

Sim-to-real Deep Reinforcement Learning for Low-cost High-resolution Optical Tactile Sensing

Yijiong Lin, John Lloyd, Alex Church, Nathan F. Lepora

Abstract—High-resolution optical tactile sensors are increasingly used in robotic learning environments due to their high resolution and good match to deep reinforcement learning systems that rely on computer vision. However, there is a high barrier of entry to research in this area due to the high cost of tactile robot platforms, specialised simulation software, and learning algorithms that lack versatility. In this letter, we extend the Tactile Gym sim-to-real deep reinforcement learning platform to include an inexpensive industrial 4-DoF robot arm and two of the most popular optical tactile sensors (TacTip and DIGIT/GelSight). We validate the extended environment on three physically-interactive tasks requiring tactile sensing: object pushing, edge following and surface following. The results of our experimental validation highlight some differences between these sensors, which may help future researchers select and customize the physical characteristics of tactile sensors for different manipulations scenarios.

Index Terms—Force and tactile sensing; Deep Learning; Dexterous Manipulation

I. INTRODUCTION

A plausible route to human-like robot dexterity is to combine deep learning with high-resolution tactile sensing, given the unprecedented recent advances in controlling robots with deep learning applied to robot vision [1]. Moreover, the use of deep reinforcement learning (RL) would seem to offer the potential for learning complex manipulation tasks based on a reward information, which is both a mechanism for how humans acquire new skills and has achieved impressive results in simulated environments such as computer games. However, there are major challenges preventing tactile deep RL from being realised: (1) the lack of available and accessible tactile sensing technologies limits the research capacity available to develop RL methods for touch; (2) those labs that have expertise in fabricating tactile sensors tend to stay with the technology where they have expertise; (3) hence, approaches to tactile deep RL, e.g. [2]–[4], stay confined within those labs, which is inefficient for progressing the field and opposite to the open culture that has benefitted AI research.

Meta AI researchers have developed and open-sourced a tactile robot learning platform called TACTO [5] and tactile processing libraries PyTouch [6] for GelSight-based tactile sensors such as the low-cost, open-sourced DIGIT tactile

YL was supported by the China Scholarship Council (CSC)/University of Bristol joint-funded scholarship. AC was supported by an EPSRC CASE award sponsored by Google DeepMind. NL and JL were supported by a Leadership 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, U.K. (email: {yijiong.lin, j.lloyd, a.church, n.lepora}@bristol.ac.uk)

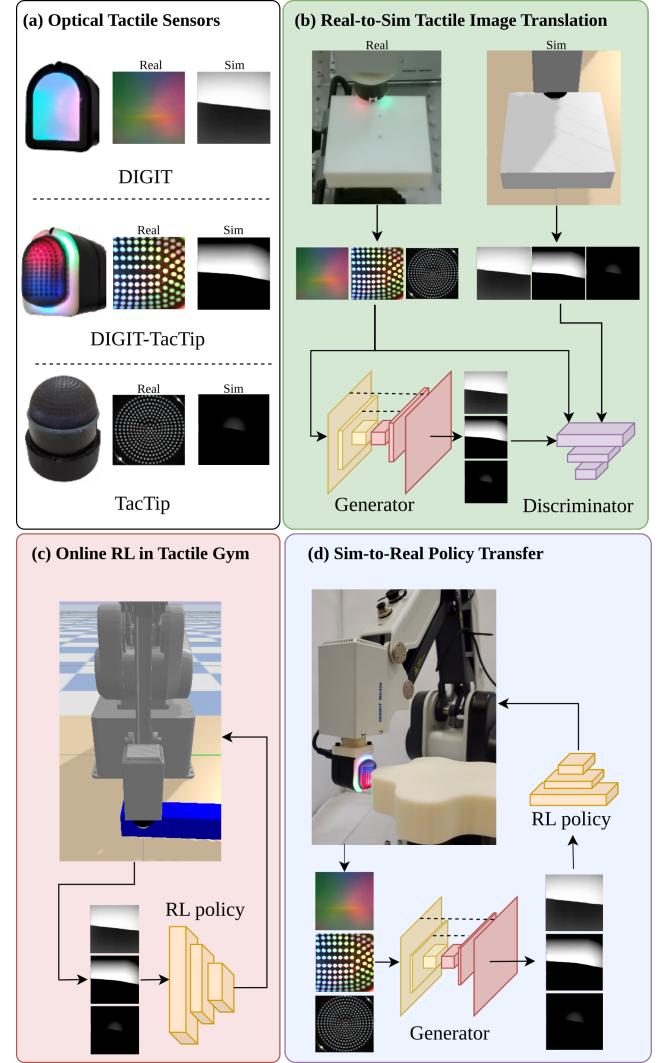


Figure 1. Overview of the tactile sim-to-real deep RL robotic system: (a) Three low-cost high-resolution optical tactile sensors and the raw sensor images. (b) Training real-to-sim tactile image translation. (c) Policy learning in the Tactile Gym with simulated tactile images. (d) A desktop robot equipped with the DIGIT-TacTip performing the surface-following task by translating real tactile images to simulated images for the RL policy.

sensor [7]. Meanwhile, researchers in Bristol Robotics Laboratory have developed a family of high-resolution biomimetic optical tactile sensors called the TacTip [8], [9], alongside an open-sourced suite of learning environments called Tactile Gym that features highly-efficient tactile image rendering [2]. In consequence, sim-to-real policy transfer via tactile image translation enabled zero-shot performance on multiple exploration and manipulation tasks requiring tactile feedback,

such as delicately tracing the edges and surfaces of complex objects and pushing objects to goal locations.

