Tactile Gym 2.0: Sim-to-Real Deep Reinforcement Learning for Comparing Low-Cost High-Resolution Robot Touch

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

Abstract—**High-resolution optical tactile sensors are increasingly used in robotic learning environments due to their ability to capture large amounts of data directly relating to agent-environment interaction. 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 sim-to-real methods that lack generality across different sensors. In this letter we extend the Tactile Gym simulator to include three new optical tactile sensors (TacTip, DIGIT and DigiTac) of the two most popular types, Gelsight-style (image-shading based) and TacTip-style (marker based). We demonstrate that a single sim-to-real approach can be used with these three different sensors to achieve strong real-world performance despite the significant differences between real tactile images. Additionally, we lower the barrier of entry to the proposed tasks by adapting them to an inexpensive 4-DoF robot arm, further enabling the dissemination of this benchmark. We validate the extended environment on three physically-interactive tasks requiring a sense of touch: 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 [2]. 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. [1], [3], [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 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 [1]. 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 [1] 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 the DIGIT sensor (of Gelsight type), a reduced form-factor TacTip sensor and a DIGIT-TacTip hybrid sensor referred to as the DigiTac. We demonstrate strong real-world performance across these sensors despite the significantly different tactile images produced by each.

The main contributions of this work are as follows:

1) We extend the Tactile Gym [1] from one to three low-cost, high-resolution optical tactile sensors. To the best of

Manuscript received 24 February 2022; accepted 18 July 2022. Date of publication 1 August 2022; date of current version 9 August 2022. This letter was recommended for publication by Associate Editor R. Liu and Editor A. Banerjee upon evaluation of the reviewers' comments. The work of Yijiong Lin was supported in part by the China Scholarship Council (CSC)/University of Bristol joint-funded scholarship. The work of Alex Church was supported in part by an EPSRC CASE award sponsored by Google DeepMind. The work of Nathan F. Lepora and John Lloyd was supported in part by a Leadership Award from the Leverhulme Trust on 'A biomimetic forebrain for robot touch' under Grant RL-2016-39. *(Corresponding author: Nathan F. Lepora.)*

The authors are with the Department of Engineering Mathematics and Bristol Robotics Laboratory, University of Bristol, BS8 1UB Bristol, U.K. (e-mail: th20988@bristol.ac.uk; jl15313@bristol.ac.uk; ac14293@bristol.ac.uk; n.lepora@bristol.ac.uk).

Code and videos are available at https://github.com/ac-93/tactile_gym and https://sites.google.com/my.bristol.ac.uk/tactilegym2.

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2022.3195195, provided by the authors.

