create the reference trajectories via Gaussian Mixture Regressions, which fit the successful demonstrations while considering the failed ones. Finally, we apply a sample-efficient deep model-based reinforcement learning method to obtain robust strategies with a few interactions. To validate our method, we developed a real-robot experimental system composed of a rigid collaborative robot arm with a compliant wrist and the teaching device. Our results demonstrated that our method learned the assembly strategies with a higher success rate than when using only successful demonstrations.**

I. INTRODUCTION

Autonomous robotic assembly is an essential component for industrial applications. Despite significant research and development, robotic assembly tasks are still challenging as they involve strict tolerances and complex contact dynamics. In order to deal with such contact-rich tasks, force controllers have been proposed [1]. Although recent studies have demonstrated high precision assembly control solutions [2], [3], these approaches largely depend on high-performance hardware, e.g., high-frequency controllers or precise force/torque sensors.

In contrast, *physically* soft robots (consisting of springs and compliant materials) have attracted much attention [4], [5] as they can handle contact-rich interactions intrinsically without the force controllers and sensors. Meanwhile, designing controllers for soft robotic assembly is much more difficult due to the complex dynamics of soft bodies in a contact-rich environment. Especially, we need to carefully design the control objective, since unlike the rigid robots, following reference trajectories precisely is difficult for the soft robots. A poorly designed control objective would often cause failures, for example, applying a high force to an environment where soft materials shrink at the limit, and unexpected behaviors such as oscillation and overshooting can occur due to the elastic effects. If we can obtain the controller and its objective in a more intuitive way and let the robots know about these potential failures in advance, they would likely perform the assembly tasks more successfully.

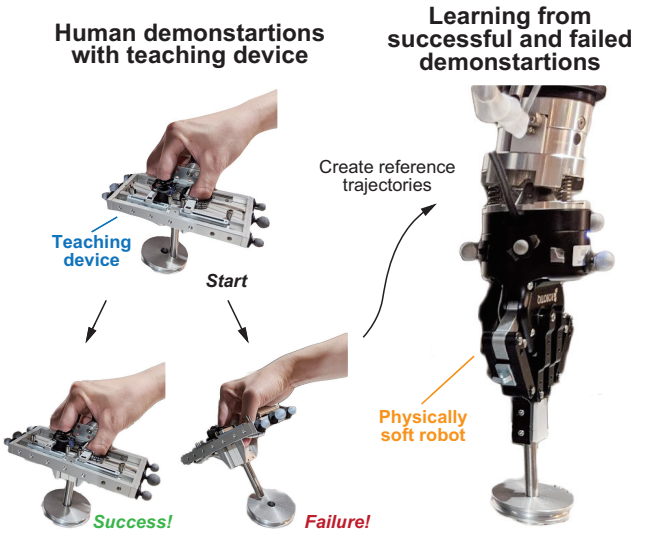

Fig. 1. Illustration of our proposed method. We propose a novel learning system for soft robotic assembly tasks. We apply a reinforcement learning approach and design the reward function from human demonstrations. Our key idea is to exploit successful and failed demonstrations to obtain assembly strategies more successfully.

In this study, we explore a data-driven approach to acquire successful soft robotic assembly strategies. We formulate this problem as a reinforcement learning task and design a reward function with a Learning from Demonstrations (LfD) approach. LfD is promising since humans can intuitively provide their strategies with the robots [6], [7], and it has also been used widely in industrial applications [8]. Furthermore, we propose to leverage failed demonstrations as well as successful ones as shown in Fig. 1. The motivation is that the failed demonstrations can be used as constraints to avoid the failed behaviors. This dual approach results in extracting the necessary strategies to complete the tasks more successfully using both the failed and successful demonstrations.

However, two serious problems arise. First, since the assembly tasks require high precision manipulation, providing adequate demonstrations is sometimes difficult. For example, in direct teaching cases using collaborative robots [9], manip-

[1] MH, FD, KT, and YI are with OMRON SINIC X Corporation, Tokyo, Japan masashi.hamaya@sinicx.com
[2] TM is with Graduate School of Science and Technology, Nara Institute of Science and Technology, Nara, Japan
[3] RL is with the Australian Center for Robotic Vision at Queensland University of Technology in Brisbane, Australia
[4] YS and CN are with OMRON Corporation, Tokyo, Japan

ulating the end effector to the desired location is challenging due to the robot joints restricting the user's movement. To solve this problem, we developed a teaching device (see Fig. 1), which mimics the robot's gripper and facilitates the demonstration of assembly strategies. Moreover, we can argue that the pose information from this device is sufficient to reproduce the contact-rich tasks using simple position or velocity control, since the robot's softness allows stable contacts [10].

Second, if we apply standard methods such as a Gaussian Mixture Model (GMM) and Gaussian Processes (GP) [7], or GMM-based GP [11] to the collected demonstrations, they are prone to fit averaged trajectories such that distinctions between the success and failure trials are difficult. To address this problem, we employ a Physically-Consistent Gaussian Mixture Model (PC-GMM) that considers similarities such as locality and directionality between the demonstrations [12]. Whereas Figueroa and Billard used GMM to clearly fit complex trajectories [12] and demonstrated better expressiveness compared to the standard EM-based GMM methods, we propose to use PC-GMM to distinguish the successful and failed demonstrations. By applying PC-GMM, the Gaussian components are unambiguously assigned to the successful and failed ones while keeping away from each other. Then, we create time-dependent reference trajectories via Gaussian Mixture Regressions (GMR) using the optimized Gaussian components by PC-GMM. We extract the Gaussian components, which are fitted to the successful demonstrations.