The goal of this present research is to bring together this progress with GelSight-type sensors with the sim-to-real deep RL methods developed with the TacTip. Whilst it was previously claimed that the approach used in [2] should be applicable to “a broad range of other high-resolution optical tactile sensors including sensors of the Gelsight type”, the approach was only demonstrated with the TacTip. Here we extend the approach to a hybrid form of the TacTip and DIGIT (the ‘DIGIT-TacTip’) and a sensor of the Gelsight-type (DIGIT), despite the significantly different tactile images produced by these sensors. We do this by extending the methods to include simulations of these three distinct tactile sensors, utilising the CAD files for 3D-printing the sensors.

This main contributions of this work are as follows:

- 1) We extend the Tactile Gym open-source simulation platform with three low-cost, high-resolution optical tactile sensors. Using this environment, we successfully learn deep RL control policies for three physically-interactive tasks (edge following, surface following and pushing). This validates and extends earlier results for this platform while making it more accessible and applicable to the wider research community.
- 2) We compare the performance of the three optical tactile sensors across the three manipulation tasks, identifying the strengths and weaknesses of each sensor in these different scenarios. To the best of our knowledge, this is the first empirical comparison of optical tactile sensors for tactile manipulation, and these results may help other researchers select the most appropriate tactile sensor for a particular task.
- 3) We also introduce a low-cost and widely-available desktop robot for implementing and testing task performance, further increasing the accessibility of this research. This also raised some challenges in adapting the manipulation tasks that originally used industrial robots to a robot with less degrees of actuation and a more limited workspace.

This paper accompanies a second paper that covers in detail the design, fabrication and testing of the DIGIT-TacTip sensor used here, and commits to openly releasing all software and hardware developments in these two papers.

II. RELATED WORK

1) Deep Reinforcement Learning in Tactile Robotics: Deep reinforcement learning (RL) has proved successful in solving many sequential decision making problems in robotics, particularly those with high-dimensional observation spaces such as in computer vision [1]. Some work has explored tactile RL with low-resolution tactile sensing to perform tasks such as object stabilisation [10] and learning a forward predictive model [11]. Work in [12] proposed a deep tactile model-predictive RL framework for learning how to re-position an object using a Gelsight-style tactile sensor, and [3] used Twin Delayed DDPG (TD3) [13] to learn a general tactile-guided insertion policy in the physical environment from a sequence of tactile images. A follow-up study simplified the observation space containing the raw tactile images and robot proprioceptive data for the deep RL policy by learning an

extrinsic contact line model for contact localization. This policy was learned in a simulation environment [14].

Recently, a tactile simulation environment for deep RL, called Tactile Gym [2], has successfully applied the trained policies to some challenging tasks such as object pushing and rolling in a real physical environment. The transfer from simulation to real-world physical environment was facilitated using a novel real-to-sim tactile image translation technique, in a zero-shot manner. Tactile-based deep RL has also been successfully applied to robotic service tasks like learning to type on a braille keyboard [15] with the TacTip optical tactile sensor and learning to play the piano [4] with the DIGIT tactile sensors.

2) Tactile Sim-to-real Transfer: The tactile sim-to-real gap significantly hinders the application of learned policy in simulation to reality. Two research directions have been explored to close this gap: using the Finite Element (FE) method to model the sensor deformation dynamics [16]–[21], or leveraging the image rendering method to replicate the sensory data [2], [5], [22]. For a more thorough review we refer to [2]. In the present work, we follow the tactile sim-to-real method described in [2] by using depth image-rendering and image translation for two popular optical tactile sensor classes: GelSight-type which is based on image shading [23] and TacTip type which is based on biomimetic marker-based transduction [9].

III. METHOD AND EXPERIMENTS

A. Tactile Robot System

In this paper, we use a tactile robot comprising a low-cost desktop robot arm with high-resolution tactile sensor mounted as an end effector. This is intended to be a lower-cost, desktop version of the setup used in Church et al [2] for investigating sim-to-real tactile deep RL, which used a 6-axis industrial robot from Universal Robotics. In this paper, we expand our treatment to compare three distinct optical tactile sensors: the DIGIT, DIGIT-TacTip and TacTip.

The robot platform and the operation of the tactile sensors are presented in detail elsewhere, with the main focus of the present paper on the application of sim-to-real deep RL on this low-cost desktop robot. Hence, we summarize just the main aspects of the tactile robot.

1) Desktop robot arm: We use a Dobot MG400 4-axis arm designed for affordable automation. The base and control unit has footprint 190 mm × 190 mm, payload 750 g, maximum reach 440 mm and repeatability ± 0.05 mm. As we describe later, the accuracy of tactile models trained using this arm is similar to larger industrial robot arms. The main constraint is that only the (x, y, z) -position and rotation around the z -axis of the end effector are actuated.

