# Tactile-RL for Insertion: Generalization to Objects of Unknown Geometry

Dong, Siyuan; Jha, Devesh K.; Romeres, Diego; Kim, Sangwoon; Nikovski, Daniel N.; Rodriguez, Alberto

TR2021-052    June 04, 2021

**Abstract**

Object insertion is a classic contact-rich manipulation task. The task remains challenging, especially when considering general objects of unknown geometry, which significantly limits the ability to understand the contact configuration between the object and the environment. We study the problem of aligning the object and environment with a tactile-based feedback insertion policy. The insertion process is modeled as an episodic policy that iterates between insertion attempts followed by pose corrections. We explore different mechanisms to learn such a policy based on Reinforcement Learning. The key contribution of this paper is to demonstrate that it is possible to learn a tactile insertion policy that generalizes across different object geometries, and an ablation study of the key design choices for the learning agent: 1) the type of learning scheme: supervised vs. reinforcement learning; 2) the type of learning schedule: unguided vs. curriculum learning ; 3) the type of sensing modality: force/torque vs. tactile; and 4) the type of tactile representation: tactile RGB vs. tactile flow. We show that the optimal configuration of the learning agent (RL + curriculum + tactile flow) exposed to 4 training objects yields an closed-loop insertion policy that inserts 4 novel objects with over 85.0% success rate and within 3   4 consecutive attempts. Comparisons between F/T and tactile sensing, shows that while an F/T-based policy learns more efficiently, a tactile-based policy provides better generalization.

*IEEE International Conference on Robotics and Automation (ICRA)*

# Tactile-RL for Insertion:
# Generalization to Objects of Unknown Geometry

Siyuan Dong[1], Devesh K. Jha[2], Diego Romeres[2], Sangwoon Kim[1], Daniel Nikovski[2] and Alberto Rodriguez[1]

[1]Massachusetts Institute of Technology, [2]Mitsubishi Electric Research Laboratories

[1]<sydong,sangwoon,albertor>@mit.edu, [2]<jha,romeres,nikovski>@merl.com

*Abstract*— Object insertion is a classic contact-rich manipulation task. The task remains challenging, especially when considering general objects of unknown geometry, which significantly limits the ability to understand the contact configuration between the object and the environment. We study the problem of aligning the object and environment with a tactile-based feedback insertion policy. The insertion process is modeled as an episodic policy that iterates between insertion attempts followed by pose corrections. We explore different mechanisms to learn such a policy based on Reinforcement Learning. The key contribution of this paper is to demonstrate that it is possible to learn a tactile insertion policy that generalizes across different object geometries, and an ablation study of the key design choices for the learning agent: 1) the type of learning scheme: supervised vs. reinforcement learning; 2) the type of learning schedule: unguided vs. curriculum learning ; 3) the type of sensing modality: force/torque vs. tactile; and 4) the type of tactile representation: tactile RGB vs. tactile flow. We show that the optimal configuration of the learning agent (RL + curriculum + tactile flow) exposed to 4 training objects yields an closed-loop insertion policy that inserts 4 novel objects with over 85.0% success rate and within $3 \sim 4$ consecutive attempts. Comparisons between F/T and tactile sensing, shows that while an F/T-based policy learns more efficiently, a tactile-based policy provides better generalization. See supplementary video and results at **https://sites.google.com/view/tactileinsertion**.

## I. INTRODUCTION

Localizing contacts between objects and their environment is central to many contact-rich robotic manipulation tasks [1]. Consider for example the classical tasks of inserting a peg in a hole or packing an object in a box. A mismatch between the real and modeled object's shape or between the real and target object's pose, generates unexpected contacts between the object and the hole, potentially leading to complex contact interlocking configurations [2].

The location of contacts during insertion depend on the geometrical attributes of the object and hole. Techniques to explicitly recover the location of those contacts, usually require knowledge of object geometries and/or are only applicable to limited types of contact configurations [3], [4], [5], [2]. The estimation of those contacts without knowledge of object geometry is also possible in some cases by aggregating tactile information over time [6].