Finally, given the reference trajectories from the demonstrations, we apply a deep model-based reinforcement learning method [13] to obtain robust assembly strategies. This method combined ensembles of deep neural network dynamics models with sampling-based propagation to contend with nonlinearity and uncertainty, and demonstrated notable sample-efficiency when compared with other modern model-based reinforcement learning methods [13]. It is suitable for our setting, which includes the model complexity due to the softness and contact richness.

The contributions of this study are as follows:

- We propose a novel learning system for soft robotic assembly strategies. We formulate the problem as a reinforcement learning task and design the reward function from successful and failed demonstrations.
- We developed a teaching device, with which the users can provide various demonstrations intuitively and employ PC-GMM to exploit the successful and failed demonstrations. In addition, we apply a state-of-the-art model-based reinforcement learning method.
- We performed real robot experiments. Our method showed a higher success rate in a peg-in-hole task compared with using only the successful demonstrations.

This paper is organized as follows. Section II presents related works for our study. We introduce our proposed method in Section III. Section IV describes the experiment, whereas, Section V discusses our results and Section VI presents our conclusion.

II. RELATED WORKS

In this section, we describe related works for the soft robotic assembly control. We also focus on recent studies learning from demonstration and failed behaviors.

A. Physically soft robotic assembly control

Physically soft robots are suitable for assembly tasks since the softness allows safer interactions with the environment [14]. Yun et al. demonstrated that lower stiffness improved the performance of the peg-in-hole tasks in simulations [15]. Nishimura et al. developed a passive compliant wrist which can deal with position uncertainty during the peg-in-hole tasks [4]. Soft tactile sensors have been used for measuring and aligning the orientations of the grasped assembly parts [16], [17].

Our prior work presented learning assembly controllers for the soft robot [18]. We leveraged softness and environmental constraints so that the robot can complete tasks in a lower-dimensional state and action spaces. Then, we applied sample efficient reinforcement learning. Although we demonstrated the sample efficiency, expert knowledge would be required to design the constraints. In addition, the robots had to learn the strategies for each sub-task. Unlike the previous work, we apply the novel learning system utilizing human demonstrations such that designing constraints or learning the strategies for each sub-task are not required.

B. Learning from demonstrations and failed behaviors

LfD approaches have previously been applied to soft robots. Nishimura et al. and Della et al. also employed human demonstrations to imitate object manipulation and grasping [19], [20]. Gupta et al. proposed to combine reinforcement learning with human demonstrations [21]. These methods showed efficient learning performances; however, they only provided successful demonstrations.

Meanwhile, recent studies have also proposed to learn from failed behaviors. Wang et al. and Kobayashi et al. proposed a dual reward function to acquire strategies to maximize the positive reward and minimize the negative reward [22], [23]. Esteban et al. proposed a reinforcement learning method, whose policies were updated to avoid the bad experiences [24]. Failed or imperfect demonstrations have also been used in inverse reinforcement learning [25] and deep reinforcement learning contexts [26]. As an alternative, we propose to leverage the successful and failed demonstrations to directly create reference trajectories, which can extract the necessary strategies to complete the tasks. To this end, we develop the teaching device, and employ PC-GMM and state-of-the-art model-based reinforcement learning.

III. PROPOSED METHOD

In this section, we present our proposed method. Our goal is to obtain the appropriate soft robotic assembly strategies. The procedure of our method is depicted in Fig. 2. First, we provide the labeled successful and failed demonstrations on

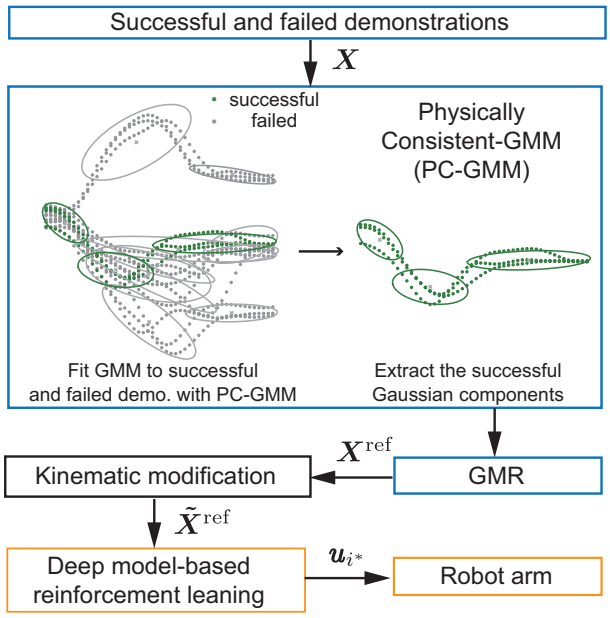

Fig. 2. Procedure of our method. First, we collect the successful and failure demonstrations with a teaching device. Next, we apply PC-GMM and GMR to the labeled demonstrations. Finally, we employ deep model-based reinforcement learning.

the peg-in-hole task with the teaching device. The demonstrated data is labeled with success or failure. Given the collected data, we apply PC-GMM to fit GMM to the both successful and failed ones. Then, we employ GMR to create the reference trajectories using the Gaussian components, which are assigned only to the successful data. Finally, based on the trajectories, we apply deep model-based reinforcement learning.

A. Physically-Consistent Gaussian Mixture Model

In attempting to exploit both successful and failed demonstrations, we cannot directly apply typical methods such as GMM or GP since they tend to fit to the average of both successful and failed trials. To address this problem, we employ PC-GMM [12]. PC-GMM uses a similarity measure based on locally-scaled cosine similarity of the velocity to bias the GMM fitting. In addition, PC-GMM automatically estimates the number of Gaussian components using a Bayesian non-parametric approach, the Chinese Restaurant Process (CRP) representation. Below, we briefly explain PC-GMM. The details can be seen in [12].