2) High-resolution optical tactile sensing: Here we consider three distinct optical tactile sensors:

- (a) The **TacTip**, a soft, curved, 3D-printed tactile skin with an internal array of pins tipped with markers, which are used to amplify the surface deformation from physical contact against a stimulus. For more details, we refer to ref. [8], [9].
- (b) The **DIGIT** shares the same principle of the Gelsight tactile sensor [24], but can be fabricated at low cost and is of a size

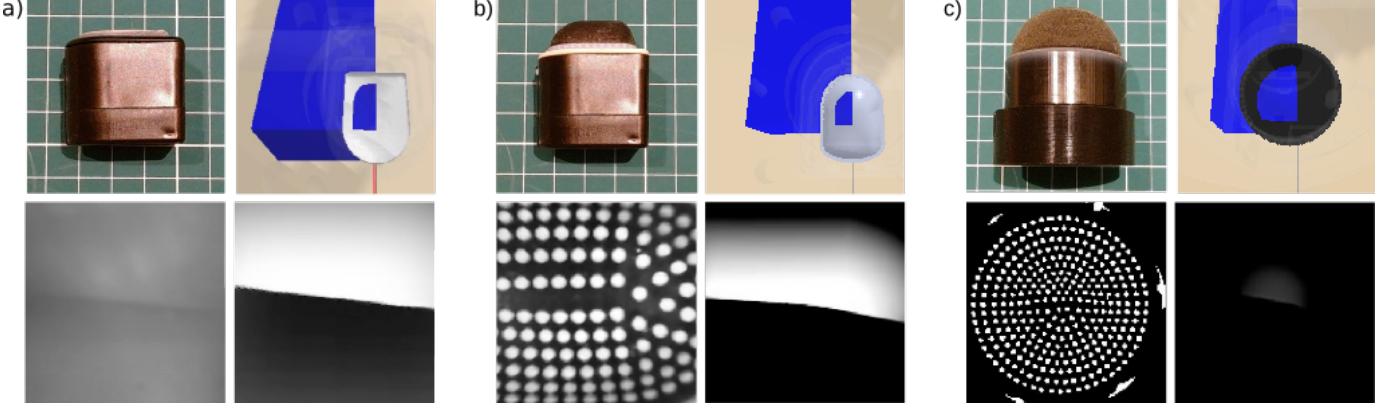


Figure 2. Comparison between the different optical tactile sensors: (a) DIGIT, (b) DIGIT-TacTip, (c) TacTip. For each sensor we show: top-left, the real sensor hardware; top-right, simulated contact geometry between sensor skin and a blue edge stimulus; bottom-left, tactile image gathered by the real sensor pressing onto an edge; and bottom-right, a generated depth image matching those gathered in simulation.

suitable for integration some robotic hands, such as on the fingertips of the Allegro hand [7].

(c) The **DIGIT-TacTip** is an adapted version of the DIGIT and the TacTip, whereby the 3D-printed skin of a TacTip is customized to integrated onto the DIGIT housing, while keeping the camera and lighting system. For the purposes of this paper, the sensor can be treated as outputting tactile images of the same resolution and sensing surface size as the DIGIT, but with a soft biomimetic skin like other TacTip sensors.

B. Tactile Sim-to-Real Deep RL Framework

Following Church et al [2] we take a sim-to-real deep reinforcement learning approach to achieve the desired robot behaviour on physical hardware. This approach consists of three distinct phases:

- 1) An online learning in simulation phase, where deep RL is applied to images captured by a simulated tactile sensor for the learning of several distinct tasks (here edge following, surface following and object pushing).
- 2) A domain adaption phase where a model is learned for the translation of images captured by a real tactile sensor to images captured by the simulated sensor.
- 3) A zero-shot sim-to-real phase where policies learned in simulation are transferred to the real hardware using the networks trained in the previous two steps. An overview of this approach is shown in Fig. 1 and more details can be found in the original reference [2].

To adapt the existing approach to our setting, we needed to make several changes. In this work we chose to use a low-cost desktop robot arm as described above (Sec.III A). This arm has only 4 actuated axes, whereas previous work used a 6-axis industrial robotic arm (UR5, Universal Robotics). To facilitate the use of this lower degree-of-freedom arm we have adapted the tasks and data collection procedures while attempting to meet the challenge of successful performance on tasks originally developed for a more capable robot arm. In particular the previous surface following experiments made use of Roll and Pitch to accurately maintain a normal orientation to a surface. Instead, we rely on a custom 3D-printed flange so the sensor is mounted perpendicular to the end effector. In this

way, we can make use of the Yaw DoF of the DOBOT MG400 when maintaining normal orientation to a surface varying in only one direction, as shown in Fig. 1(d).

Moreover, whilst it was previously claimed that the approach used in [2] should be applicable to “a broad range of other high-resolution optical tactile sensors, including sensors of the Gelsight type”, it was only demonstrated with a hemispherical TacTip and a flat Tactip. Here we validate that the approach works with alternative forms of the TacTip sensor (DIGIT-TacTip) and sensors of the Gelsight type (DIGIT), despite the significantly different tactile images produced by these sensors. To achieve this we extended the simulation to include simulated sensors that match real hardware, utilising the CAD files used for 3D-printing.