In this paper, alternatively, we are interested in feedback-based mechanisms to interactively correct the miss-alignment
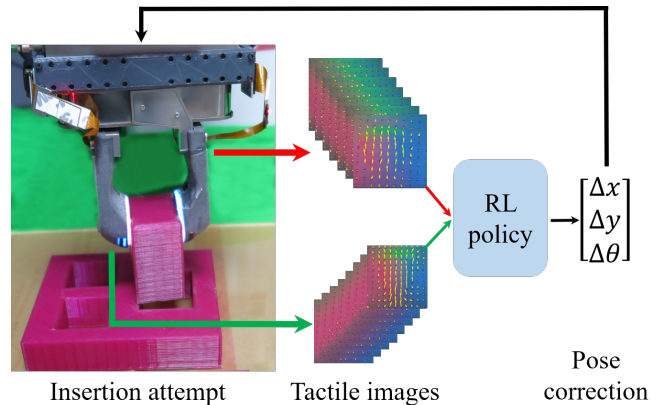
Fig. 1. Insertion task with the proposed Tactile RL policy.

between object and hole without making use of the geometry of the shapes that generated it and without making explicit inference of the contact location. Tactile sensors co-located at the gripper fingers (see Figure 1) are better positioned to capture contact events than vision sensors, which suffer from occlusions and limited accuracy. Force/Torque (F/T) sensors can also be co-located at fingers or wrists. In this work, we investigate the use of F/T and high-resolution tactile sensors to guide peg-in-hole insertion policies, and to generalize to different objects while accounting for variations in contact formations, in grasps, and in environments, i.e., a general *tactile insertion agent*.

The key challenge to a general insertion policy is that, in the absence of geometric models, the contact state is not fully observable from a single time instant, neither for force or tactile sensing. The insertion agent, then, needs to consider policies beyond greedily trying to estimate and correct the miss-alignment between part and hole, as was shown in previous schemes based on self-supervised learning, which showed limited generalization to objects and contact configurations [7], [2] In recent work [7], we demonstrated that high-resolution tactile sensors such as GelSlim [8] allow feedback correction for novel objects. However, the simplified insertion environment (a slot with constrain in one axis) only covers a very small set of possible contact formations in a object insertion problem.

In this paper, we explore the more complex and general problem of *inserting multiple objects into multiple environments with a single policy*, and we ablate several important design choices for an RL tactile-based insertion agent. In

particular we evaluate:

- **Supervised vs. Reinforced** We study the importance of the sequential nature of the learning process in rewarding successful insertions, and learn an episodic RL agent that iterates between insertion attempts and estimating alignment corrections (see Figure 1). While the supervised agent suffers in the more constrained insertion environments, the RL agent shows significantly better performance. The RL formulation, with the ability to consider delayed rewards, opens the possibility to search through non-greedy policies.
- **Curriculum training.** We show that a modest guidance of the learned policy to progressively tackle more complex environments (wall → corner → U → hole) is effective at reducing the number of necessary learning data.
- **Tactile representation.** We show that learning an insertion agent from tactile flow (represented with marker array) yields much better generalization to variations in object and grasp geometry than directly learning from raw tactile RGB images. Tactile flow is effective at preventing overfitting to object surface texture.

Since the dynamics of insertion and the dynamics of tactile sensing are both guided by complex and difficult to simulate contact mechanics, the use of real experiments and the need for data-efficiency are key drivers of this work. We also compare our tactile-RL agent with a policy trained using F/T sensors and show that the tactile-RL agent provides better generalization to objects of different geometry. These experiments also demonstrate the limitations of F/T measurements that can be alleviated using tactile sensors. All the experiments in this paper are done in a real system, with real data, and in real time. The insertion RL agent is learned in 500 episodes (8 hours of robot time). For this purpose, we developed an experimental setup (Fig. 3) that allows to run tactile-based automated insertion experiments under controlled variations on the initial grasp on different objects, and with an automatic resetting mechanism based on object-specific alignment features.

## II. RELATED WORK

Peg insertion tasks have been studied for a long time, due to their importance in manufacturing. Early work on peg-in-hole insertion algorithms were based on developing passive compliance devices to react to the experienced forces and change the peg's motions, [9], [10]. These devices, as well as similarly operating linear impedance controllers, can successfully complete insertions with chamfered pegs and holes, as long as the initial contact point is on the chamfer. Other methods use model-based approaches to directly estimate the pose of the peg relative to the hole with force feedback [11], [12]. However, these require knowledge of object models and/or are constrained to a single object type. Furthermore, there is a fundamental ambiguity in the mapping from alignment errors to sensed forces — in many contact situations, it is many-to-one, so it cannot be inverted directly [13].

An early RL approach to the actual insertion phase was proposed in [14], similarly based on an episodic RL setting and on minimizing the positional error of the peg with respect to the hole from observations of position and forces from an F/T sensor. Since then, several RL approaches have been proposed, mainly based on sensing and acting on the positional state and force-torques of the robotic manipulators. In [15], time-varying linear-Gaussian controllers combined together via a guided policy search component are presented. The authors obtained good generalization and data efficiency performance on a variety of insertion tasks. [16] studies the insertion of a rigid peg into a deformable hole, where the position of the hole relative to the robot is known, possibly with minor uncertainty. This paper combines robot-state information and filtered force/torque readings as the input of the controller. Vision-based RL approaches have also been studied. For example, [17] proposes using residual RL, to correct of a nominal policy produced by state of the art off-policy algorithms to perform connector insertions. Finally, we also find multi-modal sensing approaches. In [18] authors study the importance of each modality (vision, F/T and positional sensing) for peg insertion tasks in simulation which then transfer the policy into the real system. The authors of [19] proposed empirical studies on the effects of the three different modalities for insertion tasks in a real robotic set-up using behavioral cloning.