Digital Object Identifier 10.1109/LRA.2022.3195195

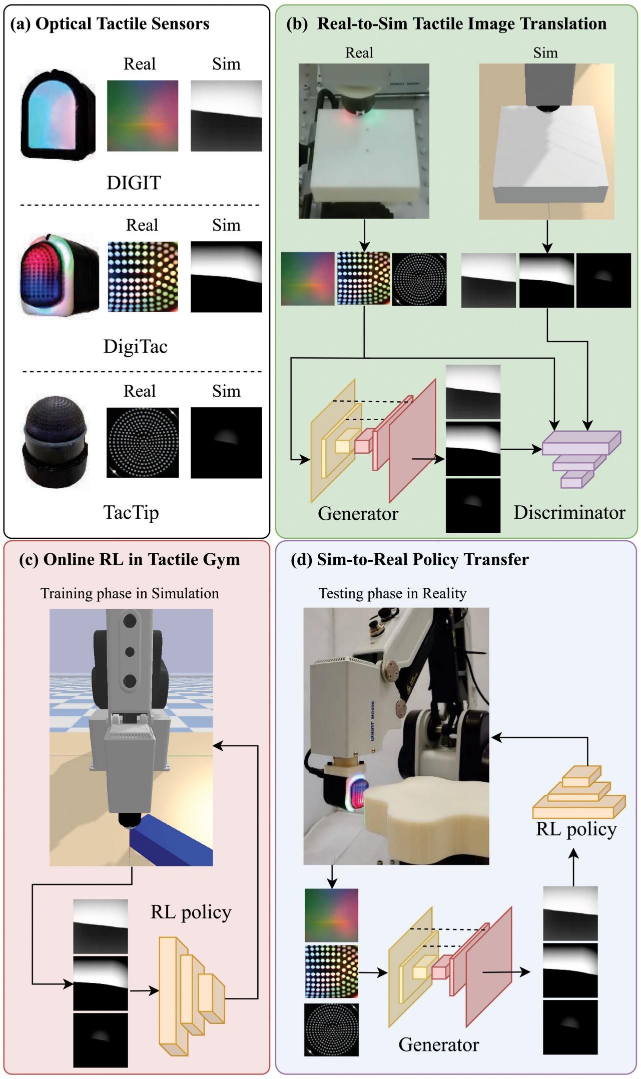

(a) Optical Tactile Sensors
Real
Sim
DIGIT
Real
Sim
DigiTac
Real
Sim
TacTip
(b) Real-to-Sim Tactile Image Translation
Real
Sim
Generator
Discriminator
(c) Online RL in Tactile Gym
Training phase in Simulation
RL policy
(d) Sim-to-Real Policy Transfer
Testing phase in Reality
RL policy
Generator

Fig. 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 [1] with simulated tactile images from multiple integrated sensors. (d) A desktop robot equipped with the DigiTac performing the surface-following task by translating real tactile images to simulated images for the RL policy.

our knowledge, this is the first work that integrates two widely-used yet fundamentally different styles of the optical tactile sensors: Gelsight-style (DIGIT) and TacTip-style (DigiTac and TacTip), into one platform. Such an integration validates and extends earlier results for this platform while making it more accessible and applicable to the wider research community.

2) With the extended Tactile Gym environments, we successfully learn deep RL policies for three physically-interactive tasks (edge following, surface following and object pushing), and transfer them into the real world without further tuning. By comparing task performance, we identify strengths and weaknesses of the tactile sensors in these different scenarios. As far as we know, this is the first empirical comparison of optical tactile sensors in a sim-to-real context, which we intend as a benchmark to aid improvement of this technology.

3) We improve the accessibility of this research by adapting the tasks to the reduced workspace and degrees of freedom of the DOBOT MG400, an inexpensive and commercially-available 4-DoF robot arm. All software and hardware developments in this work are openly released to aid dissemination of this benchmark (see www.sites.google.com/my.bristol.ac.uk/tactilegym2).

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 [2]. 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 [1], 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 and learning to play the piano [4] with the DIGIT optical tactile sensor.

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 [1], [5], [22], [23]. For a more thorough review we refer to [1]. In the present work, we follow the tactile sim-to-real method described in [1] by using depth image-rendering and image translation for two popular optical tactile sensor classes: GelSight-type which is based on image shading [24] and TacTip type which is based on biomimetic marker-based transduction [9].

III. Method and Experiments

A. Tactile Robot System

In this letter, 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 by Church et al. [1] for investigating sim-to-real tactile deep RL, which used a 6-axis industrial robot

TABLE I
Comparison of the Existing Tactile Sim-to-Real Robotic Platforms for TacTip-Style (T) and GelSight-Style (G) Sensors, Respectively

Platform	Features	[1]	[21]	[23]	Ours
Robots	Type	UR5	Sawyer	UR5	MG400
	Accuracy (mm)	0.1	0.1	0.1	**0.05**
	Price ($)	45k	26k	45k	**2.7k**
	Max. Reach (mm)	850	**1260**	850	440
	Playload (kg)	**5**	4	**5**	0.75
	Weight (kg)	18.4	19	18.4	**8**
	DoF	6	**7**	6	4
Sensors (if integrated)	TacTip (T)	✓	✓	✗	✓
	DIGIT (G)	✗	✗	✓	✓
	DigiTac (T)	✗	✗	✗	✓

from Universal Robotics. In this letter, we expand the approach to compare three distinct optical tactile sensors: the TacTip, DIGIT, and DigiTac. Table I compares this robot platform to those used in related works, in terms of the features, cost and the integration of tactile sensors. The robot platform and the operation of the tactile sensors are presented in detail in an accompanying second letter, with the main focus of the present letter 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 $\times$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 [25], but can be fabricated at low cost and is of a size suitable for integration of some robotic hands, such as on the fingertips of the Allegro hand [7].

(c) The **DigiTac** 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. In other words, this sensor outputs tactile images of the same dimension 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. [1] 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 [1].