C. Deep RL with Different Tactile Dynamics in Simulation

The sim-to-real deep RL framework relying on the Tactile Gym described above simulates the contact dynamics by rigid-body physics and renders depth images as tactile information from specific sensors. Hence, the previously-learned tactile-feedback policies cannot be immediately applied to different types of optical tactile sensor other than the standard TacTip originally considered.

Hence we extend the Tactile Gym with two new virtual optical tactile sensors: DIGIT and DIGIT-TacTip (Fig. ?? a,b), based on their open-source CAD files [7] used for 3D-printing the DIGIT. We follow the method described in [2] to efficiently capture the depth images as tactile images by synthetic cameras embedded within those sensors.

The three distinct tactile observation spaces corresponding to the three type of sensor (see Fig. 2) are then used as input to train polices for the tasks, following the methods detailed in [2]. We report on the training results for the deep RL policies in Sec. IV.

D. Sim-to-real Transfer for Tactile Images

Here we aim for zero-shot learning so the learned policy in simulation can be transferred to the real-world task without further training. Hence it is essential to have a model to



Figure 3. Tactile stimuli used in manipulation tasks: (a) square, clover leaf, and teardrop stimuli for the edge following task; (b) arch, flower, and disc for the (side) surface following task; (c) blue cylinder, yellow triangular prism, and red cube for the object pushing task (with attached ArUco markers for ground truth tracking).

bridge the gap between the simulated and real domains. The rapid advances in Generative Adversarial Networks (GANs) enables realistic image generation, which we leverage for a image-to-image translation GAN [25] applied to real-to-sim tactile image translation. The network takes advantage of the U-net architecture [26] for the image-conditioned generator to better inference the spatial features during training, alongside a standard convolutional network [27] with batch normalization [28] for the discriminator.

Since each of the three optical tactile sensor considered here have a different design and illumination, we need to fine-tune the image-preprocessing and hyper-parameters for each sensor. The arrays of markers in the TacTip and the DIGIT-TacTip appear to reflect the light more clearly, which eases the fine-tuning of this image processing compared to the DIGIT, where the image shading is more subtle. However, after tuning we find all sensors to work effectively for sim-to-real transfer, as we cover in the results.

Table I

Sensor pose sampling ranges used during data collection. Sensor poses are expressed relative to a fixed coordinate frame attached to the training feature (Rz = axial rotation around z -axis).

Sensors	Contact Features			Edge		Surface	
	Axes	y(mm)	z(mm)	Rz(deg.)	x(mm)	Rz(deg.)	
Tactip		[-6,6]	[2,5]	[-179, 180]	[1,4]	[-15, 15]	
DIGIT-TacTip		[-5,5]	[2,4]	[-179, 180]	[1,3]	[-15, 15]	
DIGIT		[-5,5]	[2,3]	[-179, 180]	[1,2]	[-11, 11]	

E. Sim-to-real Data Collection

The three manipulation tasks considered here require distinct sim-to-real models across two distinguishing contact features: an edge for the edge-following task, and a surface for the surface-following and object-pushing tasks. For each edge and surface contact features, we collected two datasets for training and validation for each of the three considered tactile sensors, leading to twelve datasets in total.

Each training dataset comprised 5000 tactile images and each validation dataset 2000 tactile images, collected by using the desktop tactile robot (Sec.III A) to randomly contact data of the appropriate edge or surface feature. For the data collection, we used the flat edge and a flat surface of a 3D-printed flat stimulus with a straight edge, labeled with the random relative poses between the sensor and the stimulus when in contact. The movement ranges for each contact feature are shown in table I. These datasets take about 6 hours to collect on the physical robot and less than 1 minute to collect in simulation.

A further subtlety for the DIGIT and the DIGIT-TacTip is that they are no longer symmetric (being broader across one axis, unlike the original TacTip). This meant we needed to customize the range of (x, y) -pose data collection depending on the rotation angle, which was implemented by applying a tangent of the angle to the y -range orthogonal to the edge. In addition, we did not find it necessary to introduce the shear perturbation during the data collection that had been used in previous work.

F. Tactile Control Tasks

Here we adapt three tactile control tasks proposed in [2] to the desktop tactile robot: edge-following, surface-following and object-pushing. Though we expect our platform would also be viable for the ball-rolling task, we do not implement it in this work since the DIGIT-TacTip does not have a flat elastomer skin that is suitable for ball-rolling.