Most RL approaches above do not use force/torque or tactile signals to extract geometric information of contacts, but rather for detecting in/out of contact. In [7] the authors demonstrate the potential use of vision-based tactile sensors by explicitly estimating the contact error with supervised learning, and driving the insertion with a proportional controller. In this work, we directly train an RL policy to guide the insertion with tactile feedback from 2 vision-based tactile sensors in a more challenging insertion scenario with different objects and environments.

## III. METHODOLOGY

In this section we introduce the method by addressing the following 4 questions: 1) Why is tactile sensing essential for capturing contacts with the environment? 2) Why is RL a good fit for the insertion task? 3) Why is curriculum learning needed in the RL training? 4) How different representations of the tactile signal affect the learning agent?

### A. Tactile Sensing for Contact Localization

Humans routinely do blind insertions by feeling the interaction between object and environment through the grasp on the object. Contact with the environment generates small forces and small object displacements which are captured by force and tactile sensors in the fingers and hands. Similarly, we use two high-resolution tactile sensors to capture the subtle rotation signals of the object during contact in the insertion task. In [7], we have shown that it is possible in some configurations to recover explicit external contacts from those subtle rotation signals. These sensors capture a

depth image during contact as well as the in-plane displacement of an array of markers on the sensor surface.

Figure 2 illustrates the signals generated on the sensors under different contact configurations. In the example, the tactile images are captured during the contact phase between a cylindrical object and an edge in the environment, while the gripper moves down vertically. To visualize the flow of the tactile imprint during that phase, we use yellow arrows to draw the relative displacement of the marker field during that contact phase for Sensor1 (black) and Sensor2 (silver).

The key observation is that some different contact locations generate different patterns of tactile flow. In the first two configurations, the object contacts an edge that is perpendicular to the surface of the tactile sensors. In this case the object rotates in the plane of the sensors. The captured patterns of rotation contains information about the external contact point. Similarly, in the last two configurations, the object contacts an edge in the environment that is parallel to the surface of the tactile sensors. In this case the object rotates out from the plane of the sensors. These capture the vertical displacements of the contacts. For simple contact configurations as in Fig. 2, and if the geometry of the object and environment are known, it is not difficult to directly describe how the tactile flow should look like. In general, however, the relationship becomes more involved for complex contact formations. In any case, the tactile image sequence during the contact period from two high-resolution tactile sensors contains key information to the location of external contacts.

### B. Deep RL Controller

In previous work Dong and Rodriguez [7] used a supervised learning (SL) model to directly estimate the misalignment error between object and hole from GelSlim [8] tactile feedback in the form of image sequences. While this approach might work well in simple scenarios, due to the partial observity, the SL model may fall apart and make the controller diverge. We will show in Sec. IV that this approach does not work well in more complex environments such as a rectangular hole.

A more general solution is to model the insertion problem as a sequential decision problem and use a reinforcement learning (RL) algorithm to find the optimal policy. The input to the RL framework is the tactile image sequence during contact. The action is the robot motion in operational space. The reward is related not only to the pose error corrected in each step, but the success or failure of the insertion. The optimal RL policy tries to achieve the maximum reward by inserting the object inside the hole within a sequence of actions, instead of greedily correcting the pose error in each step. In this way, the policy can potentially perform better under some partially observable states, where it is not possible to estimate the full pose error with the tactile signals. This is the key advantage of the RL policy over an SL policy detailed in Section IV.

### C. Curriculum Learning

Due to the challenges of simulating the deformation of the tactile sensor with accuracy during contact, we train the insertion policy on a real robot. In order to improve the data efficiency of RL and make the problem feasible to solve in a reasonable training time, we explore the use of curriculum learning.

We design the learning curriculum by scaling the complexity of the insertion environment. In Fig. 3, we introduce 4 insertion environments shown in red: line wall → corner wall → U wall → hole. Note that each environment adds one more constraint to the previous environment, and forces the insertion policy to learn to handle that new constraint. Intuitively, the curriculum is asking the RL policy to incrementally learn to: 1) distinguish the contact direction in one axis in the line wall environment; 2) reason about the contact directions in 2 axis in the corner environment; 3) control the step size to avoid overshoot in one axis and rotation in the U environment; and 4) control step size in both axis and rotation in the hole environment.

