Compliant Robust Control for Robotic Insertion of Soft Bodies

Yi Liu[1], Andreas Verleysen[1], Francis wyffels[1]

***Abstract*— This paper proposes a novel framework for insertion-type tasks with soft bodies, such as cleaning a bottle with a soft brush. First, a multimodal model based on vision and force perception is trained. Domain randomization is used for the soft body's properties to overcome the simulation-to- reality gap. Second, we propose a dynamic safety lock method based on force perception, which is embedded in the training model to make sure that the tool explores and traverses the hole's path in a compliant way. This result in a higher success rate without damaging the tools/holes. Finally, we perform experiments in simulation and the real world, and the success rate of our proposed method reaches 85.14% in simulation and 83.45% in the real world. Ablation experiments in the real world demonstrate that our method is effective for complex paths and soft bodies with varying deformation intensities. Videos and code are supplied in https://0707yiliu.github.io/SoftBodyInsertion/.**

***Index Terms*—Robot Safety, Reinforcement Learning**

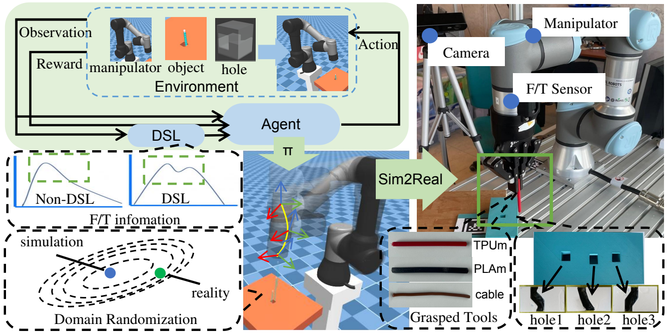


Fig. 1. The overview of the robotics soft bodies insertion policy. The proposed method Dynamic Safety Lock (DSL) changes the force information through motion compensation (III-D). Then, the generated policy by the domain randomized agent is transferred to the real robot (Sim2Real) (IV-A), which can adapt to different tools and holes (IV-B).

I. Introduction

DAILY insertion-type tasks, particularly, soft body insertion tasks where a flexible rod is inserted into the hole, such as cleaning a bottle with a soft brush, are non-trivial for humans because the unknown internal structure of the target and the difficulty of soft body manipulation. The robots can be endowed with precise force and visual perception capabilities for completing the task.

A vision-based model can be used on robots, utilizing cameras to identify the positions of tools and holes [1], [2]. Nevertheless, this model lacks the ability to determine the necessary force for soft bodies. Utilizing force sensing to offer localized feedback during collision or contact facilitates precise and secure control of the insertion process [3], [4], but purely force perception lacks global observability, leading to frequent collisions during exploration. Integrating vision, force perception, and other information to construct a multimodal model allows the robot to capture comprehensive environmental details [5]. Compared with the manipulation of rigid bodies, manipulating soft bodies requires a robot to dynamically adjust its trajectory based on the real-time soft body characteristics perception [6]–[8]. Model-based approaches have utilized the physical stiffness of soft structures to control the directionality of forces shared with the environment for insertion tasks [9]. However, the non-uniform deformability of most soft bodies, dependent on material and internal structure, presents a challenge for model-based control systems. Opting for reinforcement learning (RL) methods with strong generalization capabilities, as suggested by [10], is advisable for generating policies that enable robots to complete the task [11]–[14]. The challenge with the policy lies in the fact that different deformation intensities impact the model. The policy needs to be generalized to empower the robot to adapt to various soft bodies with differing deformation intensities. Furthermore, exploring RL causes the robot to collide with the environment, minimum contact force would be desirable because the large contact force makes large deformation of the object, resulting in task failure. We consider the force as "safety" in this paper. Therefore, the safety of robot interaction [15]–[17] deserves consideration.

To solve the safety issue in soft body insertion tasks. We propose a policy that exploits multiple types of sensors as shown in Fig. 1. The RL policy π requires low environmental refinement and remains robust to external disturbances. The policy learns the joint representation of the force/torque (F/T) sensor and vision information to obtain the motion trajectory. The force sensor is mounted on the wrist to measure the torque and feedback force returned by the hand. Moreover, as a result of the training, the representation of the high-dimensional data generated by the model is utilized as a policy input to the task. Finally, the proposed policy is used on a real robot in combination with domain randomization. We summarize the key contributions as follows:

1) A methodology to learn the soft body insertion tasks with RL based on multimodal information.
2) Insights on how to set physical parameters in the environment to achieve transfer-ability for different soft tools and contexts in both simulation and the real world.
3) We introduce DSL to compensate the robot's motion trajectory to ensure safety and enhance robustness during the insertion process.
4) Demonstrating effective use of the F/T sensor and visual feedback for hole search and insertion. An ablation study is set up to compare the effects of different modalities on task performance.

Manuscript received: January 17, 2024; Revised: March 30, 2024; Accepted: April 28, 2024. This paper was recommended for publication by Editor Clement Gosselin upon evaluation of the Associate Editor and Reviewers' comments. (*Corresponding author: Yi Liu.*)

[1] Yi Liu, Andreas Verleysen and Francis wyffels are with IDLab-AIRO, elis, Ghent University-imec, Ghent, Belgium (email: yiyiliu.liu@ugent.be)

Digital Object Identifier (DOI): see top of this page.

II. Related Work

A. Multimodal Model for the Insertion-type Task

The policy for insertion-type tasks [6], [18], [19] typically relies on visual feedback or force control. Previous vision-based research [1] has been applied to the insertion task. A vision-based model was used to estimate the insertion error and improve the insertion accuracy, making the model suitable for the insertion of cylinders [2]. A model based on high-speed motion strategy and high-speed camera was used for robotic needle-threading tasks [7]. They used visual servoing to determine and match the holes, but they could not make the robot interact with the environment smoothly. Some studies accomplished the task through force perception, such as blind search alignment based on spiral force [3], alignment based on controllers with 6-degree-of-freedom (DoF) F/T sensors [4], using a recurrent neural network with long short-term memory cells to predict contact force to complete soft body insertion [8]. These force-based methods were adapted to specially shaped tools but are inefficient and require precise settings of environment and model parameters. Others build multimodal models with rich environmental information for rigid objects [11], for instance, fusing multi-sensor data for noisy observations and using Bayesian estimation to complete the assembly of the plug [5]. These multimodal models combined the advantages of different physical information and enabled the robot to interact smoothly with the environment under the premise of quickly obtaining the overall information of the environment.

B. Reinforcement Learning for Insertion

RL uses simulations to generate the required interaction data [14]. The RL setting consists of an agent interacting with the environment, which is built as a Markov decision process with a state space $\mathcal{S}$ and an action set $\mathcal{A}$. At each time step, the agent obtains the observation s_t, then takes the action a_t, and obtains the reward r_t.

