A Robust and Safe Strategy for Robotic Assembly

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

***Abstract*— In this work we present a reinforcement learning (RL) based approach for enabling a robot to safely perform assembly-type tasks. The proposed strategy involves both grasping and assembly, although our main focus is on the latter. Instead of a pure visual approach, we opt for a combination of force feedback and visual feedback to perceive the shape and direction of the holes. To ensure safe operation, a force-based dynamic safety lock (DSL) is introduced, which limits the pressing force of the robot and prevents emergency stops from being triggered due to excessive force output. Finally, we train and test the strategy with a simulator and build ablation experiments to illustrate the effectiveness of our method. The strategies are independently tested 500 times in the simulator, and we get an 88.57% success rate with a 4*mm* gap. These models are transferred to the real world and deployed on a real robot. We conducted independent tests and obtained a 79.63% success rate with a 4*mm* gap. Simulation environments: https://github.com/0707yiliu/peg-in-hole-with-RL**

I. Introduction

In some assembly-type tasks such as key insertion, humans complete the task through a combination of visual and contact force perception. However, it is challenging to endow robots with such capabilities, which require precise and rich graphical recognition algorithms and force perception-based algorithms.

Visual feedback provides overall information about the object geometry and its surroundings used for pre-capturing and insertion. A purely vision-based model can be deployed on the robot to complete the partial assembly task [1], but these models cannot make robots determine how much force is required.

Force sensing is a way for robots to determine physical parameters. Using it to provide partial feedback during collision or contact, the assembly process can be controlled accurately and safely [2]. Therefore, the robot can obtain many environmental details by referring to multiple information such as vision and force sensing.

The application of RL [3] provides a policy for robots to complete assembly-type tasks. The challenge of RL is that the complete state of the environment cannot be immediately observed without observing the geometry of the object through the camera, but is perceived through contact feedback. Thus, the insertion policy not only tries to align the mismatch between targets but also needs to adjust the insertion direction [4]. Furthermore, the exploration of RL causes the robot to collide with the environment, hence, the safety of robot interaction deserves consideration.

In this paper, as shown in Fig. 1, we propose a strategy that can utilize multiple types of sensors with different characteristics. The proposed strategy is constructed by RL and can be generalized across similar manipulation tasks (e.g., similar geometries, configurations, and object sizes). The approach is to learn the joint representation of force/torque (F/T) sensors, robot proprioceptive information, and vision sensors through a fully connected neural network to obtain the necessary action and set the DSL for the robot.

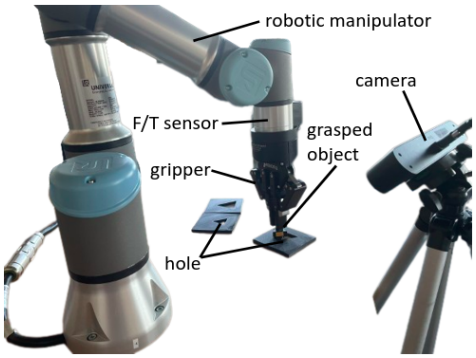

Fig. 1: The overview of the robotic platform.

We summarize the key contributions as follows.

1) The DSL is set for the robot's motion trajectory to ensure safety interaction during the insertion process.
2) Simplifying the vision function by using F/T sensors to determine the precise position and direction of the hole instead of the camera.
3) Demonstrating how to effectively use F/T sensors and visual feedback for hole searching, alignment and insertion on real robot.

II. Methodology

A. Assembly Task Setting in Simulation

As shown in Fig. 2. Firstly, the training environment includes a 6-DoF robotic arm with a gripper, a workspace with a table, holes, and objects. Then, since this paper mainly focuses on the insertion action rather than the grasping action, the initial position of the robot end-effector (EEF) is fixed. The initial robot joint configuration is calculated via inverse kinematics. The target (hole) location is randomized within a defined domain that the EEF can reach.

Finally, to encourage the robot to learn the policy effectively, we set the size of the object in the training environment to be constant and change the size of the hole, i.e., changed the gap between the hole and the object. Since the trajectory of the robot in this work is continuous, it is considered to use the standard PPO algorithm [5]. Since we mainly consider the policy of insertion action as described above, the policy of grasping action is ignored in this policy and the object is fixed to the EEF in a way that it has 1-DoF.