The probabilistic distribution of GMM at the data $\boldsymbol{x}$ (labeled with success and failure), mixture of K Gaussian distributions $\mathcal{N}(\cdot|\theta_\gamma)$ with $\theta_\gamma = \{\boldsymbol{\mu}_\gamma^k, \boldsymbol{\Sigma}_\gamma^k\}$ can be written as:

$$p\left(\boldsymbol{x}|\Theta_\gamma\right) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\boldsymbol{x}|\boldsymbol{\mu}^k, \boldsymbol{\Sigma}^k\right), \tag{1}$$

where $\pi_k = p(z_i = k)$ is a mixing coefficient, $\Theta_\gamma = \{\pi_k, \theta_\gamma^k\}_{k=1}^K$ is a parameter set for the Gaussian distribution. $Z = [z_1, \ldots, z_M]$ is an assignment variables with M samples, where $z \in [1, ..., K]$.

We then use the physically consistent similarity dependent CRP. The physically consistent similarity is composed of two main properties: directionality and locality. The similarity measure Δ is given as:

$$\Delta_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j, \dot{\boldsymbol{x}}_i, \dot{\boldsymbol{x}}_j) = \eta\left(1 + \frac{(\dot{\boldsymbol{x}}_i)^\top \dot{\boldsymbol{x}}_j}{\|\dot{\boldsymbol{x}}_i\| \|\dot{\boldsymbol{x}}_j\|}\right) \exp\left(-l_s \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right). \tag{2}$$

The first term, which represents directionality, is the shifted cosine similarity of measured velocity, the second term represents the locality with a Gaussian kernel on the position measurements, and l_s is a scale parameter. In this study, we add a modification parameter η to reduce the similarity $\eta = 0.01$ when the labels (success and failure) between the two data points are different, otherwise $\eta = 1.0$.

The similarity-dependent CRP generates a prior distribution $p(C)$ over customer seating assignments $C = [c_1, \ldots, c_M]$ where $i : c_i = j$ indicates that the i and j customers are in the same cluster. The prior is computed as:

$$p(C|\boldsymbol{\Delta}, \alpha) = \prod_{i=1}^{M} p\left(c_i = j|\Delta, \alpha\right), \tag{3}$$

$$p\left(c_i = j|\boldsymbol{\Delta}, \alpha\right) = \begin{cases} \frac{\Delta_{ij}(\cdot)}{\sum_{j=1}^{M} \Delta_{ij}(\cdot) + \alpha} & \text{if} \quad i \neq j \\ \frac{\alpha}{M+\alpha} & \text{if} \quad i = j, \end{cases} \tag{4}$$

where $\boldsymbol{\Delta} \in \mathbb{R}^{M \times M}$ is the pairwise similarity matrix between M customers and α is the probability to sit alone.

$C = [c_1, \ldots, c_N]$ is sampled from Eq. 3 and mapped to $Z = [z_1, \ldots, z_N]$ using a function $Z = \boldsymbol{Z}(C)$. The parameters θ_γ^k are sampled from the Normal-Inverse-Wishart distribution with a hyperparameter λ_0. The optimal number of the components K is derived from the number of unique clusters of C. We estimate the posterior distribution $p(C, \Theta_\gamma|\boldsymbol{X})$ given the data $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_M]$. However, since obtaining the analytic posterior is intractable, we use Collapsed Gibbs sampling for the approximated posterior as follows:

$$p(c_i = j|C_{-1}, \boldsymbol{X}, \boldsymbol{\Delta}, \alpha, \lambda_0) \propto p\left(c_i = j|\boldsymbol{\Delta}, \alpha\right) p\left(\boldsymbol{X}|\boldsymbol{Z}\left(c_1 = j \cup C_{-1}\right), \lambda_0\right), \tag{5}$$

where the second term in the right hand-side is the likelihood of table assignments coming from the current seating arrangement $\boldsymbol{Z}(c_i = j \cup C_{-i})$. C_{-i} is the customer seating assignment for all customers except for the i-th. We run the sampler in Eq. 5 iteratively and obtain the optimal number of Gaussian components K and their parameters Θ_γ via the Maximum A Posteriori.

B. Gaussian Mixture Regression

Using the Gaussian components obtained from PC-GMM, we apply GMR [27] to create the trajectories. We select the Gaussian components, which are assigned to the successful demonstration data from all components Θ_γ. At the timestep t, we decompose the data $\boldsymbol{x}_t$ into two vectors $\boldsymbol{x}_t^{\mathcal{I}}$ and $\boldsymbol{x}_t^{\mathcal{O}}$.