Previous work [12], [13], [20] used RL algorithms to adjust the gain matrix of the controller for insertion. [21] learned robot skill parameters through simulation based on different types of algorithms so that both simulated and real robots have insertion skills. [22] deployed the robot with multimodal representation and built up encoders to control the robot based on the Trust Region Policy Optimization (TRPO) [23] algorithm. The trajectory generated by the model was not constrained and allowed the robot's end-effector (EEF) to perform exploration on the surface of holes. [24] learned to aggregate source dynamics models adaptively to better fit the state-transition dynamics of target environments and execute optimal actions there. [25] used the gradient-based Proximal Policy Optimization (PPO) [26] algorithm on robot in automotive insertion tasks. These works had good robustness and used a variety of sensors. RL skillfully integrated multiple pieces of information, enabling the robot to effectively handle insertions in diverse situations.

C. Domain Randomization for Simulation to Reality

Domain randomization (DR) is an approach to overcome the reality gap by optimizing the parameter distributions [27]. It achieved a good simulation-to-reality (Sim2Real) performance by covering a range of parameters distribution in the simulation containing the real values during RL training. These efforts, while effective, had the same drawback, this method required significant engineering effort in tuning the random range, which was difficult and non-intuitive [28].

In addition to the fundamental approach, some works applied the approach in the context of existing soft bodies. Some employed finite element methods (FEM) to build simulation environments [29], and evaluate the performance of the soft body by observing the feedback of the F/T sensor. DR had a good performance in cooperating with observable devices like F/T sensors, which improved the analysis value of simulation for soft bodies. However, it's not practical to use refined FEM in the environment of RL because FEM consumes a lot of computing resources in the simulation. [30] designed a local Graph Neural Network with DR to speed up the training speed and increase the simulation accuracy. [31] constructed a platform with RL, which optionally used non-FEM to speed up the training process, but the simulator had poor compatibility with RL algorithms, and the interaction information between the rigid body and the soft body was not accurate, which increased the difficulty of Sim2Real.

III. Methodology

In this section, we describe the environment setup for the RL model and soft body simulation. We utilize DR for Sim2Real, and introduce the DSL method, which adjusts the robot's motion based on force feedback.

A. Insertion Task Setting in Simulation

In Fig. 1, the simulation training environment consists of holes, tools, and a 6-DoF manipulator with a gripper. Since our focus is on the insertion action, the initial EEF position is fixed, and the tool is fixed on the EEF. The target (hole) location is randomized within a defined domain that the EEF can reach.

Then, the designed hole has requirements, focusing on inserting complex pipeline paths. The designed holes' internal paths, shown in Fig. 1, are not straight, the slope of the designed holes is used to simulate the complex path.

Lastly, the tools used in this task are soft bodies, and we employ a non-FEM approach to design them, simplifying the complexity of deformation calculation. Illustrated in Fig. 3(a), we utilize a continuum-like spring-damper model to simulate the soft bodies, granting elasticity in all directions and reducing computational overhead during simulation. To reduce the Sim2Real gap, we set the DR for soft bodies in the simulation as shown in Fig. 1. In particular, the prime characteristic of a soft body is deformability. We randomize the damping and stiffness parameters of the soft body within a specified range in the simulation (IV-A1), where the range is obtained by testing the deformation performance of the soft body. As shown in Fig. 3(b), we adjust the stiffness and damping parameters to ensure that the tool does not embed in the obstacle as much as possible to obtain its maximum value. We ensure that the tool does not move away immediately after contact to obtain its minimum value.

B. Observation, Action, and Reward

The RL state s_t consists of the robot state s_t^r, and the insertion task state s_t^a. This model needs to search the path of the hole by rotating the EEF, therefore, the robot state s_t^r contains the robotic EEF position $\mathbf{p}_t^{ee} = [x_t^{ee}, y_t^{ee}, z_t^{ee}]$ and Euler's rotation $\mathbf{r}_t^{ee} = [rx_t^{ee}, ry_t^{ee}, rz_t^{ee}]$, and the force and torque obtained with the F/T sensor $\mathbf{f}_t^{ee} = [F_{xt}, F_{yt}, F_{zt}, \tau_{xt}, \tau_{yt}, \tau_{zt}]$ in EEF. The task state s_t^a contains the hole position $\mathbf{p}_t^h = [x_t^h, y_t^h, z_t^h]$. In particular, this model does not observe the rotation of the hole, but the Euler angles of the holes are randomized in the simulation

(IV-A1) to improve the generalization ability of the model. In addition, in combination with III-A, the initial value of $\mathbf{p}_t^{ee}$ is constant in the state sequence $s_t = [s_t^r, s_t^a]$, which does not cause any difficulties while in Sim2Real. The hole position $\mathbf{p}_t^h$ is endowed with domain randomization and is randomized at each episode and the rotation of the hole is randomized to simulate paths with varying degrees of inclination. Thus, the relative position $\mathbf{p}_h^{ee}$ and $\mathbf{p}_t^h = [x_t^h, y_t^h, z_t^h]$ are obtained and calculated by the camera so that the model can be applied in the real world to any position reachable by the robot.

The action a_t consists of the 3-dimension displacement increment of the robot EEF $[\Delta x, \Delta y, \Delta z]$ and the 3-dimension rotation increment $[\Delta\theta x, \Delta\theta y, \Delta\theta z]$, the latter being the Euler angles of the EEF for the intuitive explanation.

Reward shaping is an important part of RL, which affects the quality of the model obtained through RL. The purpose of the discussed task is to let the tool pass through the hole to reach the target, hence, for the reward function r_t required for the RL model, we choose Euclidean distance as the basic function, i.e., the distance between the hole and the tool. The observation s_t does not include the position of the tool but fixes the tool to the EEF (III-A) because of the difficulty of observing soft bodies. Therefore, the Euclidean distance between the hole and the EEF d_h^e is calculated. Furthermore, the maximum number of steps n_m in each episode is given in RL, and there should be higher rewards for success with a small number of steps. Finally, after successfully reaching the goal, it should stop instead of gliding and continue forward. Therefore, we add an integral term to the reward function to accumulate the number of consecutive successes. The overall structure is as follows:

$$\begin{aligned} r_t = &-\alpha_1(x_t^{ee} - x_t^h) - \alpha_2(y_t^{ee} - y_t^h) - \alpha_3(z_t^{ee} - z_t^h) \\ &- \alpha_4(\tanh \sum 1_{d<\delta_1} \sqrt{\left|\mathbf{p}_t^{ee} - \mathbf{p}_t^h\right|^2} - 1) \\ &- (\tanh(n_m - n_c) - 1), \end{aligned} \tag{1}$$

where $[\alpha_1, \alpha_2, \alpha_3]$ represents the weights associated with different directions, α_4 denotes the weight assigned to the integral term; n_c represents the current step number; d represents the Euclidean distance between $\mathbf{p}_t^{ee}$ and $\mathbf{p}_t^h$, $1_{d<\delta_1}$ is the indicator function of $d < \delta_1$. The last two items use the arctangent function to increase the rate of change.