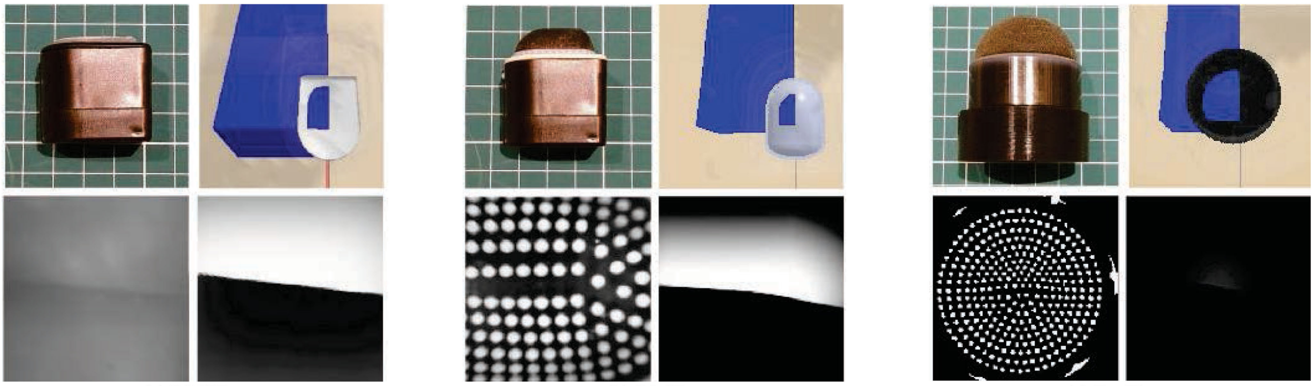
To adapt the existing approach [1] to our setting, we needed to make several changes. In this work we chose to use a low-cost desktop robot arm as described above (Section III A). This arm has only 4 actuated axes, whereas previous work used a 6-axis industrial robotic arm (UR5, Universal Robotics). 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 [1] 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 and flat TacTip. Here we validate that the approach works with the DIGIT sensor of the Gelsight-type, which has a distinct tactile sensing principle based on image shading in place of marker-based transduction. We also validate the approach with alternative forms of the TacTip sensor, including a new (more compact) version of TacTip than used in [1] and the DigiTac which has a TacTip skin on a DIGIT housing. When compared with the original work these sensors introduce differences in camera perspectives, lens dynamics, marker size, density of markers and have significantly different lighting conditions. To do this, we extended the simulation to include simulated sensors that matched real hardware using the CAD files for 3D-printing those sensors.

C. Deep RL With Different Tactile Dynamics in Simulation

The sim-to-real deep RL framework uses the Tactile Gym [1] (see above) to simulate the contact dynamics with rigid-body physics, using tactile information rendered as depth images relative to the CAD model for the sensor. Thus, the learnt tactile-feedback policies from [1] apply only the TacTip originally considered in that study.

Hence, we extend the Tactile Gym [1] with two new virtual optical tactile sensors: DIGIT (Gelsight-style) and DigiTac (TacTip-style) (Fig. 2 a,b), based on their open-source CAD files [7] used for 3D-printing the DIGIT. We follow the method described in [1] to efficiently capture the depth images as tactile



(a) DIGIT (b) DigiTac (c) TacTip

Fig. 2. Comparison between the different optical tactile sensors: (a) DIGIT, (b) DigiTac, (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.

images by synthetic cameras embedded within those sensors. Specifically, we adjust the physics (friction, stiffness, damping, etc.) of the skin and the parameters of the camera for each sensor, to achieve realistic performance during learning.

The three distinct tactile observation spaces corresponding to the three type of sensor (Fig. 2) are then used as input to train deep RL polices for the tasks, following the methods detailed in [1]. We report on the training results in Section 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 or tuning. Hence it is essential to have a model to bridge the gap between the simulated and real domains. Progress in Generative Adversarial Network (GAN) methods has enabled realistic image generation, which we leverage to learn an image-to-image translation GAN [26] applied to real-to-sim tactile image translation. The network takes advantage of a U-net architecture [27] for the image-conditioned generator to infer better the spatial features during training, alongside a standard convolutional network [28] with batch normalization [29] for the discriminator.

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

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. We collect a training and validation dataset per contact feature (edge and surface) and per sensor, leading to twelve datasets in total.

