Leveraging Touch for Dexterous Behavior Discovery and Deployment

Antonia Bronars[1], Nimrod Gileadi[2], Jose Enrique Chen[2], Zilin Si[3],
Francesco Nori[2], Nicolas Heess[2], Konstantinos Bousmalis[2], Maria Bauza[2].

***Abstract*— We study the influence of contact information in solving a high-precision plug insertion using a robot arm equipped with a dexterous hand. We discover contact-rich, dexterous behaviors to reorient, grasp, and insert the plug combining tactile and reinforcement learning (RL) in simulation. We then distill the RL policy with behavior cloning, and find that contact information is crucial for efficiently learning behaviors that transfer well to the real world. We also study and discuss the importance of exposing the behavior cloning agent to contact information. Ultimately, we demonstrate a 6x performance improvement over prior work [1] through our improved behavior discovery and deployment pipeline.**

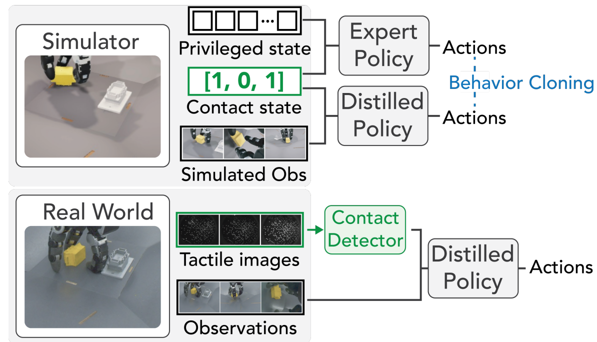

Fig. 1: **Method overview.** We train an RL agent to discover behaviors with access to various types of contact information. We then distill the behaviors and deploy the policies in the real world, inferring contact states from onboard tactile sensors.

I. INTRODUCTION

Synthesizing dexterous behaviors for robotic systems with multi-fingered hands is an important goal, but remains challenging for several reasons. For one, the lack of intuitive interfaces for teleoperation [2], [3] renders standard behavior cloning pipelines insufficient. Reinforcement learning (RL) offers a promising alternative to autonomously discover these behaviors, but the exploration problem suffers from high-dimensional action spaces [4]. In this work, we leverage a recently proposed method for demonstration-guided RL [1] to synthesize behaviors to solve a precise, contact-rich plug insertion task. We investigate the role of contact information in shaping the behaviors RL discovers. Furthermore, transferring learned policies from simulation to the real world remains a major challenge [5]. We examine how contact information affects sim2real transfer, and explore its potential to bridge the gap between simulated and real world execution. We find that contact information is crucial for efficiently learning behaviors that transfer well to the real world. We also find and discuss why then there is only a small positive benefit from also including contact information during policy distillation.

II. METHOD

Behavior discovery with RL. We leverage demonstration-guided RL [1] to discover behaviors for solving the plug insertion task (Figure 1). The RL agent is an MPO agent, and the sparse reward function is a binary success detector for inserting the plug into the socket. The agent observes: robot (including hand) joint positions, robot tool center point (TCP) position and orientation, plug pose, and socket pose. For some experiments, we also give the agent access to privileged information like robot joint velocities, hand finger joint torques, hand finger joint velocities, and commanded finger joint positions. We consider these observations priviledged because although they can be used to infer contact information in simulation, they are difficult to measure in the real world or are subject to a large sim2real gap. For a subset of experiments, we also give the agent access to binary touch sensors on each of the fingertips. These sensors directly measure contact interactions and are subject to a small sim2real gap. The action space of the robot is the 6 degree of freedom TCP velocity for the robot, and 12 degree of freedom joint positions for the hand.

Behavior deployment via teacher-student distillation. We use a teacher-student framework to distill the RL policy to operate on simulated sensor observations, then directly roll out the distilled policy in the real world. We use the RL agent to collect an offline dataset of trajectories, and train the student with behavior cloning (BC). The BC agent has access to five cameras (two wrist cameras, and three overhead cameras), robot proprioception (robot and hand joint positions), and the robot TCP pose. In a subset of our experiments, we also give it access to per-finger binary touch.