C. Training

From the RL algorithm mentioned in II-B, it can be seen that PPO is an algorithm that directly updates the agent policy $\pi : \mathcal{S} \to \mathcal{A}$. Since the trajectory of the robot in this work is continuous, this paper is considered to use this algorithm.

The corresponding network architecture is shown in Fig. 2 (the part of Dynamic Safety Lock is described in III-D, the details of the network are described in this section). The agent network contains three fully connected layers. The output of the task's observation is merged with the observation of the robot. The concatenated data vector is processed through the fully connected layers and the activation function arc-tangent function. The final output is the displacement increment a_t of the EEF. The goal of the PPO algorithm is to learn a policy π that maximizes the reward. Particularly, in order to ensure the value network and policy network have the same structure, our embedded DSL does not change the network structure.

D. Dynamic Safety Lock

To avoid severe collisions on the robot that cause system crashes, object damage, etc., we suggest a DSL method. As shown in Fig. 2, initially, the 6-DoF force signals for collision detection are generated by the F/T sensor installed on the EEF. These signals serve as input for the DSL. The DSL's objective is to enable the EEF to perform the insertion action compliantly. Due to the uncertainty in the desired force for various soft bodies, we use force changes iteratively to test the required force for the soft bodies. The DSL employs these force changes to generate dynamic weights $[\beta_x, \beta_y, \beta_z]$ and $[\beta_{rx}, \beta_{ry}, \beta_{rz}]$,

$$\begin{aligned} [\beta_x, \beta_y, \beta_z] &= [F_{x(t)}, F_{y(t)}, F_{y(t)}] - [F_{x(t-1)}, F_{y(t-1)}, F_{y(t-1)}] \\ [\beta_{rx}, \beta_{ry}, \beta_{rz}] &= [\tau_{x(t)}, \tau_{y(t)}, \tau_{y(t)}] - [\tau_{x(t-1)}, \tau_{y(t-1)}, \tau_{y(t-1)}], \end{aligned} \tag{2}$$

which are used as the gain to calculate the force compensation value $a_t^{'} = [\delta p_x, \delta p_y, \delta p_z, \delta r_x, \delta r_y, \delta r_z]$ with the element-wise product operator:

$$\begin{aligned} [\delta p_x, \delta p_y, \delta p_z] &= [\beta_x, \beta_y, \beta_z] \odot [F_{x(t)}, F_{y(t)}, F_{y(t)}] \\ [\delta r_x, \delta r_y, \delta r_z] &= [\beta_{rx}, \beta_{ry}, \beta_{rz}] \odot [\tau_{x(t)}, \tau_{y(t)}, \tau_{y(t)}]. \end{aligned} \tag{3}$$

The force change value and the force compensation value are positively related. The unit of $a_t^{'}$ is *Newton* (N). we obtain the action compensation value through Newton's second law $F = ma$. The mass of the object and the time step can be set in the simulation and are fixed values. Additionally, we apply a compensation weight to modify the displacement change's intensity. This adjustment aims to ensure that the tools attempt collision rather than mere rubbing after contact. Hence, the above parameters (fixed value and weight) can be integrated into hyper-parameter α_5, whose unit is s^2/kg. Its physical meaning is compliance, which represents the displacement generated by unit force at the contact point. To allow it to be embedded in the RL network without affecting its distribution of the on-policy. We make $a_t^{'}$ equivalent to the output layer of the RL network and use the activation function to scale to obtain displacement compensation value $a_t^{''} = \tanh(\alpha_5 a_t^{'})$.

Finally, to more concretely represent the function of the DSL, we list the pseudo-algorithm as shown in Algorithm 1,

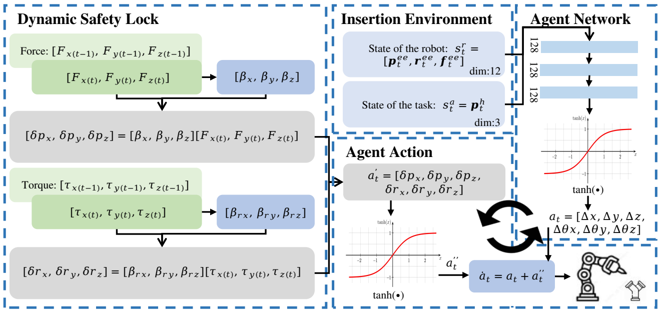

Dynamic Safety Lock
Force: $[F_{x(t-1)}, F_{y(t-1)}, F_{z(t-1)}]$
$[F_{x(t)}, F_{y(t)}, F_{z(t)}]$
$[\beta_x, \beta_y, \beta_z]$
$[\delta p_x, \delta p_y, \delta p_z] = [\beta_x, \beta_y, \beta_z][F_{x(t)}, F_{y(t)}, F_{z(t)}]$
Torque: $[\tau_{x(t-1)}, \tau_{y(t-1)}, \tau_{z(t-1)}]$
$[\tau_{x(t)}, \tau_{y(t)}, \tau_{z(t)}]$
$[\beta_{rx}, \beta_{ry}, \beta_{rz}]$
$[\delta r_x, \delta r_y, \delta r_z] = [\beta_{rx}, \beta_{ry}, \beta_{rz}][\tau_{x(t)}, \tau_{y(t)}, \tau_{z(t)}]$
Insertion Environment
State of the robot: $s_t^r = [\boldsymbol{p}_t^{ee}, \boldsymbol{r}_t^{ee}, \boldsymbol{f}_t^{ee}]$
dim:12
State of the task: $s_t^a = \boldsymbol{p}_t^h$
dim:3
Agent Network
128
128
128
tanh(z)
tanh(•)
$a_t = [\Delta x, \Delta y, \Delta z, \Delta\theta x, \Delta\theta y, \Delta\theta z]$
Agent Action
$a_t^{'} = [\delta p_x, \delta p_y, \delta p_z, \delta r_x, \delta r_y, \delta r_z]$
tanh(•)
$a_t^{''}$
$\dot{a}_t = a_t + a_t^{''}$

Fig. 2. Network structure of robot insertion models. The entire network is divided into three parts, the environment, the agent network (III-C), and the dynamic safety lock (III-D).