B. Observation, Action, and Reward

The RL state s_t consists of the robot state s_t^r, the assembly task state s_t^a. The robot state s_t^r contains the robotic EEF position $\mathbf{p}_t^{ee} = [x_t^{ee}, y_t^{ee}, z_t^{ee}]$, the EEF force and torque obtained with the F/T sensor $\mathbf{f}_t^{ee}$, angular value of the last joint q_t^6 at the end of the robot. The task state s_t^a contains the hole position $\mathbf{p}_t^h = [x_t^h, y_t^h, z_t^h]$.

The action a_t consists of a three-dimensional displacement increment of the robot EEF $[\Delta x, \Delta y, \Delta z]$ and a rotation perpendicular to the insertion direction $\Delta\theta z$. For the $[\Delta x, \Delta y, \Delta z]$,

† denotes the corresponding author.
[1] Authors are with IDLab-AIRO, elis, Ghent University – imec, Ghent, Belgium (email: yiyiliu.liu@ugent.be)

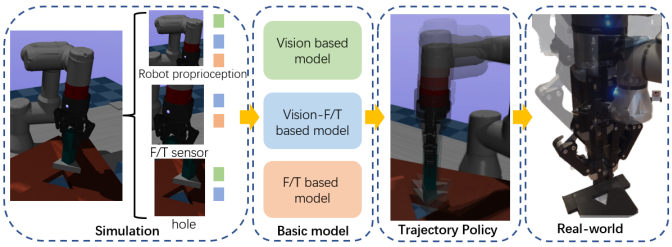

Fig. 2: Model training schematic. The simulation part contains the state of the robot and the environment that we need, and the motion trajectories from the simulation can be transferred to the real robot.

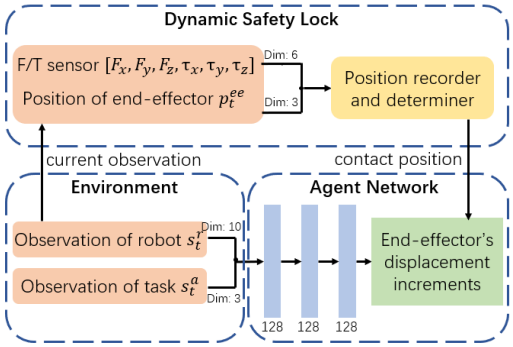

Fig. 3: Network structure of robot assembly models. The entire network is divided into three parts, the environment, the agent network, and the dynamic safety lock (II-C).

the movement of the end is considered to compose the action set in order to satisfy the relative position $\mathbf{r}_h^{ee}$ needed for the observation.

For the reward function r_t required for the RL model, we choose the Euclidean distance as the basic function, i.e., the distance between the hole and the object. The observation s_t does not include the position of the object, but fixes the object to the EEF (in II-A), therefore, the Euclidean distance between the hole and the EEF d_h^e is calculated.

$$\begin{aligned} r_t = &\sqrt{\alpha_1 * (x_t^{ee} - x_t^h)^2 + \alpha_2 (y_t^{ee} - y_t^h)^2 + \alpha_3 (z_t^{ee} - z_t^h)^2} \\ &+ \alpha_4 [d(\mathbf{p}_t^{ee}, \mathbf{p}_t^h) < \delta_1] \\ &+ \alpha_5 z_{\text{dist}}, \end{aligned} \tag{1}$$

where $[\alpha_1, \alpha_2, \alpha_3]$ is the important weight vector of the different direction, $d(\cdot)$ expresses the Euclidean distance between $\mathbf{p}_t^{ee}$ and $\mathbf{p}_t^h$, this part could be activated when the distance less than the threshold δ_1. z_{dist} represents the distance in the insertion direction. The cost is defined as:

$$z_{\text{dist}} = \begin{cases} z_t^h - z_t^{ee}, & d(\mathbf{p}_t^{ee}, \mathbf{p}_t^h) < \delta_2 \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where δ_2, similar to δ_1, is also the judgment threshold for distance $d(\cdot)$, but the value of δ_2 is greater than the δ_1 one, which means the reward can be activated when the object is inserted into the hole.