1) Distillation algorithm: We use ACT [6], with a modification of the loss function to make it more suitable for sim2real. Instead of weighting all action predictions in the chunk equally, we weight by the rollout temporal aggregation matrix so that immediate predictions are prioritized over future predictions. Empirically, we observed re-weighting the loss function with $k = -0.5$ balances smoothness and precision in the action predictions. This is useful when the dataset comes from RL rollouts, which have different characteristics (no pauses, unimodal behavior) than teleoperated demonstrations.

2) Data mixture and training augmentations: We train on one million simulated trajectories, where 70% of trajectories

[1]Massachusetts Institute of Technology, bronars@mit.edu
[2]Google DeepMind, [3]Carnegie Mellon University

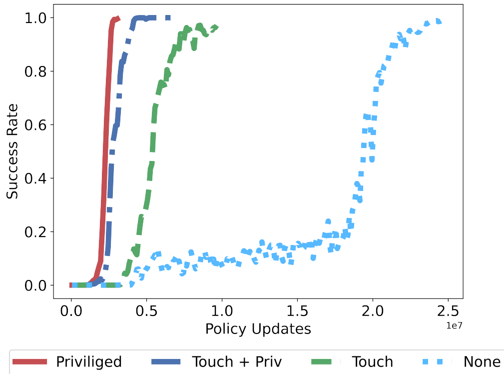

Fig. 2: **Contact information is important for discovering successful behaviors efficiently.** The only policy that takes significantly longer than the others to converge contains no information from which contact states can be inferred.

are rendered with visual domain randomization and 30% are rendered with photorealistic rendering using Filament [7] (see [1] for details on the visual randomization). To improve sim2real transfer, we found it useful to train with gaussian noise on the proprioception and images. We also apply 30% dropout on the images and 10-30% on the touch signals. Dropping out the touch signals provides robustness against missed contacts which can occur in sim2real.

III. EXPERIMENTS AND RESULTS

Policy convergence results. We first compare the convergence rates of the RL policy with four types of contact information: PRIVILEGED (robot joint velocities, hand finger joint torques, hand finger joint velocities, and commanded finger joint positions), TOUCH, TOUCH + PRIVILEGED, and NONE. In each case, we also expose robot proprioception and the object poses. We find that contact information is important for discovering behaviors efficiently (Figure 2), as the only policy that takes significantly more updates to converge is NONE.

Insertion experiments. We also evaluate the success rate of a plug insertion task in simulation and the real world. The task involves inserting the plug from an arbitrary pose in the basket, which may involve re-orienting the object in-hand or by leveraging non-prehensile contacts. We consider three outcomes: success, near success, and failure (Figure 3). In near successful trials, the robot partially inserts the plug, and holds the plug partially inserted for the remainder of the trial. We evaluate two behaviors, TOUCH + PRIVILEGED and PRIVILEGED, each distilled with and without access to touch information in the observation.

1) Simulation results: We evaluate the success rate of 100 trials per policy, where each trial lasts 10 seconds. We find the performance of each policy is similar (Figure 5, right).

2) Real-world experimental setup: We use a Kuka LBR iiwa 14 robot arm with a three-finger DEX-EE Hand, which has camera-based fingertip tactile sensors and two wrist cameras (Figure 4). We detect contact information from the fingertip tactile sensors by thresholding the average pixel-wise

Fig. 3: **Experimental outcomes.** We consider three outcomes for the plug insertion trials: success (left), near success (middle), and failure (right).