$\mathcal{I}$ is the input dimension (time step) and $\mathcal{O}$ is the output dimension (e.g., end effector pose). The data $\boldsymbol{x}_t$, $\boldsymbol{\mu}_k$, and $\boldsymbol{\Sigma}_k$ can be written as follows:

$$\boldsymbol{x}_t = \begin{bmatrix} \boldsymbol{x}_t^{\mathcal{I}} \\ \boldsymbol{x}_t^{\mathcal{O}} \end{bmatrix}, \ \boldsymbol{\mu}_k = \begin{bmatrix} \boldsymbol{\mu}_k^{\mathcal{I}} \\ \boldsymbol{\mu}_k^{\mathcal{O}} \end{bmatrix}, \ \boldsymbol{\Sigma}_k = \begin{bmatrix} \boldsymbol{\Sigma}_k^{\mathcal{I}} & \boldsymbol{\Sigma}_k^{\mathcal{I}\mathcal{O}} \\ \boldsymbol{\Sigma}_k^{\mathcal{O}\mathcal{I}} & \boldsymbol{\Sigma}_k^{\mathcal{O}} \end{bmatrix}. \tag{6}$$

We approximate the conditional distribution of $p(\boldsymbol{x}_t^{\mathcal{O}}|\boldsymbol{x}_t^{\mathcal{I}})$ to the single peak distribution:

$$p\left(\boldsymbol{x}_t^{\mathcal{O}}|\boldsymbol{x}_t^{\mathcal{I}}\right) = \mathcal{N}(\boldsymbol{x}_t^{\mathcal{O}}|\hat{\boldsymbol{\mu}}_t^{\mathcal{O}}, \hat{\boldsymbol{\Sigma}}_t^{\mathcal{O}}), \tag{7}$$

$$\hat{\boldsymbol{\mu}}_t^{\mathcal{O}} = \sum_{k=1}^{K} h_k(\boldsymbol{\mu}_k^{\mathcal{O}} + \boldsymbol{\Sigma}_i^{\mathcal{O}\mathcal{I}}\boldsymbol{\Sigma}_k^{\mathcal{I}-1}\left(\boldsymbol{x}_t^{\mathcal{I}} - \boldsymbol{\mu}_i^{\mathcal{I}}\right)), \tag{8}$$

$$\hat{\boldsymbol{\Sigma}}_t^{\mathcal{O}} = \sum_{k=1}^{K} h_k(\boldsymbol{\Sigma}_k^{\mathcal{O}} - \boldsymbol{\Sigma}_k^{\mathcal{O}\mathcal{I}}\boldsymbol{\Sigma}_k^{\mathcal{I}-1}\boldsymbol{\Sigma}_k^{\mathcal{I}\mathcal{O}} + \hat{\boldsymbol{\mu}}_k^{\mathcal{O}}\hat{\boldsymbol{\mu}}_k^{\mathcal{O}\top}) - \hat{\boldsymbol{\mu}}_t^{\mathcal{O}}\hat{\boldsymbol{\mu}}_t^{\mathcal{O}\top}, \tag{9}$$

$$h_k = \frac{\pi_k \mathcal{N}\left(\boldsymbol{x}_t^{\mathcal{I}}|\boldsymbol{\mu}_k^{\mathcal{I}}, \boldsymbol{\Sigma}_k^{\mathcal{I}}\right)}{\sum_k^K \pi_k \mathcal{N}\left(\boldsymbol{x}_t^{\mathcal{I}}|\boldsymbol{\mu}_k^{\mathcal{I}}, \boldsymbol{\Sigma}_k^{\mathcal{I}}\right)}. \tag{10}$$

We use the mean $\hat{\boldsymbol{\mu}}_t^{\mathcal{O}}$ for the reference trajectories $\boldsymbol{X}^{\mathrm{ref}}$.

C. Deep model-based reinforcement learning

Given the created trajectories, we apply the deep model-based reinforcement approach [13], [28] to the soft robot. This algorithm is called probabilistic ensembles with trajectory sampling (PETS). PETS learns an ensemble of neural network dynamics models to deal with the uncertainty that arises in the low data regime of learning.

We consider the Markov decision process formulation. The robot state (e.g., position and orientation) is $\tilde{\boldsymbol{x}}$, and the robot action (e.g., velocity command of the tip of the arm) is $\tilde{\boldsymbol{u}}$. Given the current state and action, the next state is given as $\tilde{\boldsymbol{x}}_{t+1} = f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{u}})$, where the model is parameterized by $\boldsymbol{\theta}$. Unlike previous studies [13], [28], we learn deterministic dynamics models $f_{\boldsymbol{\theta}}$ with the training dataset $D = \{\tilde{\boldsymbol{x}}_t, \tilde{\boldsymbol{u}}_t, \tilde{\boldsymbol{x}}_{t+1}\}$, as we found the probabilistic models did not assist in our problem setting. Then, we employ model ensembles of B bootstrap models with b-th models $f_{\boldsymbol{\theta}_b}$.

Next, we perform online planning with model predictive control to select optimal actions. The learned models are used to predict a short-term return $t \in [0, \ldots, H]$ at each time step. We employ the cross-entropy method [29], which starts by generating random action sequences $\boldsymbol{U}_i$ $(i \in I)$. Then, the mean and variance of the sampling distribution are updated with a smoothing operation [28] for L iterations $(l \in L)$ based on the highest J return action sequences:

$$\boldsymbol{U}_i = [\boldsymbol{u}_0^i, \ldots, \boldsymbol{u}_{H-1}^i], \boldsymbol{u}_t^i \sim \mathcal{N}(\boldsymbol{\mu}_t^l, \boldsymbol{\Sigma}_t^l), \tag{11}$$

$$\boldsymbol{U}_{\text{elites}} = \text{sort}\left(\boldsymbol{U}_i\right)[-J:], \tag{12}$$

$$\boldsymbol{\mu}_t^{l+1} = \beta * \text{mean}\left(\boldsymbol{U}_{\text{elites}}\right) + (1-\beta)\boldsymbol{\mu}_t^l, \tag{13}$$

$$\boldsymbol{\Sigma}_t^{l+1} = \beta * \text{var}\left(\boldsymbol{U}_{\text{elites}}\right) + (1-\beta)\boldsymbol{\Sigma}_t^l, \tag{14}$$

where β is a smoothing coefficient. After L iterations, we select the optimal actions $\boldsymbol{U}_{i*}$ that maximize the predicted reward: $i^* = \arg\max_i \Sigma_{t'=t}^{t+H-1} r(\tilde{\boldsymbol{x}}_t, \tilde{\boldsymbol{u}}_t)$.

