Generalizable and Robust Tactile Pushing using Sim-to-Real Deep Reinforcement Learning

Max Yang, Yijiong Lin, Alex Church, John Lloyd, Dandan Zhang, David A.W. Barton*, Nathan F. Lepora*

***Abstract*— Object pushing presents a key non-prehensile manipulation problem that is illustrative of more complex robotic manipulation tasks. While deep reinforcement learning (RL) methods have demonstrated impressive learning capabilities using visual input, a lack of tactile sensing limits their capability for fine control during manipulation. Here we propose a deep RL approach to object pushing using tactile and proprioceptive states, namely tactile pushing. We present a goal-conditioned formulation that allows both model-free and model-based RL to obtain accurate policies for tactile pushing. To achieve real-world performance, we adopt a sim-to-real approach. Our results demonstrate that it is possible to train on a single object and a limited sample of goals to produce precise and reliable policies that can generalize to a variety of unseen scenarios without domain randomization.**

I. Introduction

As the demand for versatile controllers for robot manipulation increases, reinforcement learning (RL) has become an attractive option due to its generality and ability to model complex relationships. However, most RL studies on pushing have relied on vision-based systems [1], [2] which can suffer from low accuracy and occlusions. On the other hand, tactile sensing can capture detailed contact information regarding robot-object interactions, enabling precise control of contact.

In this study, we examine the problem of pushing an object to a single arbitrary distant goal, which allows the agent to search for the shortest path to the goal. The success of training RL policies relies heavily on using features that are important for the task. When using optical tactile sensors, instead of learning directly from tactile images, we use pose-based observations which provide specific pushing-related contact features that can be more efficient for learning. We formulated the tactile pushing task as a goal-conditioned RL problem and demonstrated its suitability for both model-free RL and model-based RL with online planning to obtain accurate and reliable pushing policies. To achieve real-world tactile pushing, we leverage the tactile sim-to-real framework developed by Church et al. [3]. Our approaches can learn real-world pushing policies by training entirely in simulation from scratch, without any real-world pushing data nor requiring any exploration policy for system identification.

NL was supported by an award from the Leverhulme Trust on 'A biomimetic forebrain for robot touch' (RL-2016-39).

All authors are with the Department of Engineering Mathematics and Bristol Robotics Laboratory, University of Bristol, Bristol BS8 1UB, U.K. (email: {max.yang, yijiong.lin, alex.church, john.lloyd, ye21623, david.barton, n.lepora}@bristol.ac.uk)

* These authors contributed equally.

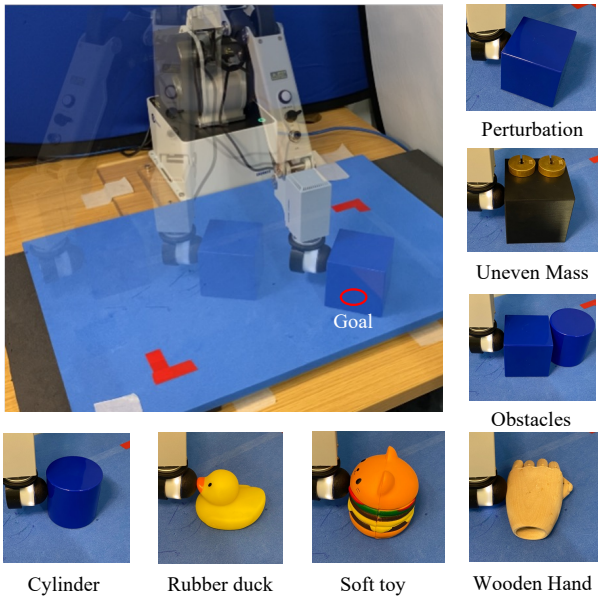

Fig. 1: Real-world object pushing setup, comprising a desktop robot (Dobot MG400) mounted on a pushing platform with a Tactip attached to serve as the pusher.