TABLE II

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)

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

Each training dataset comprised 5000 tactile images and each validation dataset 2000 tactile images, collected by using the desktop tactile robot (Section III A) to randomly contact data of the appropriate edge or surface feature. For the data collection, straight edge and flat surface of a 3D-printed stimulus, labeled with the relative poses between the sensor and the stimulus under contact. The movement ranges for each random contact feature are shown in Table II. 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 DigiTac 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 scaling the y-range orthogonal to the edge by the tangent of the angle.

F. Tactile Control Tasks

Here we adapt three tactile control tasks proposed in [1] to the desktop tactile robot: edge-following, surface-following and object-pushing. Although we expect our platform would also be viable for the ball-rolling task, we do not implement it in this work because the DigiTac does not currently have a flat 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

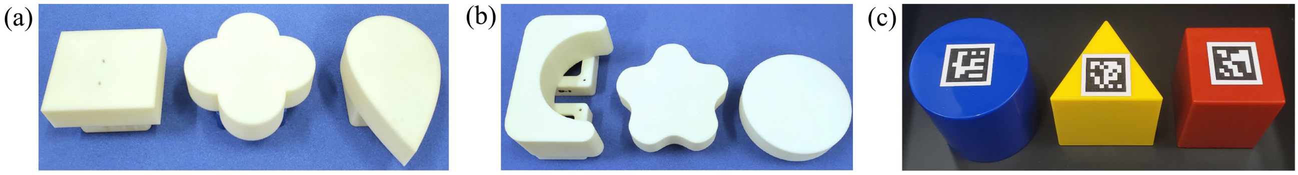


Fig. 3. Tactile stimuli used in manipulation tasks: (a) square, clover leaf, and teardrop stimuli for the edge following task; (b) arch, flower, and disc for the (side) surface following task; (c) blue cylinder, yellow triangular prism, and red cube for the object pushing task (ArUco markers for ground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

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 2D action space comprises 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 c): a triangular prism, cube and cylinder varying in weight from 185 g to 363 g. These objects differ from those used in [1], which were lighter of mass (50 g to 80 g), to better suit the elasticity of the DIGIT that required a heavier object for an appreciable tactile deformation. An ArUco marker is place on top of each object to track its movement for comparison with the ground truth using the tracking method described in [30].

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, positively/negatively curved edges and a right-angled corner (Fig. 3(a)). To measure tracking performance, the ground truth shapes are extracted from the CAD models of these 3D-printed objects by importing into the Blender CAD software [31] and outputting the point clouds of their boundaries.

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. 3(b)) 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 using Blender.

IV. Results

A. RL Performance in the Simulated Environments

The deep RL method PPO [32] 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 [33] implementation of PPO for training the policies.

TABLE III
Mean SSIM Values for Image Translation GANs Trained With Edge and Surface Features (Values Closer to 1.0 Represent Better Image Translation)

Sensors / Features	Tactip	DigiTac	DIGIT
Edge	0.9956	0.9953	0.9867
Surface	0.9927	0.9932	0.9818

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

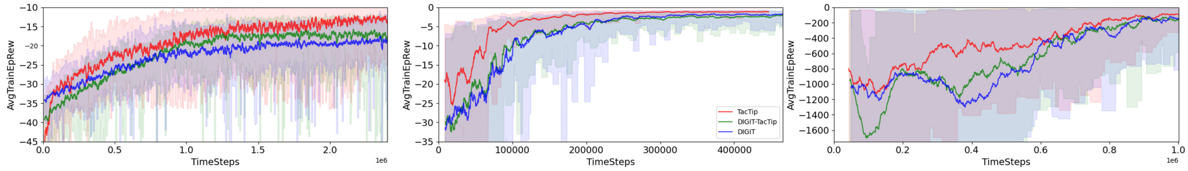
B. Image Translation for Sim-to-Real Transfer

We use the Structural Similarity Index (SSIM) to evaluate the quality of the translated images for both the edge and surface validation datasets (Table III). The high SSIM values close to unity (perfect match) obtained with the trained model shows that the image-to-image translation produces accurate tactile images for all sensors. Therefore, the method implemented from [1] 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 the TacTip and DigiTac. It is possible that a different architecture of neural network more suited to the GelSight-type sensors could change this result. We note that this minor difference does not affect the task overall performance, which we attribute to the deep RL policies being learnt with domain randomization.