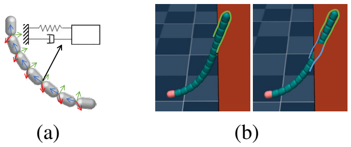


Fig. 3. Tests on soft tools with varying stiffness and damping. (a) is the theoretical diagram of the soft body in simulation, and each spherical joint is a spring-damper system. (b) is the comparison of the soft body in simulation, the left is a representation with reasonable stiffness and damping, and conversely, the right is irrational.

Algorithm 1: Dynamic Safety Lock.

```
Input:
  • $\mathbf{F_t}$=$[F_{x(t)}, F_{y(t)}, F_{y(t)}, \tau_{x(t)}, \tau_{y(t)}, \tau_{z(t)}]$
  • $\mathbf{F_{t-1}}$=$[F_{x(t-1)}, F_{y(t-1)}, F_{y(t-1)}, \tau_{x(t-1)}, \tau_{y(t-1)}, \tau_{z(t-1)}]$
  • $a_t$=$[\delta x, \delta y, \delta z, \delta\theta x, \delta\theta y, \delta\theta z]$
Output: Compensation value $\dot{a}_t$ for the EEF.
// the observation can be provided for this algorithm.
Define the six-dimensional force threshold $\boldsymbol{\delta}\mathbf{f}$
Normalized the F/T sensor data
if $-\boldsymbol{\delta}\mathbf{f} < \mathbf{F_t} < \boldsymbol{\delta}\mathbf{f}$ then
    $\dot{a}_t = a_t$
else
    Get the sequence difference of the F/T sensor
      $\mathbf{F_{t-1}} - \mathbf{F_t}$
    Obtain the agent action compensation value $a_t^{'}$
      through the difference weight of Equation 3
    Copy the activation function of the output layer of
      the policy network, and regularize the
      compensation value $a_t^{'}$ to obtain $a_t^{''}$.
    $\dot{a}_t = a_t + a_t^{''}$
```

where we set a threshold $\boldsymbol{\delta}\mathbf{f}$ for the F/T sensor, which functions as a clipping filter to clear low-level noise of $\mathbf{F_t}$.

Last but not least, compared to the common model that sliding exploration on the target surface, this model with DSL is characterized by frequent presses and touching the target. When the edge of the hole is not explored, because of the bending of the soft body, the former hardly rebounds and continues to press, while the latter can compensate for the displacement and reduce the degree of pressing, making it easier for the soft body to move and search for the hole. When exploring the edge of the hole or the path inside the hole, the former can only determine the next action of the agent based on the observation state without compensation, while the latter has a compensation gain to obtain the more obvious observation.

IV. Experiments and discussion

We aimed to build a policy for inserting soft bodies into complex paths. The experiments were divided into two parts: first, an ablation study in simulation to assess the DSL-based model's contributions, and second, applying the RL-based model on a real robot for the insertion task. Finally, all experimental results were thoroughly discussed.

A. Simulation Experiments

All training and testing in the simulation part were on the Intel(R) Core(TM) i7-1265U CPU. We tuned the hyper-parameters manually to obtain the configuration. In particular, since many factors such as friction, quality, and other settings in simulation affect the performance of the agent, the hyper-parameters we provided serve as reference values. For the reward function (Equation (1)), $\alpha_1 = \alpha_2 = 0.7$, $\alpha_3 = 0.6$, $\alpha_4 = 0.9$, δ_1=1e-4. We explain their importance next. Because the insertion task needed to first explore the location of the hole, the weight for the X-axis and Y-axis was greater than the Z-axis. α_4 was the weight of the cumulative number of successes, which took effect when the tool was close to the required target, indicating that the task had been completed at this step, hence, its value was greater than the first three. The last item in Equation (1) was the step counter, its purpose was to allow the agent to quickly obtain the optimal trajectory, therefore, its weight was the highest. For the DSL, $\alpha_5 = 130$, which was determined by the degree of force feedback and compensation. As shown in III-D, the DSL set the final contact force based on α_5 and the dynamic weights $[\beta_x, \beta_y, \beta_z]$ and $[\beta_{rx}, \beta_{ry}, \beta_{rz}]$. The latter was the change in force. When the value of the latter was small, the compensation value was small, causing the tool to rub on the target surface. The function of the former was to adjust the compensation intensity so that the tool could repeatedly contact the target. Therefore, α_5 was determined based on the deformation strength of the researcher's tool, which needed to be adjusted by the researcher. The recommended value we give allowed our softest tool to make repeated contact with the target. For the observation, to make the simulation environment robust, the Gaussian noise $\mathbf{G}(\cdot)$ was added to the observations of the holes for the experiments. The observation of the robot s_t^r had the noise of the simulator, therefore no additional noise was added. For the training model configurations, the total number of training steps was 3e06, and the maximum number of steps per episode was $n_m = 350$. For the task, we randomized the configuration of the hole position and orientation at the beginning of each episode to enhance the robustness and generalization of the model. All models were set with checkpoints and estimated models were generated every 1e04 steps. The estimated models were set up to test the success rate of the trained models. The simulation environment was built by Mujoco.

1) DR in Simulation: Before training the agent, we needed to configure the physical parameters of the simulation environment so that it matches the real world as much as possible. As mentioned above (III-A), part of the observed data had added noise $\mathbf{G}(\cdot)$ to simulate the state of the real world. Filtering resulted in cleaner data, but there was bias in the real world, therefore, we set DR for part of the observation data to reduce the impact of bias when the environment was reset. In the simulation environment, DR needed to be used in the hole's position $\mathbf{p}_t^{ee}$. As shown in Fig. 1, we used DR to make the center position offset to the surrounding encirclement. In order that the hole would not always be vertical in the simulation, we randomized the Euler angle of the hole. In addition, we normalized the observation space to make it easy to map in the real environment (II-C). The EEF's normalization range was three times that of the hole, while the F/T sensor data's normalization range was determined through pressing testing.

Finally, DR was set for the physical parameters of non-FEM soft bodies (III-A). We estimated the range of stiffness and damping by observing the contact between soft bodies and rigid bodies by minimizing collision detection in Mujoco. As shown in Fig. 3(b), using unreasonable stiffness and damping parameters could make soft bodies embedded in the rigid body in simulation. Using the method mentioned in III-A, when we set the stiffness to be greater than 0.1 and the damping to be greater than 0.05, the tool was embedded in the obstacle. When the stiffness we set was less than 0.01 and the damping was less than 0.003, the tool left as soon as it contacted the obstacle. After testing, the stiffness setting range was $r_s = [0.01, 0.1]$, the damping setting range was $r_d = [0.003, 0.05]$.