C. DSL for Robot

To ensure that the robot does not collide with the environment violently and thus has problems such as system crashes, as shown in Fig. 3, we suggest a DSL method in each control loop. The vector consisting of these two sets of signals is used as the input to the DSL. When the EEF touches around the hole, the change in the value of the F/T sensor becomes larger and the recorded position

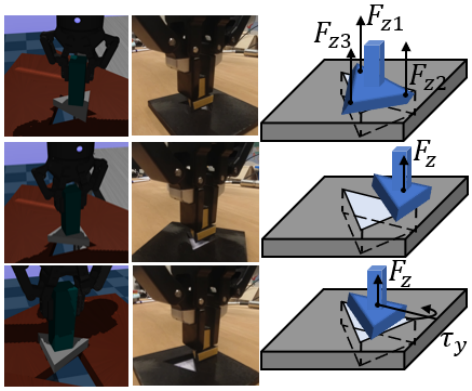

Fig. 4: Schematic diagram of the detection points. Each line is a detection behavior, respectively, simulation, real-world and geometry schematic. Each contact point represents a small contact zone, and the contact forces, where $F_z = F_{z1} + F_{z2} + F_{z3}$, τ_y represents the torque in the y-direction, are represented by the F/T sensor.

$\mathbf{R}_p = [x_{ti}^{ee}, y_{ti}^{ee}, z_{ti}^{ee}](i = 1, 2, 3, ...)$ at this moment is used as the limited contact position z_t^c for the next time of exploration. To speed up the exploration to find the initial first limit, we artificially add a tiny increment δz_t^{ee} in the direction of the insertion of the output increment, which does not affect the overall trajectory of the RL model exploration. Finally, to more concretely represent the function of the DSL, we list the pseudo-algorithm as shown in Algorithm 1.

Since exploring the edge of the hole or not has a relatively large effect on the F/T sensor as shown in Fig. 4, we set threshold $\delta\mathbf{f}$, which is activated when the edge of the hole is explored on the first exploration, to be used to broaden the limit z_t^c. The specific calculation is as follows.

$$\begin{aligned} \delta x_t^{ee} &= \beta_{11} \cdot (F_{x(t)} - F_{x(t-1)} + \tau_{x(t)} - \tau_{x(t-1)}) \\ \delta y_t^{ee} &= \beta_{12} \cdot (F_{y(t)} - F_{y(t-1)} + \tau_{y(t)} - \tau_{y(t-1)}) \\ \delta z_t^{ee} &= \beta_{13} \cdot (F_{z(t)} - F_{z(t-1)} + \tau_{z(t)} - \tau_{z(t-1)}), \end{aligned} \tag{3}$$

where $F_{\cdot(t)}$, $F_{\cdot(t-1)}$, $\tau_{\cdot(t)}$ and $\tau_{\cdot(t-1)}$ denote the last two values of the $\mathbf{R}_f$ record. $\boldsymbol{\beta}_1 = [\beta_{11}, \beta_{12}, \beta_{13}]$ represents the F/T variation gain vector. The tiny increments δx_t^{ee}, δy_t^{ee} and δz_t^{ee} obtained are used as leverage to raise the limit z_t^c.

$$z_t^c = z_t^{ee} + \delta x_t^{ee} + \delta y_t^{ee} + \delta z_t^{ee}, \tag{4}$$

where z_t^{ee} denotes the last value of the $\mathbf{R}_p$ record. On the other hand, if the threshold R is not activated, i.e., it is considered that the edge of the hole is not touched, but the strength of the last position change $\delta\mathbf{R}_p$ reflects the intensity of the downward exploration, so $\delta\mathbf{R}_p$ is used as a tiny increment to regulate the limit z_t^c.

$$\delta\mathbf{R}_p = \|\mathbf{R}_{p(t)} - \mathbf{R}_{p(t-1)}\|, \tag{5}$$

$$z_t^c = z_t^{ee} + \boldsymbol{\beta}_2 \cdot \delta\mathbf{R}_p, \tag{6}$$

where $\mathbf{R}_{p(t)}$ and $\mathbf{R}_{p(t-1)}$ denote the last value it record. $\boldsymbol{\beta}_2$ represents the gain vector of $\delta\mathbf{R}_p$.