C. RL Performance in the Physical Environments

1) Object Pushing: We consider first the object pushing task, where the tactile robot must move 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 trajectories (straight: $y = kx$, curved: $y = 0.001x^2$, sinusoidal:



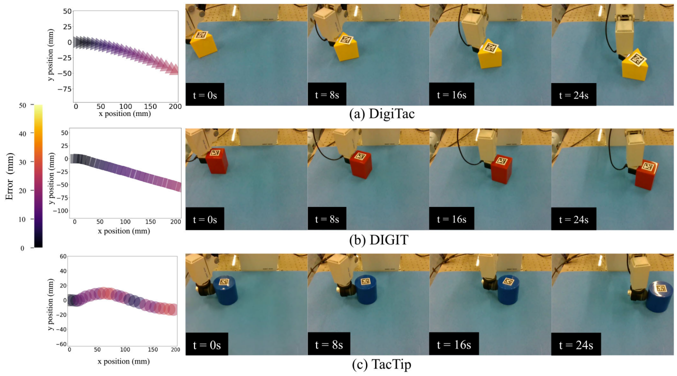
(a) Edge-following (b) Surface-following (c) Object-pushing

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

TABLE IV

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". All the Experiments are Using the Objects Without Additional Weights Except the Final Row "DIGIT \ (weighted)" Where We Increase All the Objects' Weights by 150 g

Trajectories	Straight			Curve			Sine		
Obj. Sensors	Cube	Cylinder	Tri. Prism	Cube	Cylinder	Tri. Prism	Cube	Cylinder	Tri. Prism
Tactip	**10.33 mm**	**9.21 mm**	**11.51 mm**	**12.19 mm**	11.29 mm	**13.20 mm**	**11.93 mm**	**11.30 mm**	**13.89 mm**
DigiTac	11.25 mm	10.24 mm	16.09 mm	13.01 mm	**11.20 mm**	16.41 mm	12.32 mm	11.48 mm	15.13 mm
DIGIT	11.20 mm	10.13 mm	N/A	12.94 mm	12.00 mm	N/A	12.41 mm	11.33 mm	N/A
DIGIT \ (weighted)	10.92 mm	11.00 mm	16.65 mm	12.28 mm	11.51 mm	16.58 mm	12.07 mm	11.53 mm	17.06 mm



(a) DigiTac

(b) DIGIT

(c) TacTip

Fig. 5. The tactile robot executing 3 pushing policies: (a) triangular prism along a curve trajectory with the DigiTac, (b) cube along a straight line with the DIGIT, (c) cylinder along a sinusoidal trajectory with the TacTip. The plots in the most left columns are the objects actual paths recorded by the tracking system; the right four columns are snapshots taken from the camera.

$y = 0.02\sin(0.02x)$, where $k \in [-0.3, 0.3], x \in [0, 200]$ mm), using feedback from the tactile sensor to maintain the object on its desired path.

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 IV), compared to an overall distance travelled of 200–280 mm (250 steps in total for each episode) 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 [1].

Examining the results more closely, the TacTip performs slightly better than the DigiTac with accuracies of 9-13 mm compared with 11-16 mm, which we attribute to a stabler push due to the larger contact surface. The DIGIT has similar accuracy to the DigiTac, but failed at the pushing task with the triangular prism for all trajectories. We hypothesise that the main reason of this failure is that the triangular prism is the lightest object (185 g) and the DIGIT has a relatively stiff elastomer compared with the TacTip and DigiTac, which causes the tactile image translation to fail on this object. To validate this hypothesis, we extend the experiments with the DIGIT to pushing the triangle prism with additional weights (ranging from 50–200 g) using the same deep RL and GAN models. The results (Table V) show that the performance improves with increased object weight upto 150 g after which there was no benefit. We expect this is because the additional weight helps the tactile images lie within the distribution of the GAN training for accurate real-to-sim transfer. Once in this range, the bottleneck on the performance is the RL policy instead of the GAN.