1) *Object Pushing*: This task aims for the robot to push an object through a sequence of goal positions along a trajectory on a flat surface. Three trajectories are considered: straight, curved and sinusoidal. In practise, each trajectory was divided into 10 equal-length sections with the final point on each section specified as the goal position; thus, there are ten goals for each trajectory. The 2-D action space comprising the x -position and rotation angle of the tool center point (TCP) located at the tip of the tactile sensor. We use three distinct objects with different shapes and weights (Fig. 3): a triangular prism, cube and cylinder varying in weight from 185 g to

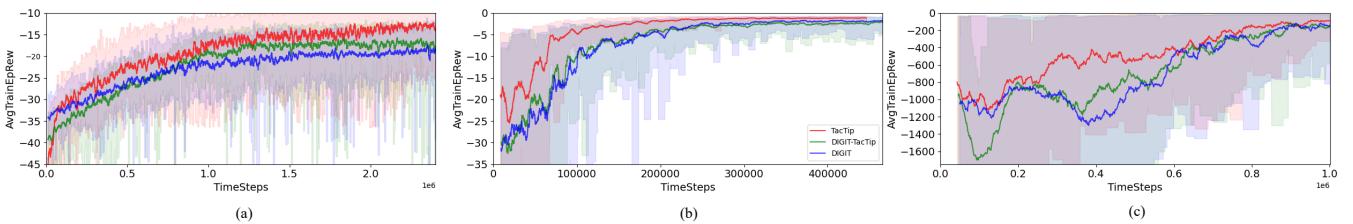


Figure 4. Averaged training performance of learned policies for different sensors in (a) edge-following, (b) surface-following, (c) object-pushing tasks.

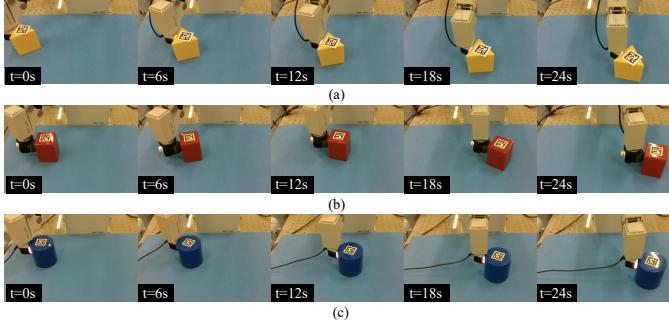


Figure 5. The tactile robot executing 3 pushing policies: (a) triangular prism along a straight line with the DIGIT-TacTip, (b) cube along a sinusoidal trajectory with the TacTip, (c) cylinder along a curve with the DIGIT.

363 g. These objects differ from the ones used in [2] which were somewhat lighter, having a mass of around 50 g, which is because we found the DIGIT and DIGIT-TacTip worked better with heavier objects due to the greater tactile deformation. An ArUco marker is placed on top of each object to track its movement for comparison with the ground truth using the tracking method described in [29].

2) *Edge Following*: This task aims for the robot to slide the tactile sensor along a contacted edge while maintaining a fixed distance between the edge and the TCP located at the tip of the tactile sensor. The 2D action space comprises the x and y position of the TCP. To evaluate the robustness and maneuverability of the tactile robot, we use three stimuli that contain interesting geometrical features such as straight edges, a positively/negatively curved edges and a right-angle corner (Fig. 3a). To measure tracking performance, the ground truth shapes are extracted from the CAD models of these 3D-printed objects by importing the CAD models into the Blender CAD software [30] and outputting the point clouds of their outside edges.

3) *Surface Following*: This task aims for the robot to slide the tactile sensor over a contacted surface while maintaining a fixed contact depth and orientating the TCP representing the tip of the sensor normal to that surface. The 2D action space comprises the y -position and rotation angle of the TCP. To evaluate the robustness and maneuverability of the tactile robot, we use the side surfaces of three objects comprising closed-loop 2D surfaces in 3D space (Fig. 3b) that contain interesting geometrical features such as locally planar, concave and convex surfaces. As with the edge-following task, the ground truth shapes and dimensions of the 3D-printed stimuli are obtained from the CAD models.

IV. RESULTS

A. RL Performance in the Simulated Environments

The deep RL method PPO [31] is here used to learn policies for the three simulated optical tactile sensors on the three considered tasks (pushing, edge-following, surface-following). Specifically, we use the Stable-Baselines-3 [32] implementation of PPO for training.

The results of training in the simulation are given in Fig. 4. Although there are slight differences in performance during training between the sensors, all the learned policies achieved

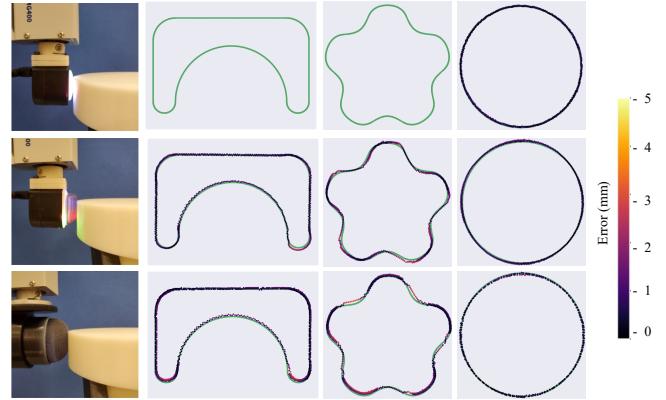


Figure 6. The tactile robot executing 3 surface-following policies on 3 distinct shapes, corresponding to the DIGIT (top row), DIGIT-TacTip (middle row) and TacTip (bottom row). The ground truth for each object is shown in green and errors of the traced contour from the ground truth are colour-coded (side colour bar). The DIGIT failed to follow the arch and flower objects.