III. Experiments and Results

A. Simulation Experiments

1) Experimental Setup Details: All training and testing in the simulation part are on the Intel(R) Core(TM) i7-1185G7 CPU. The hyper-parameters mentioned in this work were manually optimized. For the reward function (Equation (1)), $\alpha_1 = \alpha_2 = 2.30$, $\alpha_3 = 1.23$, $\alpha_4 = 2$, $\alpha_5 = 0.5$, $\delta_1 = 1e{-}04$, $\delta_2 = 0.01$ (Equation (2)). For the DSL, $\boldsymbol{\beta}_1 =$

Algorithm 1: DSL.

Input:

- F/T sensor: $[F_x, F_y, F_z, \tau_x, \tau_y, \tau_z]$
- Position of the EEF: $[x_t^{ee}, y_t^{ee}, z_t^{ee}]$

Output: Contact position z_t^c from the recorder.

// the observation can be provided for this algorithm.
Initialize p_t^h, p_t^{ee} randomly
Define tiny increment δz_t^{ee}
Define the six-dimensional force threshold $\delta \mathbf{f}$
Normalized the F/T sensor data
while not contact ($F_z < 0.5$) **do**
  Record F/T sensor's data $[F_x, F_y, F_z, \tau_x, \tau_y, \tau_z]$ and position $[x_t^{ee}, y_t^{ee}, z_t^{ee}]$ as $\mathbf{R}_f$, $\mathbf{R}_p$
  Add tiny increment δz_t^{ee} into z_t^{ee} in the direction of insertion to gradually explore downward.
if $\mathbf{R}_f(t) - \mathbf{R}_f(t-1) > \delta \mathbf{f}$ **then**
  Utilize changes in F/T sensor to obtain the tiny increments $\delta x_t^{ee}, \delta y_t^{ee}, \delta z_t^{ee}$ (Equation (3))
  Tiny increments act as gains on z_t^c (Equation (4))
else
  Calculate the intensity of the position $\delta \mathbf{R}_p$ change at the last moment (Equation (5))
  Tiny increments $\delta \mathbf{R}_p$ act as gains on z_t^c (Equation (6))

$[1e{-}03, 1e{-}03, 5e{-}04]$, $\beta_2 = [1e{-}07, 1e{-}07, 1e{-}03]$, $\delta \mathbf{f} = [0.15, 0.15, 0.45, 0.1, 0.1, 0.2]$. For the observation, Gaussian noise is added to the observations of the holes. The observation of the robot s_t^r has the noise of the simulator, so no additional noise is added to it. The total number of training steps is 2e06, and the maximum number of steps per episode is 110. For the task, we randomize 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 are set with checkpoints and estimated models are generated every 1e04 steps. The estimated models are set up to test the success rate of the trained models, which is judged by the bottom of the object being more than 2.5*mm* below the surface of the hole.

2) Vision and F/T Model: We set up three sets of experiments with different data inputs, which constitute three models with DSL, the vision-based model (VM), the vision-F/T-based model (VFTM), and the F/T-based model (FTM). For VM, the observation space contains $\mathbf{p}_t^{ee}$ and $\mathbf{p}_t^h$, the VFTM one contains all of the state in II-B, the last one contains $\mathbf{p}_t^{ee}$ and $\mathbf{f}_t^{ee}$.

We set a single fixed assembly gap with a size of 4.0*mm* and a fixed size of the grasped object. As shown in Fig. 5, firstly, for the VM, we set the $\beta_2 = 0$ to disable the DSL. Then, we can see that VFTM has the highest reward with the fixed gap and has a high success rate. The success rate performance of FTM is better than VM one. A reason is that we do not use a camera-based shape recognition algorithm that requires a large amount of data to identify the shape and orientation of the hole, but rather randomize the position and orientation of the hole, while FTM is able to learn the skill of alignment through the encoding of F/T sensor data. However, as this model has no visual support, FTM does not know where the hole is. Thus, FTM can only get the position of the hole by constantly exploring, which leads to sometimes when the entire episode is over did not explore the position of the hole so that the success rate is not remarkable.