2) DSL-based Model: The simulation experiment primarily aimed to analyze the effects of the proposed method. Since the proposed method resembled the admittance controller, an RL model with the admittance controller was included as a baseline. Additionally, force penalty, a factor in the reward function, was considered as another baseline.

We named the basic model as the visual-force model (VF). To analyze the effectiveness of the proposed model (VFDSL),

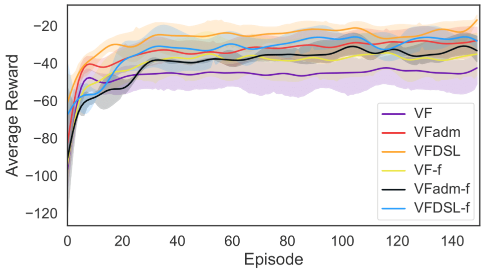


Fig. 4. Training method's performance (DSL vs. admittance controller)

TABLE I
REWARD (R), REWARD VARIANCE (σ(R)) AND SUCCESS RATE (S) IN SIMULATION TESTING.

	VF	VFadm	VFDSL	VF-f	VFadm-f	VFDSL-f
R	-44.12	-28.95	-23.36	-39.57	-37.23	-29.34
σ(R)	1.67	0.72	0.41	4.29	3.77	3.08
S	19.29%	55.86%	85.14%	41.63%	53.17%	79.55%

we added an RL model with an admittance controller (VFadm) as a comparison [20], which followed the control law

$$F_{ext} - F_d = m_d \ddot{x} + b_d \dot{x} + k_d x, \tag{4}$$

where m_d, b_d, and k_d represented the desired inertia, damping, and stiffness matrices respectively. F_{ext} represented the actual contact force vector. F_d represented the desired force. We allowed tools to touch targets because touching enables exploration. x, $\dot{x}$, $\ddot{x}$ were the displacement of the EEF, its velocity, and acceleration respectively. Referring to [20], we set the parameter m_d to be the identity matrix, set the damping ratios to fixed value 100 and added the 6-dimensional k_d of EEF to the mapped action space as the controllable variable, whose range was $[500, 700]$. In order not to affect the results of the ablation experiment, we used the same reward function. In [20], F_d was set to zero as the path was straight, and the tool was a rigid body. However, in our approach, we set $F_d = 0.1$ (normalized), a value close to zero, to encourage the agent to explore complex paths without exerting a substantial influence on actions. Moreover, the RL algorithm was replaced with the same algorithm used in this paper.

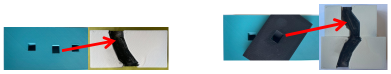
Secondly, a force penalty term was added to the reward function to construct a new reward function as another baseline. Due to the different reward functions, based on the original proposed DSL-based model (VFDSL), three additional models were added (VF-f, VFadm-f and VFDSL-f). The new reward function r_2 added a force penalty term based on r_1.

$$r_2 = -\sqrt{(F_d - F_{ext})^2} + r_1, \tag{5}$$

where the F_d referred to the setting of VFadm. The weight of the force penalty term was 1, indicating that the force was important in this task, other hyper-parameters were adjusted based on their importance. We trained each model 500 times to get TABLE I. The performance of the trained agent was summarized as shown in Fig. 4, VFDSL had the highest reward, followed by VFadm and VF the lowest.

B. Insertion in Real World Environments

We tested the methods in real-world environments, where the hole position was recognized by a ZED2i camera, and all experiments were completed on the device UR3e with a Robotiq-2F85 gripper. We first demonstrated the effectiveness of the DR used (IV-A1) through ablation experiments (IV-B1), which include experiments on the holes' position and soft body parameters. Moreover, the position of the hole was obtained through the AprilTag marker [32] instead of identifying the hole. The second experiment (IV-B2) was to demonstrate the effect of our proposed method.



(a) original hole (b) offset hole

Fig. 5. The offset of the hole. (a) is the original hole, and (b) is one case of the offset hole defined in IV-B1. The right side of each sub-figure is the cross-section inside the hole, and the cross-section in (b) is one of the cases.

TABLE II
THE RESULT OF SUCCESS RATE, Z-FORCE (ZF) AND ITS VARIANCE (σ(ZF)) IN REAL-WORLD ENVIRONMENTS.

	VF	VFadm	VFDSL	VF-f	VFadm-f	VFDSL-f
Zf	0.769	0.685	0.527	0.604	0.573	0.551
σ(Zf)	0.035	0.018	0.112	0.079	0.091	0.140
S	11.79%	49.49%	83.45%	31.07%	43.21%	71.79%

1) Validity Test for Sim2Real: The purpose of this experiment was to prove the effectiveness of DR for Sim2Real, which only used the proposed VFDSL.

We added the DR for the position of holes to reduce the Sim2Real gap. We made a gap between the position in the observation space and the real position as the offset hole, which we call *l2hole*. As shown in Fig. 5, the camera obtained the location of the original hole, but did not know where the offset hole is. Similarly, the path of the *l2hole* had a slope so that the entire path became a complex path, such as an S-shape or a protruding blocking surface. The *l2hole* was utilized in the whole real world experiment for comparison. Then, we tested the hole in the middle, and fixed the hole position, manually changing the position of the upper hole each time. We tested 50 times, the success rates of searching for holes using the DR-based method and the method without DR were 78% and 26% respectively. The success rate for searching was defined when a hole was successfully located and entered, without the need to explore paths. As shown in the part of position DR by the human in Fig. 6, it showed 12 positions, all offset from the original hole position, as we focused on DR's impact on position. More detailed comparisons with the model without offset setting can be found on the GitHub page mentioned in the abstract. The curve on the right showed the EEF's motion in the z-axis direction, which was most representative as it reflected the largest change after searching for the hole.

For the soft body, as shown in Fig. 1, we used tools with varying deformation intensities, which were the cable, the 3D printed stick based on the thermoplastic polyurethanes flexible material (TPUm), and 3D printed stick based on the poly lactic acid material (PLAm). As shown in Fig. 6, for comparison, we trained a model without DR for soft body parameters (the DR model vs. the non-DR model). For the tool, the cable was a 3*mm* diameter enamel-insulated wire, and both TPUm and PLAm were 6*mm* diameter rods with 10% filled, the length of all tools was set to 11*cm*. We tested each of the three tools at the same hole 20 times. The success rate for insertion was 78.33% for the DR model and 31.67% for the model without DR. Specifically, unlike the success rate for searching, the insertion success rate was defined as when the tool appeared at the bottom after passing through the hole. The determination of whether the tool appeared at the bottom was judged manually, and the manipulator