II. Methodology

A. Reinforcement Learning Methods

We formulate this task as a finite horizon goal-conditioned Markov Decision Process (MDP) defined by a continuous state $s \in \mathcal{S}$, a continuous action space $a \in \mathcal{A}$, a probabilistic state transition function $p(s_{t+1}|s_t, a_t)$, a reward function $r \in \mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \to \mathbb{R}$, and a goal space $\mathcal{G}$ with goal $g \in \mathcal{G}$.

For model-free RL, the aim is to obtain a policy $\pi^*_\theta(a_t|s_t, g)$ parameterized by θ that maximizes the expected return over an episode τ and goals g. A goal is randomly sampled at the start of the episode and remains fixed until the episode ends. We train using an off-policy algorithm, Soft Actor-Critic (SAC) [4].

For model-based RL with online planning, the objective is to learn a reliable forward dynamics model f_θ parameterized by θ to approximate the probabilistic state transition function $p(s_{t+1}|s_t, a_t)$. The learned model is then used for online planning in an MPC framework. We use Probabilistic Ensemble Trajectory Sampling (PETS) [5] which uses cross-entropy method (CEM) as the MPC optimizer.

B. Task Formulation

*1) **Tactile Observations**:* We construct two goal-aware observations ($\mathcal{S}_1$ and $\mathcal{S}_2$) for model-free policies using the tactile image I_{tactile} and tactile pose P_{tactile} respectively. We

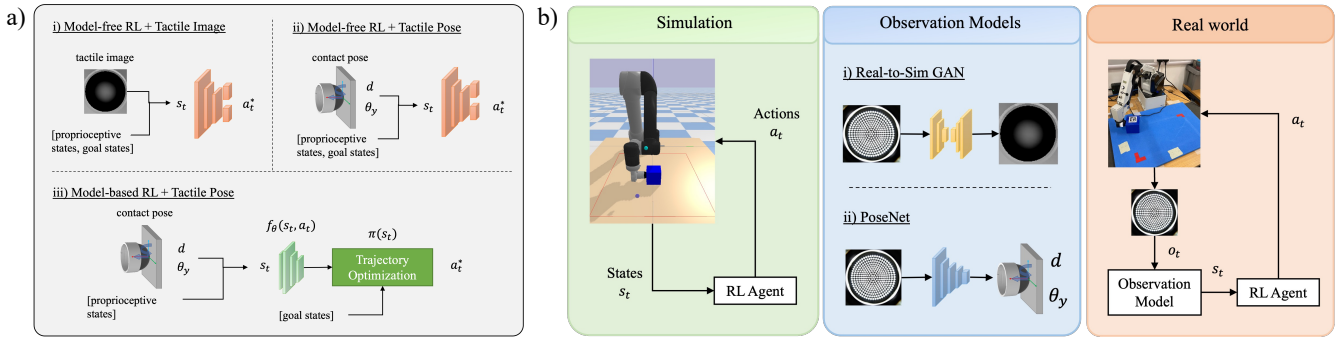

Fig. 2: Overview of modelling and sim-to-real transfer. (a) The three tactile RL pipelines: i) a model-free agent trained on tactile-image-based observations, ii) a model-free agent trained on pose-based observations, and iii) a model-based agent trained on pose-based observations. (b) Workflow for sim-to-real tactile RL. This involves training the RL policy in simulation with tactile observations, then collecting tactile data and training an observation model to close the sim-to-real gap, and finally, the RL policy and observation model are combined for real-world implementation.

also used tactile pose to construct a pose-based state space ($\mathcal{S}_3$) for the model-based agent to learn a dynamics model.

$$\begin{aligned} \mathcal{S}_1 &= [I_{\text{tactile}},\ x_g^p,\ y_g^p,\ \theta_g^p], \\ \mathcal{S}_2 &= [P_{\text{tactile}},\ x_g^o,\ y_g^o,\ \theta_g^o], \\ \mathcal{S}_3 &= [P_{\text{tactile}},\ x_o,\ y_o,\ \theta_o], \end{aligned} \tag{1}$$

where the state variables x, y represent the position, θ the orientation, all with respect to workframe of the workspace. Figure 2a) demonstrates the construction of different input observations from a tactile image and their integration with model-free and model-based RL.