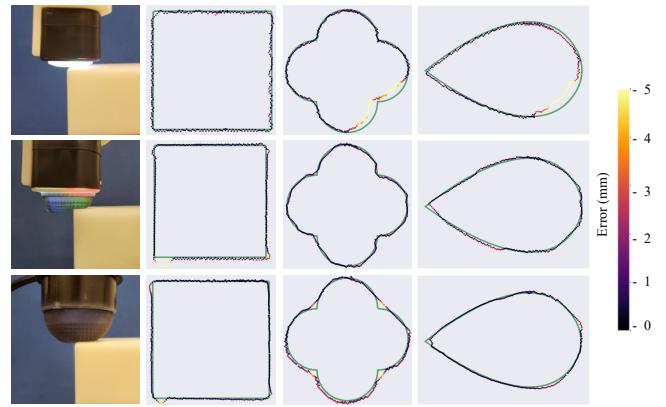


Figure 7. The tactile robot executing edge-following policies on 3 distinct shapes for the DIGIT (top row) DIGIT-TacTip (middle row) and TacTip (bottom row). The ground truth for each object is shown in green and errors of the traced contour from the ground truth are colour-coded (side colour bar).

similar accurate final performances in all tasks. We attribute these small differences to the different contact dynamics due to the different shapes of the tips for the three tactile sensors, but as we mentioned this does not affect the overall performance. In particular, we did not find it necessary to fine-tune the training hyperparameters for each tactile sensor. In addition, we apply the learned policy in each simulated task to the real robot later without any further fine-tuning.

B. Image Translation for Sim-to-real Transfer

We use the Structural Similarity Index (SSIM) to evaluate the quality of the translated images for both the edge and surface validation datasets (Table II). The high SSIM values close to unity (perfect match) obtained with the trained model

Table II
Mean SSIM values for image translation GANs trained with edge and surface features (values closer to 1.0 represent better image translation).

Features \ Sensors	Tactip	DIGIT-TacTip	DIGIT
Edge	0.9956	0.9953	0.9867
Surface	0.9927	0.9932	0.9818

Table III

Mean Euclidean distances of the actual trajectories from the ground-truth trajectories for the object pushing task. The numbers in bold denote the best result among the three sensors for the same object and trajectory. Failure cases are denoted "N/A".

Trajectories		Straight			Curve			Sine		
Sensors	Objects	Cube	Cylinder	Tri. Prism	Cube	Cylinder	Tri. Prism	Cube	Cylinder	Tri. Prism
Tactip	10.329 mm	9.211 mm	11.505 mm		12.191 mm	11.293 mm	13.202 mm	11.933 mm	11.301 mm	13.889 mm
DIGIT	11.201 mm	10.134 mm	N/A		12.942 mm	12.002 mm	N/A	12.410 mm	11.333 mm	N/A
DIGIT-TacTip	11.253 mm	10.244 mm	16.09 mm		13.009 mm	11.200 mm	16.414 mm	12.322 mm	11.481 mm	15.129 mm

Table IV

Mean Euclidean distances of the trajectories from the ground truth for the edge following task. The number in bold denotes the best result among the three sensors.

Sensors	Obj.	Square	Clover	Foil
DIGIT		0.875 mm	1.709 mm	1.821 mm
DIGIT-TacTip		1.036 mm	0.851 mm	0.863 mm
TacTip		0.631 mm	1.418 mm	0.668 mm

shows that the image-to-image translation produces accurate tactile images for all sensors. Therefore, the method implemented from [2] is applicable not only for TacTip family of tactile sensors but also for GelSight-style sensors such as the DIGIT. Some examples of tactile images and their translated versions for edge contacts are shown in Fig. 2.

As an aside, we comment that the performance of the DIGIT image translation was slightly lower than for the TacTip and DIGIT-TacTip. It is possible that a different architecture of neural network more suited to the GelSight-type sensors could change this result, but for the time being we note the difference does not seem to be important later for the task performance.

C. RL Performance in the Physical Environments

1) *Object Pushing*: We consider first the object pushing task, where the tactile robot must push the object along a desired trajectory through a series of goal points. Successful task performance corresponds to accurately pushing the test object (a cube, cylinder or triangular prism) along the trajectory (straight, curved or sinusoidal), using feedback from the tactile sensor to maintain the object on its desired path.

In most cases, the tactile robot successfully pushed the object along its desired trajectory, with a typical mean Euclidean distance the actual trajectory from its intended trajectory of ~ 10 mm (Table III), compared to an overall distance travelled of 250 mm and a sensor tip size of 20-40 mm. The successful task performance is also indicated by snapshots taken from the trajectories (Fig. 5) and videos are provided in supplementary material. Overall, the performance when successfully pushing the objects is similar to that reported in [2].

Examining the results more closely, it seems that the TacTip performs slightly better than the DIGIT-TacTip with accuracies of 9-13 mm compared with 11-16 mm, which we attribute to a stabler contact due to the larger contact surface. The DIGIT has similar accuracy to the DIGIT-TacTip, but failed at the pushing task with the triangular prism for all trajectories. We think this is because the triangular prism is the lightest object and the DIGIT has a relatively stiff elastomer compared with

Table V

Mean Euclidean distances of the actual trajectories from the ground truth trajectories for the surface following task. The number in bold denotes the best result among the three sensors. Failure cases are indicated by "N/A".