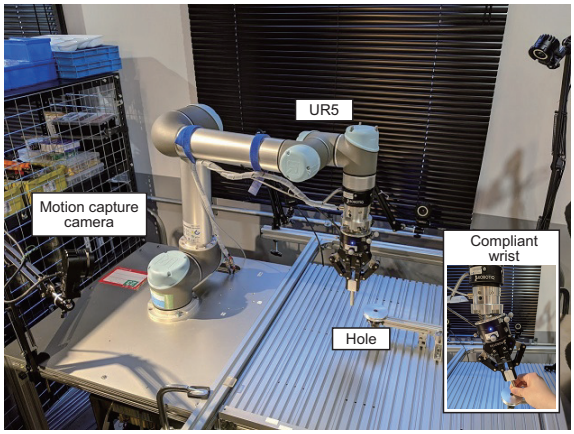

Fig. 3. The experimental setup consisted of a UR5, the motion capture system, and the compliant wrist.

The reward can be written as:

$$r(\tilde{\boldsymbol{x}}_t, \tilde{\boldsymbol{u}}_t) = -(\tilde{\boldsymbol{x}}_t^{\mathrm{ref}} - \tilde{\boldsymbol{x}}_t)^\top \boldsymbol{W}_{\boldsymbol{x}}(\tilde{\boldsymbol{x}}_t^{\mathrm{ref}} - \tilde{\boldsymbol{x}}_t) - W_u \tilde{u}_z^2$$
$$W_u = \begin{cases} 0.01 & \text{if } d > \epsilon \\ 0 & \text{otherwise,} \end{cases} \tag{15}$$

where $\boldsymbol{W}_{\boldsymbol{x}}$ is a weight matrix of the reward related to the position error. We also design the action reward for the z direction to avoid applying force to the environment excessively. If the robot is far from the goal $d > \epsilon$, where d is an x-y distance from the goal position, we increase a weight W_u so that the robot does not exceed the workable limit of the environment.

IV. EXPERIMENT

To validate our method, we conducted a real robot experiment. We developed the experimental system composed of a robot with a compliant wrist, a motion capture system, and a teaching device. This experiment goal was to investigate whether using both the successful and failure demonstrations could make learning perform better than when using only successful ones.

A. Setup

We utilized a Universal Robot (UR5) and a compliant wrist, which was attached between the arm and a Robotiq 2F-85 gripper (Fig. 3) [18]. A peg was firmly affixed to the gripper using a 3D printed jig for simplicity. We employed MoveIt to control Cartesian positions and velocities for the tip of the arm [30], and measured the 6D pose of the gripper using Optitrack, a motion capture system due to the under actuation of the compliant wrist. The diameter of the peg was 10 mm. The tolerance between the peg and the hole was 15 μm. This peg-in-hole task is considered difficult due to the low tolerance and no chamfer.

We designed a teaching device in which users could demonstrate their skills intuitively (Fig. 4). The device mimics the Robotiq gripper to reduce kinematic differences and allows the users to easily manipulate it with their fingers. The opening width of the fingers can be measured

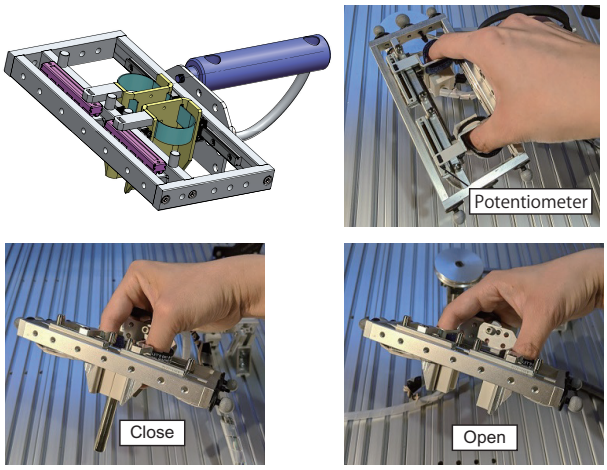

Fig. 4. The teaching device. It was designed so that the users can provide demonstrations easily while keeping the kinematics structure similar to the robot gripper.

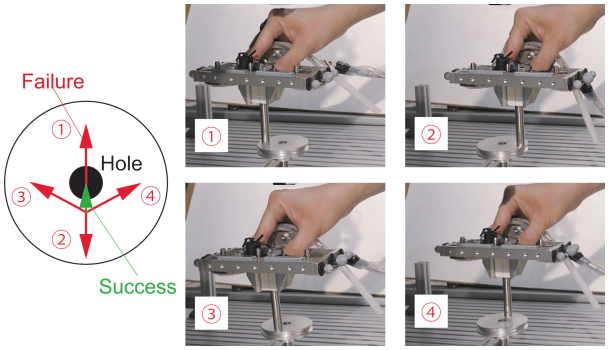

Fig. 5. Examples of failure demonstrations. A participant was instructed to slide the peg in four incorrect directions to imitate overshooting.