TABLE V

Mean Euclidean Distances of the Trajectories From the Ground Truth for the Specific Object (Triangle Prism) Pushing Task Using DIGIT. The Number in Bold Denotes the Best Result Among the Weights

Traj. Weights	Straight	Curve	Sine
185g + 0g	N/A	N/A	N/A
185g + 50g	18.05 mm	17.73 mm	17.95 mm
185g + 100g	16.93 mm	17.54 mm	18.12 mm
185g + 150g	**16.65 mm**	**16.58 mm**	**17.06 mm**
185g + 200g	17.17 mm	16.92 mm	17.26 mm

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 generalizes 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 DigiTac and TacTip, and 0.9-1.8 mm for the DIGIT (Table VI). The successful task performance is shown by the trajectories superimposed on the ground truth shapes (Fig. 6)

TABLE VI
MEAN EUCLIDEAN DISTANCES OF THE TRAJECTORIES FROM THE GROUND TRUTH FOR THE EDGE FOLLOWING TASK. THE BEST RESULT FROM THE THREE SENSORS IS SHOWN IN BOLD

Obj. / Sensors	Square	Clover	Foil
DIGIT	0.88 mm	1.71 mm	1.82 mm
DigiTac	1.04 mm	**0.85 mm**	0.86 mm
TacTip	**0.63 mm**	1.42 mm	**0.67 mm**

TABLE VII
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 FROM THE THREE SENSORS. FAILURE CASES ARE INDICATED BY "N/A"

Obj. / Sensors	Arch	Flower	Circle
DIGIT	N/A	N/A	**0.47 mm**
DigiTac	**0.79 mm**	**1.04 mm**	0.58 mm
TacTip	0.91 mm	1.23 mm	0.59 mm

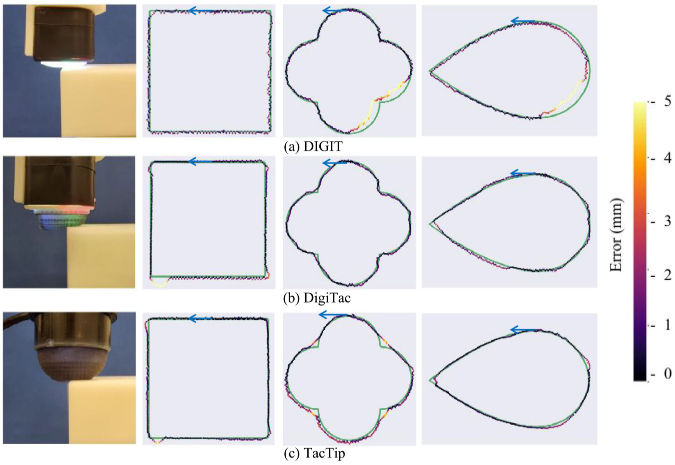

(a) DIGIT
(b) DigiTac
(c) TacTip

Fig. 6. The tactile robot executing edge-following policies on 3 distinct shapes for the (a) DIGIT, (b) DigiTac and (c) TacTip. 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 blue arrow denotes the starting point and direction.

and videos are provided in supplementary material. Again these results are comparable to those reported in [1] and also for servo control under supervised learning of the pose [34].

Examining the results more closely, the DIGIT can achieve accurate performance (mean Euclidean position errors of 1.5 mm overall) when traversing the edge contours despite having some regions of larger error (Fig. 6 top row, coloured regions up to 5 mm error). This is because the flatter stiffer elastomer of the DIGIT sensing surface causes an increased sensitivity to small deviations in penetration distance while moving around the object, relative to the softer TacTip and DigiTac sensing surfaces. For sufficient deformation of the DIGIT sensor to give good performance, a more forceful contact needs to be applied, which also increases the frictional force. To avoid damaging the sensor, we mitigate this friction by coating the objects with wax to ease the sliding motion.

3) Surface Following: Finally, we consider the surface following task, where the tactile sensor must slide around the curved surfaces 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 generalizes to more complex shapes.