Sensors	Obj.	Arch	Flower	Circle
DIGIT		N/A	N/A	0.466 mm
DIGIT-TacTip		0.790 mm	1.037 mm	0.579 mm
TacTip		0.911 mm	1.226 mm	0.591 mm

the TacTip and DIGIT-TacTip, which causes the tactile image translation to fail on this object.

2) *Edge Following*: Next, we consider the edge following task, where the tactile robot must slide the sensor around the edge of various planar objects with geometrical features such as curved edges and a corner. We note that the sim-to-real image translation was trained only on a straight edge, but as we see below the method generalized to more complex shapes.

In all cases, the tactile robot successfully completed the edge-following trajectory, with typical mean Euclidean position errors of 0.6-1.4 mm for the DIGIT-TacTip and TacTip, and 0.9-1.8 mm for the DIGIT (Table IV). The successful task performance is shown by the trajectories superimposed on the ground truth shapes (Fig. 7) and videos are provided in supplementary material. Again these results are comparable to those reported in [2] and also for servo control under supervised learning of the pose [33].

Examining these results more closely, it seems that the DIGIT can mostly traverse the edge contour accurately but has some regions where larger errors are introduced (coloured regions on top row of Fig. 7). In our view, this is likely because of the flat profile of the DIGIT sensing surface and that small deviations in height can occur around the object, which will have relatively less affect on the more compliant TacTip and DIGIT-TacTip sensors. Furthermore, there are higher frictional force due to the relatively more forceful contact of the DIGIT to deform sufficiently, which to avoid damaging the elastomer we coated the objects with wax to smooth the sliding motion.

3) *Surface Following*: Finally, we consider the surface following task, where the tactile sensor must slide the sensor around the curved surface of various objects with vertical walls, which have geometrical features such as concave and convex surfaces. We note that the sim-to-real image translation was trained only on a planar surface, but as we see below the method generalized to more complex shapes.

In most cases, the tactile robot successfully completed the surface-following trajectory, with typical mean Euclidean position errors of 0.6-1.2 mm for the DIGIT-TacTip and TacTip, and the DIGIT giving a more accurate trajectory with 0.5 mm

error on the circular wall. The successful task performance is shown by the trajectories superimposed on the ground truth shapes (Fig. 6) and videos are provided in supplementary material. Again these results are comparable to those reported in [2] and also for servo control under supervised learning of the pose [33].

During the experiments, we observed that the DIGIT tends to get stuck in the concave-shape surface (shown in the supplementary video). This is again because of the DIGIT's flat shape which hinders its sliding movement over concave surfaces. To avoid breaking the sensor, we decided not to conduct this task with DIGIT on the flower and the arch. On the object where the DIGIT could be used, the circular disc, the DIGIT achieved the best accuracy of all sensor.

V. DISCUSSION AND FUTURE WORK

In this paper, we developed a low-cost tactile robot platform for sim-to-real deep reinforcement learning based on Tactile Gym [2]. The hardware included a desktop robot (DOBOT MG400) and three low-cost high-resolution optical tactile sensors: the TacTip, DIGIT, and DIGIT-TacTip. We also integrated CAD models of the Dobot MG400 and the considered sensors into the Tactile Gym and successfully learned policies for all sensors in several physically-interactive tasks involving different contact dynamics. To train an effective GAN model for sim-to-real image translation, we fine-tuned the image preprocessing parameters and calibrated the internal camera of each simulated tactile sensor so that the distribution of the simulated dataset was well-matched with the real dataset.

The performance of our low-cost tactile sim-to-real deep RL robot platform was evaluated in three real-world physically-interactive tasks: edge-following, surface-following and object-pushing. The experimental results show that the developed platform is effective for all tasks with zero-shot performance on real objects or trajectories unseen in the simulation learning for all three tactile sensors. The main differences in performance between the sensors were due to the physical construction and material properties, rather than the different sensing mechanisms. For example, the flat profile of the DIGIT with a GelSight-type sensing surface made it unsuitable for following concave surfaces, unlike the soft domed structure of the TacTip and DIGIT-TacTip.

Such empirical studies should be helpful to other researchers to select and customize the appropriate physical characteristics of tactile sensors for different manipulation scenarios. Overall, we view the generality of our low-cost platform as opening up the possibility to apply either TacTip-style or GelSight-style tactile sensors to learning general sim-to-real deep RL policies for desired complex behaviors. The tactile robot platform should also benefit sim-to-real prehensile and dexterous manipulation, by enabling the fundamental methods to be developed in controlled scenarios before applying them to more challenging applications with dexterous robot hands.

Acknowledgements: We thank Mike Lambeta and Roberto Calandra for donating the DIGIT sensors. We thank Di Wu for her preliminary work on the Dobot hardware development.