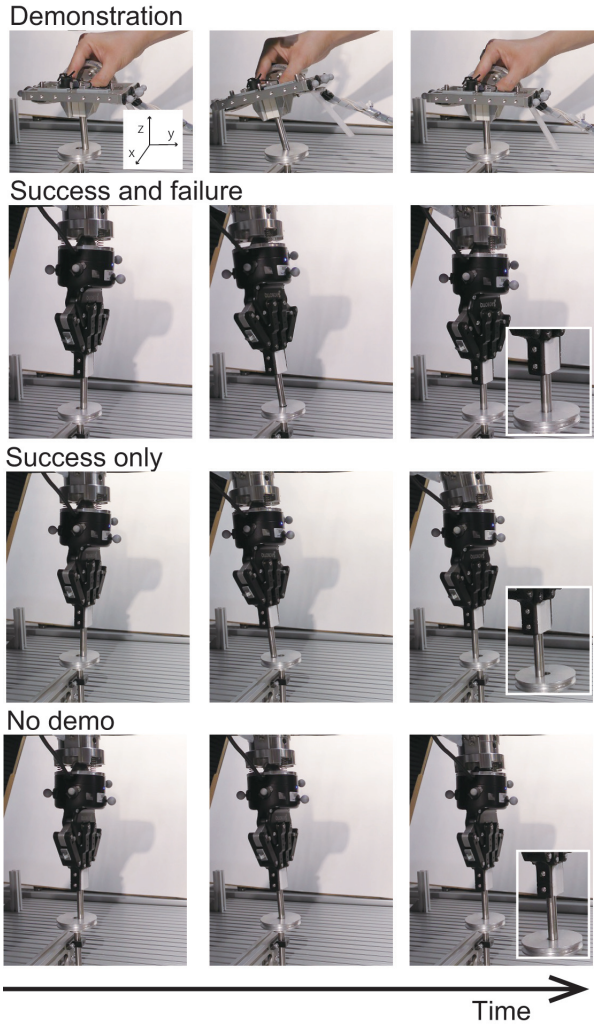

Fig. 6. Snapshots of the demonstrations and the robot's task execution on each condition. The robot successfully imitated the participant's peg-in-hole demonstrations on the success and failure condition. Meanwhile, we observed overshooting on the only success condition and getting stuck on the no demo condition.

with potentiometers, although the data was not used in our experiments because of the peg jig. We attached IR markers for the motion capture system to measure the 6D pose of the device. To match the kinematics between the device and the gripper, we calculated the offset δ. The reference trajectory $\boldsymbol{x}^{\mathrm{ref}}$ was then modified as : $\tilde{\boldsymbol{x}}^{\mathrm{ref}} = \boldsymbol{x}^{\mathrm{ref}} + \delta$.

A participant provided demonstrations with the teaching device 15 times (one demonstration corresponded 10 seconds). Three demonstrations were successful demonstrations and 12 were failed ones (three demonstrations with four directions). In the successful demonstrations, the participant was instructed to imitate the manipulation primitives [31], [18] to help the robot perform the task more successfully. The manipulation primitives consisted of three step: 1) fit the tip of the peg into the hole, 2) align the peg vertically while keeping contacts with the hole, and 3) insert the peg to the bottom of the hole. In the failed demonstrations, a participant was instructed to slide incorrectly in four directions under the assumption that the robot caused overshooting as shown in Fig. 5. The sampling frequency was 5.0 Hz. We applied a low pass filter zero phase shift (second order Butterworth filter, where cutoff frequency was 2.0) to the collected data.

We used a Matlab open-source code for PC-GMM [32] and GMR [33].

We next used PyTorch implementation for PETS [34] The neural network dynamics model consisted of three fully connected layers, 100 neurons per layer with ReLU activation functions. The number of ensembles B was three. The number of particles I to generate action candidates and iterations L were 300 and five, respectively. The top 100 elites were selected to create the next distribution. The predictive horizon H was three. The control frequency was 5.0 Hz to avoid the large computational cost.

For evaluation, we compared the three conditions: 1) using both successful and failed demonstrations and PETS (called success and failure), 2) only successful demonstrations and PETS (called success only; we used only successful demonstrations for PC-GMM), and 3) only PETS with a reward to minimize the x-y-z hole position error (called no demo). We investigated the performance and success rate under

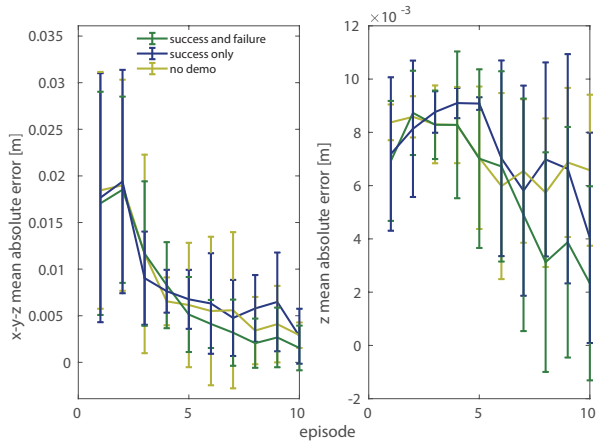

Fig. 7. The absolute mean errors of x-y-z (left) and z (right) positions between the position at the end of each episode and the hole position during 10 learning sessions. The green line is success and failure, the blue line is success only, and the light green line is no demo condition. The success and fail condition performed best.