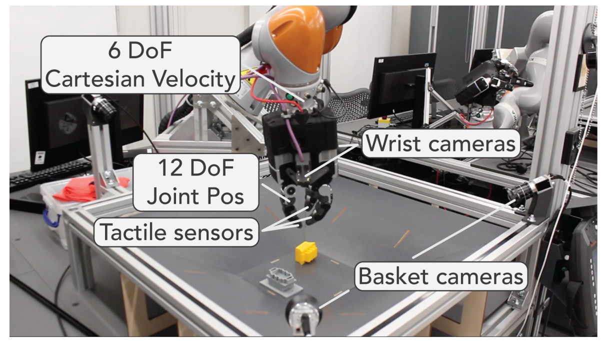

Fig. 4: **Experimental setup.** We perform a high-precision plug insertion task using a Kuka LBR iiwa 14 robot arm with a Dex-EE hand.

difference between observed tactile images and a non-contact image. We also attach three cameras to the environment. We evaluate the performance of each policy with 50 real-world trials, each lasting 30 seconds.

3) Real-world results: We find that TOUCH + PRIVILEGED is the more robust behavior, resulting in 88% successful or near successful trials when distilled with touch, and 86% when distilled without touch (Figure 5). We find that the policy performance is not strongly influenced by the inclusion of touch during distillation. The intuition behind this unexpected finding is that including touch during RL behavior discovery tends to produce intrinsically robust strategies for grasping, which requires less of touch sensing during execution.

The PRIVILEGED (no touch) behavior, on the other hand, results in 68% successful or near successful trials when distilled with touch, and 54% when distilled without touch (Figure 5). We see a larger performance boost when including touch as an observation during distillation (Figure 5). Because this behavior is less intrinsically robust, it may be more important to directly supervise each point of contact during execution.

REFERENCES

[1] M. Bauza, J. E. Chen, V. Dalibard, N. Gileadi, R. Hafner, M. F. Martins, J. Moore, R. Pevceviciute, A. Laurens, D. Rao, M. Zambelli, M. Riedmiller, J. Scholz, K. Bousmalis, F. Nori, and N. Heess, "Demostart: Demonstration-led auto-curriculum applied to sim-to-real with multi-fingered robots," 2024. [Online]. Available: https://arxiv.org/abs/2409.06613

[2] Y. Park, J. S. Bhatia, L. Ankile, and P. Agrawal, "Dexhub and dart: Towards internet scale robot data collection," 2024. [Online]. Available: https://arxiv.org/abs/2411.02214

[3] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu, "Dexcap: Scalable and portable mocap data collection system for dexterous manipulation," *arXiv preprint arXiv:2403.07788*, 2024.

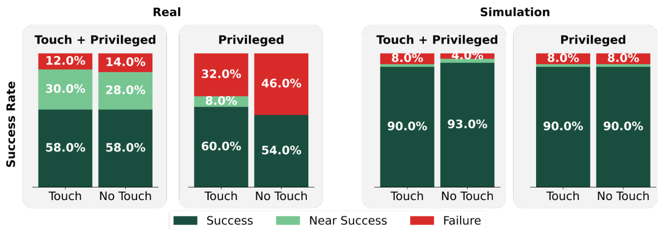

Fig. 5: **Touch helps discover behaviors that transfer sim2real better.** We find that discovering behaviors with touch biases the policy toward behaviors that transfer better to the real world.

[4] T. Lin, K. Sachdev, L. Fan, J. Malik, and Y. Zhu, “Sim-to-real reinforcement learning for vision-based dexterous manipulation on humanoids,” 2025. [Online]. Available: https://arxiv.org/abs/2502.20396

[5] J. Wang, Y. Yuan, H. Che, H. Qi, Y. Ma, J. Malik, and X. Wang, “Lessons from learning to spin "pens",” 2024. [Online]. Available: https://arxiv.org/abs/2407.18902

[6] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, “Learning fine-grained bimanual manipulation with low-cost hardware,” 2023. [Online]. Available: https://arxiv.org/abs/2304.13705

[7] Google, “Filament: A real-time physically based rendering engine,” 2025, version 1.52.0. [Online]. Available: https://github.com/google/filament/