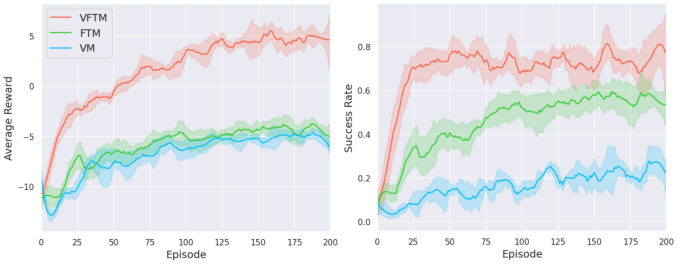

Fig. 5: Training performance of three assembly models (VFTM, FTM and VM) based on vision and F/T sensor with the 4*mm* gap.

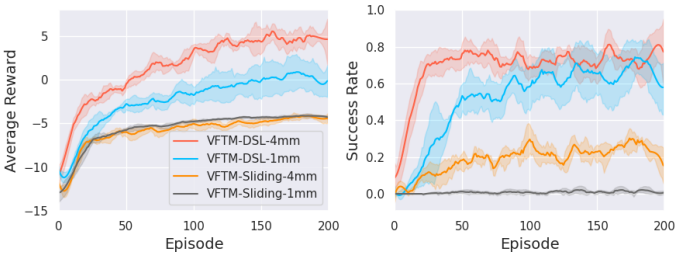

Fig. 6: Performance of training method (DSL vs. Sliding).

Finally, we take the best result of each model as the best model for testing and get the results as shown in Table I, where each model is tested 500 times independently to obtain the reward and success rate. In this subpart, we focus on the 4*mm* gap and we can see that the success rates of VFTM, FTM and VM are 88.57%, 56.42% and 25.59%, respectively, which are consistent with the results analyzed from Fig. 5 in the above. For the reward, FTM has the largest variance, which is reasonable because of its need for continuous exploration.

It can be proven that the model with rich data is more effective, the F/T sensor can replace the shape recognition function of the visual sensor to accomplish the task.

TABLE I: Model testing results in simulation. (r: reward, sr: success rate)

		4mm	1mm
VFTM-DSL	r	5.73±0.51	-0.32±0.44
	sr	88.57%	41.95%
VFTM-Sliding	r	-4.29±0.03	-4.29± 0.48
	sr	25.03%	0.64%
FTM-DSL	r	-4.56±0.69	-
	sr	56.42%	-
VM	r	-5.31±0.23	-
	sr	25.59%	-

3) DSL Experiment: This part, at last, establishes ablation experiments about the proposed DSL method. As mentioned in II-C, the DSL method differs from the traditional sliding one by having a repetitive pressing action. Therefore, this experiment only distinguishes the effects of different action styles (DSL vs. Sliding). From the above experimental results, it is obtained that VFTM is the most effective, so this experiment is conducted with VFTM at different sizes of gaps (4*mm* and 1*mm*).

As shown in Fig. 6, qualitatively, for the reward, the obtained score by the DSL method is higher than the sliding

TABLE II: Insertion experiment results in real world.

	tr		rtr	trm		cir	b-trm		b-rtr
proportion	26.3%	7.8%	24.7%	30.65%	9.33%	13.78%	20.50%	5.8%	5.5%
success rate	79.63%	18.64%	68.52%	83.67%	15.43%	20.00%	41.37%	13.95%	22.81%

method one for each size of the gap. For the success rate, The performance of both sliding methods is low, while the performance of the DSL method is remarkable. Quantitatively, as shown in Table I, the data corresponding to Fig. 6 in this subpart are VFTM with DSL method and VFTM with sliding method. The success rate of the sliding method with the 4*mm* gap is 25.03% but the DSL method has 88.57%, for the 1*mm* gap, the DSL method has 41.95% success rate and the sliding method cannot complete the task, which indicates that the sliding method does not match the hole well at the smaller gap. For the fluctuation of the data, i.e., the variance, the fluctuation of the sliding method is smaller compared to the DSL method, which indicates that the method repeatedly explores around the hole location after exploring it and does not find the insertion direction.