In order to generalize the RL policy to different objects, we train the policy with 4 objects with different shapes (cylinder, hexagonal cylinder, elliptical cylinder, and cuboid) shown in Fig. 3. We also use curriculum learning by gradually increasing the number of objects to insert in the line environment. Generally, the cylinder, hexagonal cylinder, and elliptical cylinder are easier to insert, since they are less sensitive to rotational error. The cuboid is the most difficult to insert because a small misalignment in yaw will block the insertion.

### D. Tactile representation

The choice of the tactile signal representation affects the generalization of the policy to new objects, since the combination of a high-dimensional space, and a small number of training objects can easily lead to overfitting. High-resolution tactile images encode rich information of the contact surface. Some information content such as texture might not be relevant to the insertion task, while some other information might be more closely related to the location of external contacts, such as force distribution and tactile flow. Here we study the effect of data representation of the tactile signal on the RL policy using 2 different types of data input: 1) the raw RGB image, and 2) the marker flow image.

## IV. EXPERIMENTAL DETAILS

### A. Experimental Setup

Figure 3 shows the experimental setup, including a 6-DoF Mitsubishi Electric RV-4FL robot arm, a WSG-32 parallel-jaw gripper, two GelSlim sensors [8], one Mitsubishi F/T sensor 1F-FS001-W200 (only used to train the F/T RL policy training), four training objects and their resetting fixtures, and four environments for insertion, including line walls, corner walls, U walls and holes. Note that the fixtures on the sides of the experimental setup are used for automatic object pose resetting for the training objects. Note also that the clearance of the U wall and the hole environments is 3mm, while the average width of the objects is 35mm.

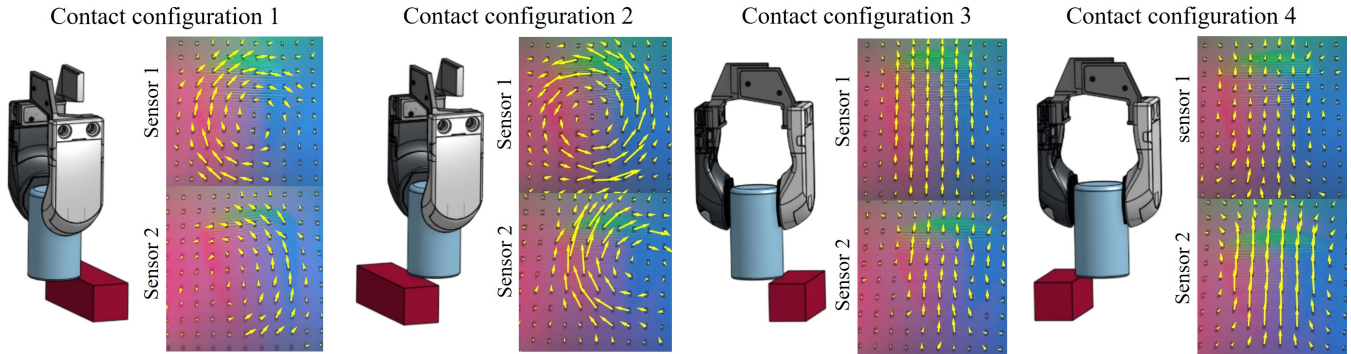Contact configuration 1     Contact configuration 2     Contact configuration 3     Contact configuration 4

Fig. 2. Tactile signals for four contact configurations. In the first 2 cases, the object contacts an edge in the environment that is perpendicular to the plane of the sensors, and consequently the object rotates in the plane of the sensors. Note that the rotation directions are opposite in cases 1 and 2. In the last 2 cases, the object contacts an edge parallel to the plane of the sensor, and consequently rotates out from the sensor plane. Note that the relative vertical displacements in the two sensors is different for cases 3 and 4.

### B. Insertion Experiment

Similar to [7], we assume that during actual operation, a noisy position of the hole would be estimated by a vision system, and the objective of the tactile-RL policy is to correct for the remaining relative pose error. To emulate this scenario, we introduce random translation error ranging from $-6 \sim 6$ mm in x and y axis (17% of the object's width) and rotation error in yaw angle ($\theta$) ranging from $-10° \sim 10°$. The resetting mechanism for objects, allows to pick them with controlled noise, different grasping forces and different heights. This is helpful to avoid overfitting to the height of the object or certain local tactile image features.