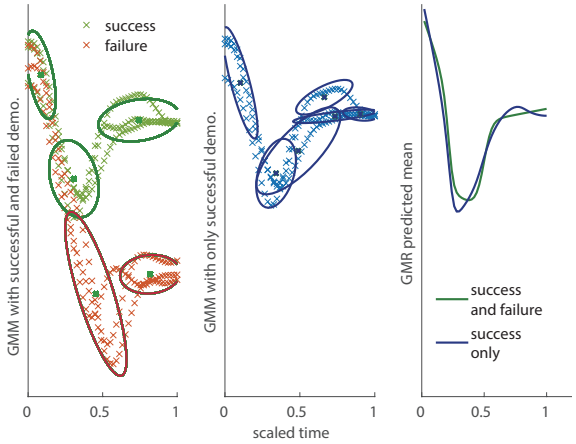

Fig. 8. The fitted GMM assigned to the successful demonstrations (green) and one of the group of the failed demonstrations (red) on the success and failure condition (left), and on the only successful condition (middle). The predicted mean of fitted GMR (right). The GMM with the successful and failed demonstrations was roughly fitted while avoiding each other. Meanwhile, the GMM with the only successful demonstrations was tightly fitted and GMR showed the sharper trajectory.

TABLE I
SUCCESS RATE

Condition	Success and failure	Success only	No demo
Success rate	8/10	5/10	0/10

each condition. The number of episodes for one learning session was 10 and we repeated 10 learning sessions with 10 different random seeds.

B. Results

PC-GMM fitted all successful and failed trajectories with 13 Gaussian components. In the success only condition, it fitted the trajectories with six Gaussian components.

Fig. 6 shows snapshots of the demonstrations and the robot's task execution with learned dynamics. The robot successfully obtained the peg-in-hole strategies by imitating human demonstrations. Meanwhile, we observed overshooting when using only successful demonstrations, and the controller getting stuck when no demonstrations were provided.

In addition, to demonstrate the learning performance, Fig. 7 shows the absolute mean errors of x-y-z and z positions between the position at the end of each episode and the hole position. The green line shows success and failure, the blue line shows success only, and the light green line shows no demo condition. The error bar indicates standard deviations during 10 learning sessions. In all of the conditions, performance improved over the course of the 10 episodes. Using demonstrations resulted in smaller z position errors than no demo, and the success and failure condition shows a smaller z error than the success only condition. The success rates on each condition can be seen in Table I.

From the results, we conclude that using demonstrations yielded better performance than without demonstrations and using both successful and failed demonstrations further improved the performances over using only successful ones.

V. DISCUSSION

In this section, we discuss the experiment results, limitations, and future works.

We obtained better performance utilizing demonstrations than without. Using only PETS showed larger z-position errors since the robot often got stuck at the edge of the hole or during insertion. This is because the reward function used Euclidean distance between the hole position. The robot applied high forces while attempting to move downwards to minimize the error, even though the tip of the peg was not above the hole.

To investigate the performance of our method, Fig. 8 shows the fitted PC-GMM and GMR in y position, which was the direction toward the hole. The x-axes show scaled time. Markers and circles show the collected data and fitted GMM assigned to the successful demonstrations on the success and failure condition (left panel) and the only success condition (middle panel), respectively. In the left panel, we also added the data and GMM assigned one of the group of the failed demonstrations, which represented overshooting towards the y-direction. In the right panel, the lines show the predicted mean of the fitted GMR. On one hand, the GMM with the successful and failed demonstrations roughly fitted to the entire demonstrations while avoiding both the successful and failed ones. On the other hand, the GMM with only successful demonstrations fitted tightly to the data, yielding GMR with a sharper trajectory. This would be the reason for the overshooting. Much biasing to the successful demonstrations did not always yield good performances. Meanwhile, leveraging both success and failure demonstrations could extract necessary strategies to complete the tasks.

As limitations, we fixed the peg into the gripper and the teaching device to reduce the uncertainty of the peg's pose. Instead of using the jig, it would be interesting to use tactile sensors to measure the peg pose and perform in-

hand-manipulation from demonstrations. In this experiment, we designed the number of demonstrations empirically so that they could cover the entire workspace with minimum efforts of the demonstrator resulting in the small number of demonstrations. However, there is room to explore how many demonstrations can affect performances in more detail. To this end, we will increase the number of demonstrations or change the percentages between the number of successful and failed ones as well as increase the number of experiments with multiple participants.

In future works, we will test our framework with multiple peg-in-hole tasks such as different tolerance levels or materials. Moreover, we will extend our method to consider the uncertainty of the peg locations using tactile sensors.

VI. Conclusion

In this study, we proposed the novel system for learning soft robotic assembly strategies. To this end, we employed PC-GMM to exploit the successful and failed trajectories and the deep model-based reinforcement learning to efficiently obtain the robust strategies. We performed experiments using a teaching device and a real robot equipped with a compliant wrist. The experimental results showed that our method utilizing the successful and failed demonstrations could learn strategies with a higher success rate than using only the successful demonstrations.

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," *arXiv preprint arXiv:1904.05240*, 2019.

[2] Y. Karako, S. Kawakami, K. Koyama, M. Shimojo, T. Senoo, and M. Ishikawa, "High-speed ring insertion by dynamic observable contact hand," in *IEEE Int'l Conf. on Robotics and Automation*, 2019, pp. 2744–2750.

[3] J. Luo, E. Solowjow, C. Wen, J. A. Ojea, A. M. Agogino, A. Tamar, and P. Abbeel, "Reinforcement learning on variable impedance controller for high-precision robotic assembly," in *IEEE Int'l Conf. on Robotics and Automation*, 2019, pp. 3080–3087.

[4] T. Nishimura, Y. Suzuki, T. Tsuji, and T. Watanabe, "Peg-in-hole under state uncertainties via a passive wrist joint with push-activate-rotation function," in *IEEE-RAS Int'l Conf. on Humanoid Robotics*, 2017, pp. 67–74.

[5] S. Wang, G. Chen, H. Xu, and Z. Wang, "A robotic peg-in-hole assembly strategy based on variable compliance center," *IEEE Access*, vol. 7, pp. 167 534–167 546, 2019.