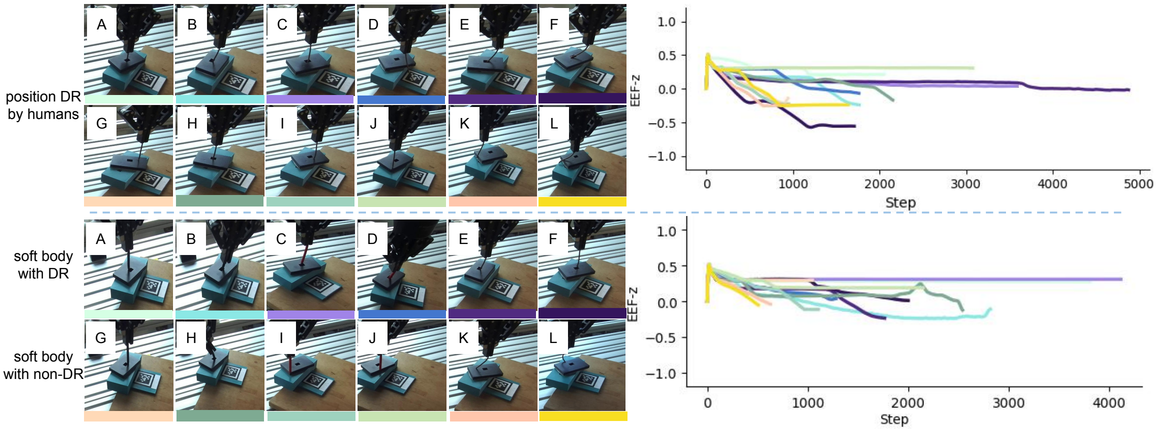


Fig. 6. Offset experiment for the hole position and DR experiment for the soft body (refer to the video from the link in abstract). The former experiment only utilizes the cable as the tool, while the latter one employs three types of tools. The colors below the 12 subgraphs correspond to different colors in the curves, with *A* to *L* used as their aliases in discussions. These graphs mainly show the state of the EEF in the z-axis direction, which has been normalized.

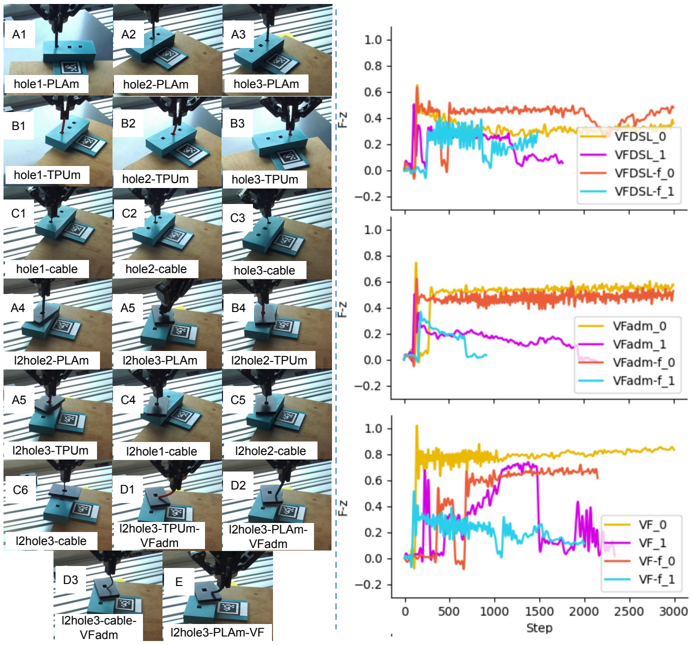


Fig. 7. Performance of insertion experiments. It encompasses all cases related to VFDSL (from series *A* to series *C*) as well as partial cases of VFadm (series *D*) and VF (*E*) on the left part. The curve on the right represents the performance of each model exploring *hole*2 using the TPUm-based tool, where for example VF_0 indicates that the task failed and VF_1 indicates that the task succeeded.

automatically stopped when EEF was below the surface of the hole. All the holes we designed were square with 9*mm* side length. The inclination angle of the paths ranges randomly between 55 and 65 degrees, and the paths feature irregular protrusions, the vertical thickness of the hole was 3*cm*.

2) Soft Bodies Insertion: The main feature of the proposed method was to solve the problem of soft body insertion. The insertion experiment designed in this section included the attempt of unexplored hole paths, the verification of the use of DR for soft bodies set in the simulation (IV-A1), and the comparison of agents trained with different methods (TABLE II). Initially, we designed hole paths with varying complexity, as shown in Fig. 1, in the real world. All three paths differed from the path simulated. Next, to confirm the applicability of the proposed model with different soft bodies, we utilized three tools with varying stiffness for comparison. Then, we complicated the hole path by adding *l2hole* to reflect the ability of the proposed model to explore the hole paths. Finally, as shown in Fig. 7, the main focus was on demonstrating the performance of the proposed VFDSL for different hole paths and tools with varying stiffnesses. We showed a portion of the force curves on the right, which included both successful and failed cases of inserting the TPUm-based tool into *hole*2 in the z-axis direction for all models we mentioned. As shown in TABLE II, Zf represented the normalized force in the z-axis direction in the observation space, which objectively reflected the strength of the manipulator pressing, including searching for holes and paths in holes. We tested each method and path 20 times, however, *hole*1 was ineffective for the PLAm-based tool and the TPUm-based tool because the diameter was larger than the diameter of the path during inclination. Therefore, we tested 280 times for each method, which included *hole*1, *hole*2, *hole*3 as shown in Fig. 1 and their *l2hole* as the example shown in Fig. 5(b), where the interface inside *l2hole* offered irregular shapes.

C. Discussion

1) Simulation Results: We first analyzed and discussed the simulation results. For the simulation experiment IV-A2, as shown in Fig. 4 and TABLE I, the reward and success rate of the VFDSL was the highest. The r_1 expressed that the shorter the trajectory in an episode, the higher the reward, which shows that the VFDSL was the most efficient when searching for the hole and the path in the hole. VF had a low success rate (19.29%), implying ineffective path exploration, while VFadm showed a significant improvement (55.86%), highlighting the importance of force feedback in task performance, which was reflected in the comparison between VF and VF-f (41.63%). VFDSL surpassed VFadm in performance (85.14% vs. 55.86%), thanks to its dynamic desired force that converged to feasible contact between the soft body and the environment. In contrast, VFadm, employing a fixed desired force, exhibited a lower success rate. For the model with different reward functions, we did not compare the reward due to different calculation methods of reward, but obviously, as shown in Fig. 4, due to the force penalty term, r_1 was higher than r_2. In VF-f, VFadm, and VFadm-f, the latter two shared a similar (55.86% and 53.17%), higher success rate compared to the former. This implies that both the force feedback from the admittance controller and the force penalty in the reward function had similar effects. In VFDSL and VFDSL-f, the success rate of the former was slightly higher than the latter one (85.14% vs. 79.55%). This discrepancy arose because the fixed F_d in the force penalty was not accurate for all tools, resulting in a negative impact. In short, the force feedback played a vital role in this task, however, because the soft body's desired force was uncertain, a fixed force could not be applied to all tools and might even have negative effects. The dynamic desired force obtained by interacting with the environment worked better.