**Experimental process** For every episode, the robot randomly chooses and grasps an object and moves it to the top of the insertion environment with additional translational and rotational error. Subsequently, the robot moves down vertically to attempt an insertion, while monitoring collisions with slip detection [20]. If the slip detector is not triggered while moving down, that indicates a successful insertion. Otherwise, the insertion is blocked by the environment. We capture the tactile image sequences (or the F/T data when using the F/T policy) from the two tactile sensors during the contact and feed it to the controller. The robot moves the object to the next position according to the output of the controller, and starts another trial until either the object has been inserted or the insertion has failed. The robot resets the pose of the object by returning the object to its fixture after every 10 trials during the training process.

### C. Deep RL Policy

We choose an off-policy RL algorithm, the Twin Delayed DDPG (TD3) [21], to train the tactile-RL policy for better sample-efficiency. The underlying RL framework is represented by a tuple of $(S, A, p, R, \gamma)$. Here we use 12 images (downsampled from 30 images within 0.5 seconds of contact to avoid the gradient vanishing problem in the RNN model, with $640 \times 480$ pixel resolution) captured by each tactile sensor (24 in total) during the contact period as the state $s_t \in S$. The terminal states are either the successful insertion state, or the state that has error larger than a threshold in any

axis, or the state when the attempted number of insertions is over a limit. We set the max error $E_{max}$ in x and y axis to be 12mm, and 15° in $\theta$. The maximum number of attempts is 15. The action $a_t \in A$ is the continuous robot displacement in $(\Delta x, \Delta y, \Delta \theta)$ in the current gripper frame. The reward function $R$, shown in (1), is calculated with the difference between the current contact error ($e_t$) and the previous contact error ($e_{t-1}$). A constant penalty term is given for each insertion attempt ($P$), and additional reward ($R_s$) is given to a successful insertion ($\alpha = 1$, otherwise $\alpha = 0$).

$$R_t = e_{t-1} - e_t - P + \alpha R_s \tag{1}$$

We use a discount factor $\gamma - 0.99$ for computing the Q function. As mentioned in Sec. III, we use curriculum learning for RL training, and we stop training when the mean of the rewards crosses a threshold and the standard deviation is lower than a threshold in the last 30 episodes.

**Model architecture** TD3 is an actor-critic method [21]. We use a CNN+RNN+MLP model for the actor network with the 2 tactile image sequences used as input shown on the right-hand side of Fig. 3. Since the true state (the contact errors) of the task is known and thanks to precise initial grasps and careful resetting mechanisms during training, we directly use the current contact error in $(x, y, \theta)$ axis and the action $a_t$ from the actor model as the input to the critic model, instead of the tactile images. For the critic model, we use a small MLP model to estimate the $Q-$function. This can potentially speed up the convergence of the RL policy.

**Training details** To further accelerate training of the RL policy, we bootstrap the actor network with a supervised learning policy trained with 300 data points collected under a random policy in the first environment. In the first 50 episodes of RL training, we freeze the actor network to avoid updates by the untrained critic network. This helps to accumulate positive examples into the replay buffer.

### D. Supervised Learning Policy

To compare with previous work [7], we train a supervised learning (SL) policy with the data (tactile image sequence)
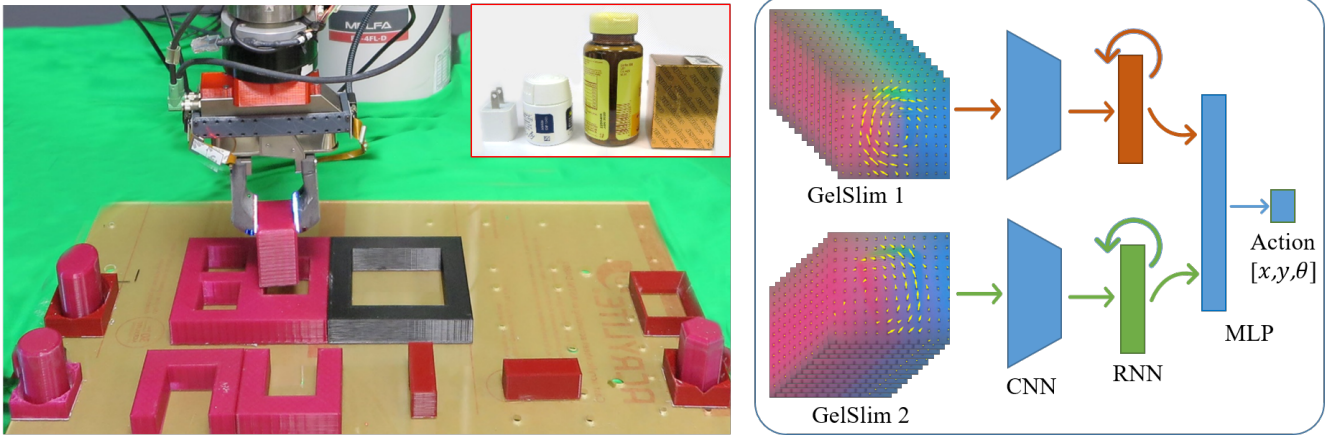
Fig. 3. Left: the experimental setup including a Mitsubishi Electric RV-4FL model robot arm, a WSG-32 parallel-jaw gripper, 2 revised GelSlim sensors, 4 sets of insertion environments, 4 training objects (cylinder and elliptical cylinder on the left, hexagonal cylinder on the right, cuboid held), and 4 novel everyday objects (phone charger, small bottle, big bottle, and box). Right: The Actor Neural Network model in TD3. The CNN includes 4 convolutional layers with 5, 3, 3, 3 kernel size and extracts $1 \times 512$ features. The RNN are 2 layers of LSTM with 512 memory units and output $.1 \times 512$ features. The MLP includes 2 fully connected layers with 512 and 256 hidden units. The marker flow is for visualization, not included in the training.