[6] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Survey: Robot programming by demonstration," *Handbook of Robotics*, vol. 59, 2008.

[7] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[8] Z. Zhu and H. Hu, "Robot learning from demonstration in robotic assembly: A survey," *Robotics*, vol. 7, no. 2, p. 17, 2018.

[9] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018.

[10] K. Hang, A. S. Morgan, and A. M. Dollar, "Pre-grasp sliding manipulation of thin objects using soft, compliant, or underactuated hands," *IEEE Robotics and Automation Lett.*, vol. 4, no. 2, pp. 662–669, 2019.

[11] N. Jaquier, D. Ginsbourger, and S. Calinon, "Learning from demonstration with model-based gaussian process," in *Conf. on Robot Learning*, 2019, pp. 247–257.

[12] N. Figueroa and A. Billard, "A physically-consistent bayesian non-parametric mixture model for dynamical system learning," in *Conf. on Robot Learning*, 2018, pp. 927–946.

[13] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," in *Advances in Neural Information Processing Systems*, 2018, pp. 4754–4765.

[14] J. Morimoto, "Soft humanoid motor learning," *Science Robotics*, vol. 2, no. 13, p. eaaq0989, 2017.

[15] S.-k. Yun, "Compliant manipulation for peg-in-hole: Is passive compliance a key to learn contact motion?" in *IEEE Int'l Conf. on Robotics and Automation*, 2008, pp. 1647–1652.

[16] R. Li, R. Platt, W. Yuan, A. ten Pas, N. Roscup, M. A. Srinivasan, and E. Adelson, "Localization and manipulation of small parts using gelsight tactile sensing," in *IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, 2014, pp. 3988–3993.

[17] K. Nozu and K. Shimonomura, "Robotic bolt insertion and tightening based on in-hand object localization and force sensing," in *IEEE/ASME Int'l Conf. on Advanced Intelligent Mechatronics*, 2018, pp. 310–315.

[18] M. Hamaya, R. Lee, K. Tanaka, F. Von Drigalski, C. Nakashima, Y. Shibata, and Y. Ijiri, "Learning robotic assembly tasks with lower dimensional systems by leveraging physical softness and environmental constraints," in *IEEE Int'l Conf. on Robotics and Automation*, 2020, pp. 7747–7753.

[19] T. Nishimura, K. Mizushima, Y. Suzuki, T. Tsuji, and T. Watanabe, "Thin plate manipulation by an under-actuated robotic soft gripper utilizing the environment," in *IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, 2017, pp. 1236–1243.

[20] C. Della Santina, V. Arapi, G. Averta, F. Damiani, G. Fiore, A. Settimi, M. G. Catalano, D. Bacciu, A. Bicchi, and M. Bianchi, "Learning from humans how to grasp: a data-driven architecture for autonomous grasping with anthropomorphic soft hands," *IEEE Robotics and Automation Lett.*, vol. 4, no. 2, pp. 1533–1540, 2019.

[21] A. Gupta, C. Eppner, S. Levine, and P. Abbeel, "Learning dexterous manipulation for a soft robotic hand from human demonstrations," in *IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, 2016, pp. 3786–3793.

[22] J. Wang, S. Elfwing, and E. Uchibe, "Deep reinforcement learning by parallelizing reward and punishment using the maxpain architecture," in *IEEE Int'l Conf. on Development and Learning and Epigenetic Robotics*, 2018, pp. 175–180.

[23] T. Kobayashi, T. Aotani, J. R. Guadarrama-Olvera, E. Dean-Leon, and G. Cheng, "Reward-punishment actor-critic algorithm applying to robotic non-grasping manipulation," in *IEEE Int'l Conf. on Development and Learning and Epigenetic Robotics*, 2019, pp. 37–42.

[24] D. Esteban, L. Rozo, and D. G. Caldwell, "Learning deep robot controllers by exploiting successful and failed executions," in *IEEE-RAS Int'l Conf. on Humanoid Robots*, 2018, pp. 1–9.

[25] K. Shiarlis, J. Messias, and S. Whiteson, "Inverse reinforcement learning from failure," in *Int'l Conf. on Autonomous Agents & Multiagent Systems*, 2016, pp. 1060–1068.

[26] Y. Gao, H. Xu, J. Lin, F. Yu, S. Levine, and T. Darrell, "Reinforcement learning from imperfect demonstrations," *Int'l Conf. on Learning Representations*, 2018.

[27] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intelligent Service Robotics*, vol. 9, no. 1, pp. 1–29, 2016.

[28] A. Nagabandi, K. Konoglie, S. Levine, and V. Kumar, "Deep dynamics models for learning dexterous manipulation," in *Conf. on Robot Learning*, 2019, pp. 1101–1112.

[29] A. Tahirovic and B. Lacevic, "Possibilities of the cross-entropy method usage in the control theory," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 7796 – 7801, 2008.

[30] "MoveIt," https://moveit.ros.org/.

[31] L. Johannsmeier, M. Gerchow, and S. Haddadin, "A framework for robot manipulation: skill formalism, meta learning and adaptive control," in *IEEE Int'l Conf. on Robotics and Automation*, 2019, pp. 5844–5850.

[32] "phys-gmm," https://github.com/nbfigueroa/phys-gmm.

[33] "Gaussian Mixture Regression (GMR)," https://jp.mathworks.com/matlabcentral/fileexchange/19630-gaussian-mixture-model-gmm-gaussian-mixture-regression-gmr.

[34] "probabilistic-model-based-rl," https://github.com/rlee3359/probabilistic-model-based-rl.