*2) **Action Space**:* We define the action space $\Delta y \in$ [-1mm, 1mm] and $\Delta\theta \in [-1°, 1°]$, representing the change in position and orientation in the pusher's frame of reference. We fix the forward velocity of the pusher to have a constant $\Delta x = 1$mm action.

*3) **Reward Shaping**:* To shape the reward, we define a desired orientation as the object-goal bearing angle $g_\theta = \text{atan2}(o_{xy}, g_{xy})$, and use distance functions $f(a, b)$ and $g(a, b)$ to represent the Euclidean and cosine distances between a and b, respectively. Our reward function contains components $f(o_{xy}, g_{xy})$, the Euclidean distance between contact and goal positions, $g(o_\theta, g_\theta)$, the cosine distance between the contact surface and the goal orientation, and $g(p_\theta, o_\theta)$, the cosine distance between the pusher and contact surface orientation:

$$r = \begin{cases} -(g(o_\theta, g_\theta) + g(p_\theta, o_\theta)) & \text{if } ||o_{xy} - g_{xy}|| > d, \\ -(f(o_{xy}, g_{xy}) + g(p_\theta, o_\theta)) & \text{if } ||o_{xy} - g_{xy}|| \leq d. \end{cases} \tag{2}$$

C. Sim-to-Real Transfer

For real-world deep RL with tactile information, we adopt the sim-to-real framework comprising 3 main steps: 1) training the RL agent in simulation, 2) collecting tactile data and training observation models to translate across the sim-to-real gap, and 3) performing zero-shot sim-to-real policy transfer. As the framework was designed for tactile image observations, here we achieve sim-to-real for pose-based tactile observations by adapting it with a PoseNet [6] where step 2) is replaced with pose training (workflow of the sim-to-real procedure shown in Fig. 2b).

III. Experiments and Results

The training result is shown in table I. Model-free RL achieved asymptotic performance with 100× more training samples needed as compared to model-based RL. However, with significantly more training, the final policy achieved better rewards. We also found the tactile-pose-based model-free agent to be more sample efficient and achieved better final performance as compared to the tactile-image-based agent, suggesting tactile pose is a more effective feature for learning to push objects.

TABLE I: Training samples for within 10% of best reward and final best rewards.

RL Agent	Samples	Best Reward
Model Free + Tactile Image	3.2m	-124.86
Model Free + Tactile Pose	2.8m	**-122.85**
Model Based + Tactile Pose	**25k**	-144.70

We also tested each agent in the real world using their respective observation models for sim-to-real transfer. The experiment setup is shown in Fig. 1. We found that the tactile-pose-based model-free agent performed the best, generalizing to test objects and disturbances not seen during training better than other agents.

IV. Conclusions

In this paper, we proposed several successful deep RL methods for the tactile pushing problem. This relied on a problem formulation that allowed us to obtain accurate and reliable policies for goal-conditioned pushing using deep RL. A key finding from this study was that RL policies trained without domain randomization or a diverse training curriculum that involved multiple objects were able to perform a wide range of pushing tasks, displaying strong generalization skills. This suggests that tactile information when used appropriately, can facilitate the efficient learning of general manipulation skills, and so make previously intractable robot manipulation problems tractable.

REFERENCES

[1] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4238–4245.

[2] L. Manuelli, Y. Li, P. Florence, and R. Tedrake, "Keypoints into the future: Self-supervised correspondence in model-based reinforcement learning," *arXiv preprint arXiv:2009.05085*, 2020.

[3] A. Church, J. Lloyd, R. Hadsell, and N. Lepora, "Tactile Sim-to-Real Policy Transfer via Real-to-Sim Image Translation," in *Proceedings of the 5th Conference on Robot Learning*. PMLR, Oct. 2021, pp. 1645–1654.

[4] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[5] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," *Advances in neural information processing systems*, vol. 31, 2018.

[6] N. F. Lepora and J. Lloyd, "Optimal deep learning for robot touch: Training accurate pose models of 3d surfaces and edges," *IEEE Robotics & Automation Magazine*, vol. 27, no. 2, pp. 66–77, 2020.