and labels (object-hole misalignment) saved during the RL policy training as a baseline. The SL policy takes the marker flow image sequence as the input and outputs the estimated contact error in $x, y$ and $\theta$ axis. For a fair comparison, we use the same architecture for the SL policy as the RL actor model. The SL policy is trained to estimate the contact error with an MSE loss. During execution, we use the action which is the opposite of the output of the SL policy to minimize the contact error iteratively. The maximum number of attempts is also 15.

### E. RL policy with F/T sensor

We train a RL policy with F/T sensing feedback as another baseline. We collect a stream of F/T signals (32 samples for each of the 6 dimensions of F/T, downsampled from 120) during each contact (0.5 s) as the input to the RL policy. The downsampling is used to avoid the gradient vanishing problem for the RNN model, but still keep the original force profile. Since the torque about the vertical axis $T_z$ only changes minimally during the contact, we remove it from the input to avoid unnecessary noise. We keep the general RL framework and only modify the neural network architecture of the actor network to a smaller RNN + MLP model to adapt to the data format of the F/T sensor. Since the model size is much smaller, we can train the policy directly in the hole environment with a SL policy initial bootstrap until the policy converges.

## V. RESULTS

We evaluate the performance of the polices by conducting 250 insertion experiments under different initial pose errors, with each of the 4 training objects and each of the 4 novel objects shown in Fig 3. The initial pose errors are uniformly sampled in the error space, where the errors in $x$ and $y$ axis range from -5mm to 5mm and the error in $\theta$ is from $-10°$ to $10°$. The performance metric includes the success rate and the average number of attempts, which is averaged only over the successful cases.

### A. RL Policy with Curriculum and Tactile Flow (RL* policy)

**Convergence** of the RL* policy is surprisingly fast. Due to the warm start of the actor network, the policy can already insert the cylinder most of the time in the line-wall environment at the start. Since some object-specific features (texture and shape) of individual objects are removed in the marker flow, the policy trained with 1 object can quickly adapt to the other three objects. Because the policy from the first environment can already perform well in the corner-wall environment, we stop training after 25 episodes (50 data points, which means ∼2 attempts each episode). Since the U-shaped environment is much harder compared to the first 2 environments, the policy needs around 150 episodes (500 data points) to adapt. The policy converges after 300 episodes (2000 data points) in the hole environment, even with the warm start from the previous environment. From our observation, the policy can quickly learn how to insert the first three objects, but spends most of the time to learn how to correct the rotation error for the cuboid object. The whole training process uses around 8 hours, 3000 data in total.

The **performance** of the RL* policy in the test experiments is illustrated in Table I. For the 4 training objects, the RL* policy achieves 90% success rate with the cuboid and over 97% success rate for the remaining 3 objects. The cuboid object needs around $5 \sim 6$ attempts to be inserted, whereas the other three objects only need $2 \sim 3$ attempts. This result agrees with the respective difficulty of the tasks. As we described earlier, whereas the policy needs to correct the rotation error of cuboid to complete insertion, the other three objects are not sensitive to rotational error. Misalignment in rotation is hard to observe, especially when accompanied by errors in $x$ and $y$ axis. This is because the tactile signal is generated by a complex combination of object's rotation into and rotation parallel to the sensor surface. The policy can also generalize well to 4 novel objects. It achieves over 96% success rate with the small bottle and the phone charger. Both