REFERENCES

- [1] N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford *et al.*, “The limits and potentials of deep learning for robotics,” *The International journal of robotics research*, vol. 37, no. 4-5, pp. 405–420, 2018.
- [2] A. Church, J. Lloyd, R. Hadsell, and N. F. Lepora, “Optical tactile sim-to-real policy transfer via real-to-sim tactile image translation,” *arXiv preprint arXiv:2106.08796*, 2021.
- [3] S. Dong, D. K. Jha, D. Romeres, S. Kim, D. Nikovski, and A. Rodriguez, “Tactile-rl for insertion: Generalization to objects of unknown geometry,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6437–6443.
- [4] H. Xu, Y. Luo, S. Wang, T. Darrell, and R. Calandra, “Towards learning to play piano with dexterous hands and touch,” *arXiv preprint arXiv:2106.02040*, 2021.
- [5] S. Wang, M. M. Lambeta, P.-W. Chou, and R. Calandra, “Tacto: A fast, flexible, and open-source simulator for high-resolution vision-based tactile sensors,” *IEEE Robotics and Automation Letters*, 2022.
- [6] M. Lambeta, H. Xu, J. Xu, P.-W. Chou, S. Wang, T. Darrell, and R. Calandra, “Pytouch: A machine learning library for touch processing,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 208–13 214.
- [7] M. Lambeta, P.-W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer *et al.*, “Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation,” *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3838–3845, 2020.
- [8] B. Ward-Cherrier, N. Pestell, L. Cramphorn, B. Winstone, M. E. Giancaccini, J. Rossiter, and N. F. Lepora, “The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies,” *Soft robotics*, vol. 5, no. 2, pp. 216–227, 2018.
- [9] N. F. Lepora, “Soft biomimetic optical tactile sensing with the tactip: A review,” *IEEE Sensors Journal*, vol. 21, no. 19, pp. 21 131–21 143, 2021.
- [10] H. Van Hoof, N. Chen, M. Karl, P. van der Smagt, and J. Peters, “Stable reinforcement learning with autoencoders for tactile and visual data,” in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2016, pp. 3928–3934.
- [11] F. Veiga, D. Notz, T. Hesse, and J. Peters, “Tactile based forward modeling for contact location control,” in *RSS Workshop on Tactile Sensing for Manipulation*, 2017.
- [12] S. Tian, F. Ebert, D. Jayaraman, M. Mudigonda, C. Finn, R. Calandra, and S. Levine, “Manipulation by feel: Touch-based control with deep predictive models,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 818–824.
- [13] 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.
- [14] S. Kim and A. Rodriguez, “Active extrinsic contact sensing: Application to general peg-in-hole insertion,” *arXiv preprint arXiv:2110.03555*, 2021.
- [15] A. Church, J. Lloyd, R. Hadsell, and N. F. Lepora, “Deep reinforcement learning for tactile robotics: Learning to type on a braille keyboard,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6145–6152, 2020.
- [16] Y. S. Narang, B. Sundaralingam, K. Van Wyk, A. Mousavian, and D. Fox, “Interpreting and predicting tactile signals for the syntouch biotac,” *arXiv preprint arXiv:2101.05452*, 2021.
- [17] Y. Narang, B. Sundaralingam, M. Macklin, A. Mousavian, and D. Fox, “Sim-to-real for robotic tactile sensing via physics-based simulation and learned latent projections,” *arXiv preprint arXiv:2103.16747*, 2021.
- [18] C. Sferrazza, T. Bi, and R. D’Andrea, “Learning the sense of touch in simulation: a sim-to-real strategy for vision-based tactile sensing,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4389–4396.
- [19] C. Sferrazza and R. D’Andrea, “Sim-to-real for high-resolution optical tactile sensing: From images to 3d contact force distributions,” *arXiv preprint arXiv:2012.11295*, 2020.
- [20] T. Bi, C. Sferrazza, and R. D’Andrea, “Zero-shot sim-to-real transfer of tactile control policies for aggressive swing-up manipulation,” *IEEE Robotics and Automation Letters*, 2021.
- [21] Z. Ding, N. F. Lepora, and E. Johns, “Sim-to-Real Transfer for Optical Tactile Sensing,” *ICRA*, pp. 1639–1645, 2020.
- [22] D. F. Gomes, P. Paoletti, and S. Luo, “Generation of gelsight tactile images for sim2real learning,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 4177–4184, 2021.

- [23] A. C. Abad and A. Ranasinghe, "Visuotactile sensors with emphasis on gelsight sensor: A review," *IEEE Sensors Journal*, vol. 20, no. 14, pp. 7628–7638, 2020.
- [24] W. Yuan, S. Dong, and E. H. Adelson, "Gelsight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, no. 12, p. 2762, 2017.
- [25] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.
- [26] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.
- [28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.
- [29] J. Lloyd and N. F. Lepora, "Goal-driven robotic pushing using tactile and proprioceptive feedback," *IEEE Transactions on Robotics*, 2021.
- [30] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: <http://www.blender.org>
- [31] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
- [32] A. Raffin, A. Hill, M. Ernestus, A. Gleave, A. Kanervisto, and N. Dormann, "Stable baselines3," 2019.
- [33] N. F. Lepora and J. Lloyd, "Pose-based tactile servoing: Controlled soft touch using deep learning," *IEEE Robotics & Automation Magazine*, vol. 28, no. 4, pp. 43–55, 2021.