2) Performance of DR: We discussed the results of Sim2Real experiment IV-B1, which included whether the model utilized DR regarding hole positioning and the properties of the soft body. On the one hand, as shown the position DR by humans in Fig. 6, we could see that some curves (F, H, K, L) drop directly below the hole position, indicating that it directly searched for the hole, some curves (A, B, C, D, E, J) stagnated in an area and then descended, indicating that they searched for the hole position after trying to search on the surface of the hole, and other curves (G, I) stuck in an area, indicating that it's trying to search, but it's unsuccessful. Moreover, each hole was offset, and after 50 independent experiments, the success rate of the method based on DR was higher than that without DR (78% vs. 26%). This indicates that DR has stronger resistance to interference in position information and can more efficiently locate the hole position. Hence, using DR for the hole position in the simulation could reduce the Sim2Real gap of position detection. On the other hand, regarding the impact of DR on soft body parameters, quantitatively, under identical conditions, the success rate of the method based on DR was higher than that without DR (78.33% vs. 31.67%), indicating that DR was helpful for Sim2Real of soft bodies in this work. Qualitatively, as shown the soft body part in Fig. 6, it could be seen that both models can search for the hole when clamping the PLAm-based tool (A, B vs. E, H), and the curve on the right also corresponds to it, which is because the PLAm-based tool was the most rigid among the three tools. However, when switching to the softer TPUm-based tool (C, D vs. I, J), the non-DR model moved away from the hole after contact, while the DR model still searched near the hole. Particularly, the dip of some curves like J curve was caused by the wrong search, which meant that not all descending curves indicate success. When switching to the cable tool (E, F vs. K, L), the non-DR model deformed the cable directly before searching, while the DR model only slightly deformed the cable. From the curve perspective, it could be observed that the curve of the DR model gradually descends, whereas the curve of the non-DR model directly descends, but it is not successful. To sum up, when transferring the model from simulation to the real world, the DR for hole position helps improve the model's robustness, and the DR for soft body parameters enables the agent to adapt to tools with varying deformation intensities.

3) Performance of Insertion: We discussed the results of soft body insertion in the real world. The partial performance was shown in Fig. 7, it could be seen that for the PLAm-based tool, the VFDSL did not deform it (series A), the VFadm bent it slightly ($D2$), which meant that when the desired force setting of the latter was inappropriate, it would damage the tool due to the tool's excessive stiffness, this kind of damage was more pronounced in the model (E) without force feedback. For the softer TPUm tool, both models show more pronounced distortion of the tool (series B vs. $D1$). For the softest tool, the cable, VFadm caused deformation even before locating the hole, whereas VFDSL repeatedly made contact attempting to locate the hole's position (series C vs. $D3$). Next, we discussed the force-containing models. From the curve in Fig. 7, it could be seen that compared to VF, whether it was due to the failure of searching the hole (VF_0 vs. VF-f_0) or the successful exploration of the path (VF_1 vs. VF-f_1), the force of VF-f was smaller. This was the advantage of the force penalty term. Comparing the curves of VFadm and VFadm-f, it could be observed that in the failure cases (VFadm_0 vs. VFadm-f_0), the force stayed within a similar range, and in the successful cases (VFadm_1 vs. VFadm-f_1), the force during path exploration was similar. This indicated that the effect of the force penalty term was reduced. Specifically, when the force penalty term was assigned an unsuitable value F_d for the tool, VFDSL-f exhibited failure, with its value exceeding VFDSL (VFDSL_0 vs. VFDSL-f_0). After locating the hole, significant fluctuations occurred during path exploration (VFDSL_1 vs. VFDSL-f_1).

Quantitatively, we could see that the VFDSL completed the task with a small force (0.527 ± 0.112), and the success rate (83.45%) was the highest among the three models in TABLE II. This was because it had a large compensation after contact, and it searched for the hole by leaving the surface of the hole and pressing frequently. Compared with the VF, the VFadm had a higher success rate (11.79% vs. 49.49%) because it had a certain position compensation, but it still made tools rub against the surface of the hole. For VF and VF-f, the latter exhibited a markedly enhanced success rate at 31.07%, underscoring the substantial impact of force feedback on task performance. Nevertheless, the success rate of VFadm-f closely paralleled that of VFadm, suggesting a comparable influence between the force penalty term and the admittance controller. The success rate of VFDSL-f was lower than VFDSL one, which meant that the force penalty term in the reward function did not improve the model performance of VFDSL. This was because we encouraged the soft body to contact the target during the exploration phase. However, soft bodies with different stiffnesses required different desired forces, and the fixed F_d in the force penalty term could not offer suitable force feedback. In particular, the force penalty term constrained the movement of the EEF, resulting in a slightly reduced Zf in models incorporating this term compared to those without it. VFDSL-f stood out as an exception mainly due to its higher frequency of failures than the VFDSL one, leading to more instances where the soft object remained at the target and underwent compression.

Then, to obtain a more comprehensive discussion, we analyzed the results of real and simulated. Since the inconsistent running time of real-world experiments and simulation experiments, we compared the success rate for insertion, the values of all models were lower in the real world. One of the contributing factors was the limited quantity of samples available for analysis. The second reason was that the real world model was derived from a simulated model transfer, so the lower success rate was expected. The final reason was the persistent Sim2Real gap, which could not be fully eliminated. Moreover, we analyzed the characteristics of VFDSL to discuss its scope of application, Its ability to adapt to soft bodies and explore complex paths makes it suitable for tasks like clamping soft brushes for cleaning items, laying wires, and more. However, the repeated touching action made it unsuitable for work that requires efficiency such as assembly lines. In summary, the VFDSL performed search and insertion tasks more smoothly, and the proposed model was applicable to tools with varying stiffnesses.

V. Conclusion and Future Work

This paper proposes a novel framework for insertion-type tasks that can generate dynamic compensation with the DSL method and adapt to soft bodies with varying deformation intensities and the complex hole's path. First, we propose a multimodal model to search the hole and insert the soft bodies. Then, using visual and force perception information, we train a policy with PPO algorithm for the insertion task, which embeds the DSL method. Finally, the DR is used for the soft bodies' parameters to reduce the difficulty of Sim2Real. We validate our approach by implementing it on a real robot system and achieve comparable performance results. The results show that our method outperforms the admittance controller for searching and insertion.