TABLE I

| | | Cylinder | Hexagonal cylinder | Elliptical cylinder | Cuboid | Big bottle | Small bottle | Phone charger | Paper box |
|---|---|---|---|---|---|---|---|---|---|
| RL* | Success | 97.1% | **97.1 %** | 98.0% | **89.6%** | 80.8% | 96.7% | **96.8%** | **70.1%** |
| | Attempt | 2.96 | 3.83 | **2.34** | **5.42** | 3.53 | **2.10** | **2.55** | **7.91** |
| SL | Success | 84.7% | 70.0% | 93.7% | 15.2% | 27.0% | 76.3% | 42.46% | 18.3% |
| | Attempt | 3.04 | 3.34 | 2.60 | 3.83 | 4.23 | 2.02 | 5.10 | 3.58 |
| RL w/o Curr | Success | 99.1% | 89.5% | 98.0% | 57.1% | 30.3% | 36.23% | 71.2% | 9.01% |
| | Attempt | 2.97 | 5.41 | 4.15 | 11.3 | 8.79 | 11.8 | 6.69 | 7.32 |
| RL RGB | Success | 98.8% | 93.1% | **99.15%** | 88.0% | 37.8% | 95.7% | 74.1% | 13.4% |
| | Attempt | 2.26 | **2.65** | 2.42 | **5.35** | 5.49 | 3.16 | 4.92 | 8.35 |
| RL F/T | Success | **100.0%** | 89.3% | **99.15%** | 61.2% | **99.5%** | 95.2% | 53.5% | 54.1% |
| | Attempt | **1.82** | 3.26 | 2.52 | 5.35 | **2.48** | 3.82 | 6.53 | 8.35 |
| RL SRL | Success | **100.0%** | 89.3% | 91.8% | **98.0%** | **99.5%** | 95.2% | 53.5% | 54.1% |
| | Attempt | **1.82** | 3.26 | 2.68 | **3.56** | **2.48** | 3.82 | 6.53 | 8.35 |

of them are either round or have round corners, which makes the insertion easier. The success rate of the big bottle drops to 80%, because the chamfer on the bottom edge changes the way the object rotates. The box can be inserted in over 70% of the tests, which has the lowest success rate, but is also the most difficult task. Compared to the training object cuboid, the success rate is lower but the RL* policy still achieves good performance. In terms of the number of attempts, the results for the novel objects are comparable to those for the training objects.

Most of the **failure** cases happen when both the translational (in $x$ and/or $y$) and rotational components (in $\theta$) of the initial pose error are large. Since the rotation of the object is too little to be captured by the tactile sensors in these cases, the RL policy can output an action in the opposite direction, resulting in a even bigger error, until it finally diverges.

*B. Baseline Comparison*

**SL policy** According to Table I, the SL policy performs a bit worse in the first 3 training objects, compared to the RL* policy, but can barely insert the cuboid during the test. In the test cases with 4 novel objects, only the small bottle is inserted with over 70% success rate, and the others have significantly lower success rate. Considering the number of attempts, the RL* policy is also more efficient. We find that the cuboid objects are particularly challenging for the SL policy, due to the fact that the tactile signals are similar in many contact configurations, and it's very difficult for the SL policy to make the right move. In addition, considering the SL policy is trained with MSE loss, a safe choice to the controller is to output 0 when the signal is not distinctive, which makes the SL policy get stuck. This comparison demonstrates that the RL policy learns a better strategy to insert objects even with some partially observable states.

**RL policy without curriculum** To evaluate the improvement of the curriculum training on the convergence speed of the RL policy, we train another RL policy directly in the hole environment with all 4 training objects. To keep it consistent with RL* policy, we also bootstrap the RL policy with the same supervised learning policy, and use maker flow as the policy input. We stop the training when it reaches

the same amount of data used for RL* policy training. The performance of this policy is shown in the 4th row of Table I. The policy achieves a similar success rate for the first three objects, but uses nearly twice the number of attempts compared to RL*. The policy fails in almost all of the cuboid insertions with large rotational error, and only gets 57% success rate, using 11 attempts on an average. Based on the comparison, we can interpret that 1) localizing the contact direction in x and y axis is easy to learn, 2) optimizing the number of attempts, relating to precisely localizing the amplitude of the contact error and a high-level insertion strategy to quickly gather information, is harder, 3) estimating the rotational error is also difficult and data-intensive. This RL policy does not generalize to the new objects well, and only achieves 30% average success rate.

**RL policy with RGB input** The raw RGB tactile image contains task-relevant as well as detailed object-specific features. It is much easier for the RL policy using RGB images to overfit to the training objects. This hypothesis agrees well with the result shown in Table I, 5th row. The performance of the policy with training objects is as good as that of the RL* policy. However, the success rate drops dramatically in the test of novel objects, especially with the big bottle and paper box. The policy keeps outputting the same action during the testing of these two objects, which reveals that the policy is likely overfitted. Using a proper tactile representation is important for generalization. Marker flow may not be the best representation, and we leave the problem of learning a good representation for future work.