B. Real-World Robotics Experiments

The experiments are completed on the device UR3e equipped with Robotiq2F. The hole position with the marker is captured by a ZED2i camera. Since the effectiveness of the DSL method has been demonstrated in the simulation, we no longer compare the DSL method and the sliding method in the real world. According to the characteristics of the F/T sensor, we designed an exploratory experiment to generalize the model based on the triangle training in Fig. 4 to other shapes or other sizes as shown in Fig.7.

Based on Fig.7, we also changed the size of the hole and used the area proportion of the object to the hole instead of the gap size (4*mm*) to describe the gap. The smaller the proportion, the smaller the gap. As shown in Fig. 8, we conduct experiments with the objects shown in Fig. 7 and holes of different sizes. Note that all experiments are performed with the same model (VFTM-DSL), the purpose is to test the generalization effect of the model by using the perceptual ability of the F/T sensor.

We tested different situations about 50 times, each time artificially changing the orientation and position of the hole randomly, and got the results shown in Table II. It can be seen that the success rate of *tr* (original model) is 79.63%, which is similar to the test results in simulation, and it has a success rate of 28.64% when the gap is 1*mm* (7.8% proportion). Objects of similar shape (*rtr*, *trm*) can have a higher success rate under similar size conditions (68.52%, 83.67% and 25.43%). In particular, the proportion of *trm* is larger than the *tr* one, and its success rate is higher than the original model one. For the object (*cir*) with a large difference in shape, the success rate is reduced to 20%. This is because the force sensor failed to find a suitable pose to insert during the pressing process due to the mismatch of the object shape. For objects of different sizes (*b-trm*, *b-rtr*), the success rates (31.37%, 13.95% and 22.81%) are reduced compared to the original model. This is because when the object is larger, the change of the F/T sensor's value is reduced during the pressing process, thereby reducing the probability of exploring the hole. However, there is still a success rate (31.37%) when the gap proportion is large (20.50%).

In short, we use the F/T sensor to replace some of the functions of the camera and use the changing characteristics of the sensor's feedback during the exploration process, so that the proposed model can be applied to objects of similar shape and size.

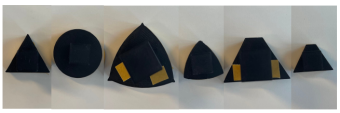

Fig. 7: Objects of different shapes and sizes, from left to right are triangle (tr), circle (cir), big size Reuleaux triangle (b-rtr), Reuleaux triangle (rtr) and triangle with missing corner (trm) and the big size one (b-trm).

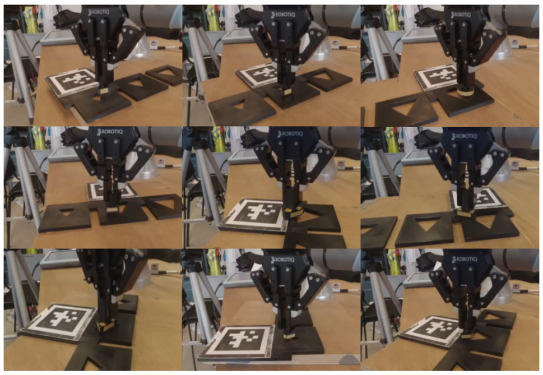

Fig. 8: Insertion experiment, from top to bottom, from left to right are *tr* with 7.8% and 26.3% proportion, *cir* with 13.78% proportion, *rtr* with 24.7% proportion, *b-rtr* with 5.5% proportion, *b-trm* with 20.50% and 5.8% proportion, *trm* with 9.33% and 30.65% proportion.

IV. Conclusion and Future Works

In this work, a reinforcement learning based approach with the dynamic security lock (DSL) is proposed. This can improve the robustness and safety of the robot in the real world, and improve the ability of the robot to make compliant contact with the environment. Moreover, this work replaces the visual shape recognition ability with the F/T sensor. For future work, we can be extended to build a more complete model to work in situations where the data of the camera is corrupted, for example.

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] B. Ti, Y. Gao, M. Shi, and J. Zhao, "Generalization of orientation trajectories and force–torque profiles for learning human assembly skill," *Robotics and Computer-Integrated Manufacturing*, vol. 76, p. 102325, 2022.

[3] 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.

[4] 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.

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