However, this work is a preliminary exploration of the DSL-based insertion task. There are limitations to this work. For instance, the hyper-parameters of DSL are manually specified. It would be interesting to use optimization methods to automatically adjust hyper-parameters or set the range and add the hyper-parameters to the action space; for hole identification, more tasks can not use the markers; the fingers of the gripper have no perceptual information, and tactile information can be added to the fingers to enrich the model.

References

[1] C. Jiao, X. Jiang, X. Li, and Y. Liu, "Vision based cable assembly in constrained environment," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 8–13, 2018.

[2] J. Xu, K. Liu, Y. Pei, C. Yang, Y. Cheng, and Z. Liu, "A noncontact control strategy for circular peg-in-hole assembly guided by the 6-dof robot based on hybrid vision," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–15, 2022.

[3] H. Park, J. Park, D.-H. Lee, J.-H. Park, and J.-H. Bae, "Compliant peg-in-hole assembly using partial spiral force trajectory with tilted peg posture," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4447–4454, 2020.

[4] Y.-L. Kim, B.-S. Kim, and J.-B. Song, "Hole detection algorithm for square peg-in-hole using force-based shape recognition," in *2012 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 1074–1079, 2012.

[5] Y. Kobari, T. Nammoto, J. Kinugawa, and K. Kosuge, "Vision based compliant motion control for part assembly," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 293–298, 2013.

[6] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Science Robotics*, vol. 6, no. 54, p. eabd8803, 2021.

[7] S. Huang, Y. Yamakawa, T. Senoo, and M. Ishikawa, "Robotic needle threading manipulation based on high-speed motion strategy using high-speed visual feedback," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4041–4046, 2015.

[8] R. Zanella, D. De Gregorio, S. Pirozzi, and G. Palli, "Dlo-in-hole for assembly tasks with tactile feedback and lstm networks," in *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 285–290, 2019.

[9] F. Stella, J. Hughes, D. Rus, and C. Della Santina, "Prescribing cartesian stiffness of soft robots by co-optimization of shape and segment-level stiffness," *Soft Robotics*, vol. 10, no. 4, pp. 701–712, 2023.

[10] M. Vecerík, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. M. O. Heess, T. Rothörl, T. Lampe, and M. A. Riedmiller, "Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards," *ArXiv*, vol. abs/1707.08817, 2017.

[11] J. Langaa and C. Sloth, "Expert initialized reinforcement learning with application to robotic assembly," in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, pp. 1405–1410, 2022.

[12] S. Kozlovsky, E. Newman, and M. Zacksenhouse, "Reinforcement learning of impedance policies for peg-in-hole tasks: Role of asymmetric matrices," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10898–10905, 2022.

[13] S. Li, X. Yuan, and J. Niu, "Robotic peg-in-hole assembly strategy research based on reinforcement learning algorithm," *Applied Sciences*, vol. 12, no. 21, 2022.

[14] C. C. Beltran-Hernandez, D. Petit, I. G. Ramirez-Alpizar, and K. Harada, "Variable compliance control for robotic peg-in-hole assembly: A deep-reinforcement-learning approach," *Applied Sciences*, vol. 10, no. 19, 2020.

[15] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, pp. 411–444, 2022.

[16] C. D. McKinnon and A. P. Schoellig, "Experience-based model selection to enable long-term, safe control for repetitive tasks under changing conditions," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2977–2984, 2018.

[17] T. Mannucci, E.-J. van Kampen, C. de Visser, and Q. Chu, "Safe exploration algorithms for reinforcement learning controllers," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 4, pp. 1069–1081, 2018.

[18] Y. Zheng, R. Pei, and C. Chen, "Strategies for automatic assembly of deformable objects," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pp. 2598–2603 vol.3, 1991.

[19] T. Jiang, H. Cui, X. Cheng, and W. Tian, "A measurement method for robot peg-in-hole prealignment based on combined two-level visual sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.

[20] C. C. Beltran-Hernandez, D. Petit, I. G. Ramirez-Alpizar, T. Nishi, S. Kikuchi, T. Matsubara, and K. Harada, "Learning force control for contact-rich manipulation tasks with rigid position-controlled robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5709–5716, 2020.

[21] A. Lämmle, P. Tenbrock, B. Bálint, F. Nägele, W. Kraus, J. Váncza, and M. F. Huber, "Simulation-based learning of the peg-in-hole process using robot-skills," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9340–9346, 2022.

[22] M. A. Lee, Y. Zhu, P. Zachares, M. Tan, K. P. Srinivasan, S. Savarese, F.-F. Li, A. Garg, and J. Bohg, "Making sense of vision and touch: Learning multimodal representations for contact-rich tasks," *IEEE Transactions on Robotics*, vol. 36, pp. 582–596, 2019.

[23] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 1889–1897, 2015.

[24] K. Tanaka, R. Yonetani, M. Hamaya, R. Lee, F. v. Drigalski, and Y. Ijiri, "Trans-am: Transfer learning by aggregating dynamics models for soft robotic assembly," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4627–4633, 2021.

[25] L. Leyendecker, M. Schmitz, H. A. Zhou, V. Samsonov, M. Rittstieg, and D. Lütticke, "Deep reinforcement learning for robotic control in high-dexterity assembly tasks - a reward curriculum approach," in *2021 Fifth IEEE International Conference on Robotic Computing (IRC)*, pp. 35–42, 2021.

[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv e-prints*, p. arXiv:1707.06347, 2017.

[27] Y. Ma, D. Xu, and F. Qin, "Efficient insertion control for precision assembly based on demonstration learning and reinforcement learning," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4492–4502, 2021.

[28] S. James, A. J. Davison, and E. Johns, "Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task," in *Proceedings of the 1st Annual Conference on Robot Learning*, vol. 78, pp. 334–343, 2017.

[29] J. Z. Zhang, Y. Zhang, P. Ma, E. Nava, T. Du, P. Arm, W. Matusik, and R. K. Katzschmann, "Sim2real for soft robotic fish via differentiable simulation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12598–12605, 2022.

[30] Y. Deng, C. Xia, X. Wang, and L. Chen, "Deep reinforcement learning based on local gnn for goal-conditioned deformable object rearranging," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1131–1138, 2022.

[31] P. Schegg, E. Ménager, E. Khairallah, D. Marchal, J. Dequidt, P. Preux, and C. Duriez, "Sofagym: An open platform for reinforcement learning based on soft robot simulations," *Soft Robotics*, vol. 10, no. 2, pp. 410–430, 2023.

[32] M. Krogius, A. Haggenmiller, and E. Olson, "Flexible layouts for fiducial tags," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1898–1903, 2019.