**RL policy with F/T sensing** achieves almost 100% success rate with cylindrical objects with round edges in both the training (cylinder and elliptical cylinder) and novel objects (2 bottles), shown in Table I, 6th row. In some of the objects, it uses even fewer number of attempts compared to RL* policy. However, its success rate drops to 89% for hexagonal ellipse and further drops to around 55% for the three cuboid shaped objects (cuboid, phone charger, paper box). According to our observation, the policy with F/T sensor is good at localizing contact errors in x and y directions, but has trouble to distinguish the rotation error. Compared to F/T sensor, the tactile sensor observes not only the force information, but

also the object motion in the contact phase, which may contain useful information to differentiate the rotation error.

## VI. CONCLUSION

In this paper, we proposed a RL insertion policy with tactile feedback from 2 GelSlim [8] tactile sensors without prior knowledge of the object's geometry. We show that our proposed RL policy with designed curriculum training and tactile flow representation, provides several advantages over other tactile-based baseline policies, including: 1) supervised learning policy, 2) RL policy without curriculum, 3) RL policy with RGB input. Furthermore, we also showed the advantage of using tactile over F/T sensors in terms of the generalization, which are the most widely used sensors for these kind of applications.

## REFERENCES

[1] M. T. Mason, "Toward robotic manipulation," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 1–28, 2018.

[2] K.-T. Yu and A. Rodriguez, "Realtime state estimation with tactile and visual sensing for inserting a suction-held object," in *2018 IEEE/RSJ IROS*. IEEE, 2018, pp. 1628–1635.

[3] A. Bicchi, J. K. Salisbury, and D. L. Brock, "Contact Sensing from Force Measurements," *The International Journal of Robotics Research*, vol. 12, no. 3, p. 249, 1993.

[4] S. Li, S. Lyu, J. Trinkle, and W. Burgard, "A comparative study of contact models for contact-aware state estimation," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[5] L. Manuelli and R. Tedrake, "Localizing external contact using proprioceptive sensors: The contact particle filter," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[6] D. Ma, S. Dong, and A. Rodriguez, "Extrinsic contact sensing with relative-motion tracking from distributed tactile measurements," *arXiv preprint arXiv:2103.08108*, 2021.

[7] S. Dong and A. Rodriguez, "Tactile-based insertion for dense box-packing," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[8] E. Donlon, S. Dong, M. Liu, J. Li, E. Adelson, and A. Rodriguez, "Gelslim: A high-resolution, compact, robust, and calibrated tactile-sensing finger," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[9] D. E. Whitney, "Quasi-Static Assembly of Compliantly Supported Rigid Parts," *Journal of Dynamic Systems, Measurement, and Control*, vol. 104, no. 1, pp. 65–77, 03 1982.

[10] D. Whitney and J. Nevins, "Research on advanced assembly automation," *Computer*, vol. 10, no. 12, pp. 24–38, dec 1977.

[11] H. Bruyninckx, S. Dutre, and J. De Schutter, "Peg-on-hole: a model based solution to peg and hole alignment," in *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, vol. 2.

[12] M. E. Came, T. Lozano-Pérez, and W. P. Seering, "Assembly strategies for chamferless parts," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1989, pp. 472–477.

[13] W. S. Newman, Y. Zhao, and Y. H. Pao, "Interpretation of force and moment signals for compliant peg-in-hole assembly," in *IEEE International Conference on Robotics and Automation*, vol. 1, 2001, pp. 571–576.

[14] V. Gullapalli, "Learning control under extreme uncertainty," in *Advances in neural information processing systems*, 1993, pp. 327–334.

[15] S. Levine, N. Wagener, and P. Abbeel, "Learning contact-rich manipulation skills with guided policy search," *CoRR*, vol. abs/1501.05611, 2015. [Online]. Available: http://arxiv.org/abs/1501.05611

[16] J. Luo, E. Solowjow, C. Wen, J. A. Ojea, and A. M. Agogino, "Deep reinforcement learning for robotic assembly of mixed deformable and rigid objects," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2062–2069.

[17] G. Schoettler, A. Nair, J. Luo, S. Bahl, J. A. Ojea, E. Solowjow, and S. Levine, "Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[18] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks," in *International Conference on Robotics and Automation (ICRA)*, 2019.

[19] Y. Liu, D. Romeres, D. K. Jha, and D. Nikovski, "Understanding multimodal perception using behavioral cloning for peg-in-a-hole insertion tasks," *arXiv preprint arXiv:2007.11646*, 2020.

[20] S. Dong, D. Ma, E. Donlon, and A. Rodriguez, "Maintaining grasps within slipping bound by monitoring incipient slip," in *International Conference on Robotics and Automation (ICRA)*, 2018.

[21] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1587–1596.