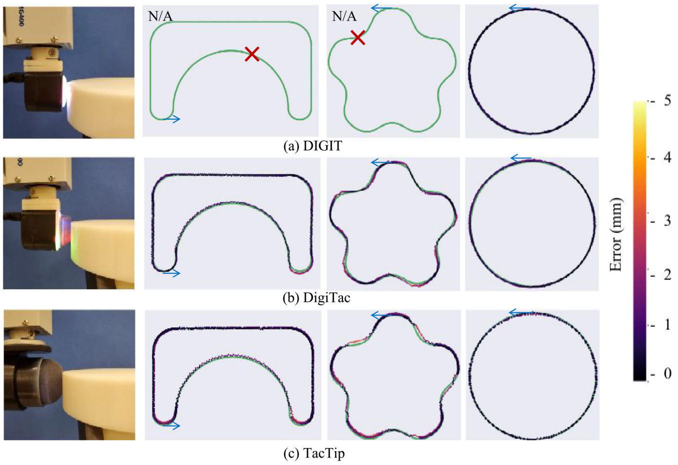

(a) DIGIT
(b) DigiTac
(c) TacTip

Fig. 7. The tactile robot executing 3 surface-following policies on 3 distinct shapes, corresponding to the (a) DIGIT, (b) DigiTac and (c) TacTip. 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 blue arrow denotes the starting point and direction. The DIGIT failed to follow the arch and flower objects at points denoted by the red crosses.

The tactile robot successfully completed the surface-following trajectory, with typical mean Euclidean position errors of 0.6-1.2 mm for the DigiTac and TacTip, and the DIGIT giving the most accurate trajectory of 0.5 mm error on the circular the quantified results are given in Table VII. 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 [1] and also for servo control under supervised learning of the pose [34].

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 flatter, stiffer sensing surface 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.

V. DISCUSSION AND FUTURE WORK

In this letter, we developed a low-cost tactile robot platform for sim-to-real deep reinforcement learning based on Tactile Gym [1]. The hardware included a desktop robot (DOBOT MG400) and three low-cost high-resolution optical tactile sensors: the TacTip, DIGIT, and DigiTac. 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 real-to-sim 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 flatter, stiffer construction the DIGIT with a GelSight-type sensing surface made it unsuited for following concave surfaces, unlike the soft domed structure of the TacTip and DigiTac.

Such empirical studies should help other researchers 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 tasks, for example by enabling the fundamental methods to be developed in controlled scenarios before applying them to more challenging applications with dexterous robot hands.

Acknowledgment

The authors thank Mike Lambeta and Roberto Calandra for donating the DIGIT sensors. They thank Di Wu for her preliminary work on the DOBOT hardware development. They thank Andrew Stinchcombe for helping with the 3D-printing of the stimuli.

References

[1] A. Church, J. Lloyd, R. Hadsell, and N. Lepora, "Tactile sim-to-real policy transfer via real-to-sim image translation," in *Proc. 5th Conf. Robot Learn.*, 2022, pp. 1645–1654.

[2] N. Sünderhauf et al., *The Limits and Potentials of Deep Learning for Robotics*, vol. 37. London, U.K.: SAGE, Apr. 2018, pp. 405–420.

[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 *Proc. IEEE Int. Conf. Robot. Automat.*, 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," Jun. 2021, *arXiv:2106.02040*. [Online]. Available: https://arxiv.org/abs/2106.02040

[5] S. Wang, M. Lambeta, P. Chou, and R. Calandra, "TACTO: A fast, flexible, and open-source simulator for high-resolution vision-based tactile sensors," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 3930–3937, Apr. 2022.

[6] M. Lambeta et al., "PyTouch: A machine learning library for touch processing," in *Proc. IEEE Int. Conf. Robot. Automat.*, Xi'an, China, 2021, pp. 13208–13214.

[7] M. Lambeta et al., "DIGIT: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation," *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 3838–3845, Jul. 2020.

[8] B. Ward-Cherrier et al., "The TacTip family: Soft optical tactile sensors with 3D-printed biomimetic morphologies," *Soft Robot.*, vol. 5, no. 2, pp. 216–227, Apr. 2018.

[9] N. Lepora, "Soft biomimetic optical tactile sensing with the TacTip: A review," *IEEE Sensors J.*, vol. 21, no. 19, pp. 21131–21143, Oct. 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 3928–3934.

[11] F. Veiga, D. Notz, T. Hesse, and J. Peters, "Tactile based forward modeling for contact location control," in *Proc. RSS Workshop Tactile Sens. Manipulation*, 2017, pp. 1–4.

[12] S. Tian et al., "Manipulation by feel: Touch-based control with deep predictive models," in *Proc. IEEE Int. Conf. Robot. Automat.*, Montreal, QC, Canada, 2019, pp. 818–824.

[13] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80 of Proceedings of Machine Learning Research, 2018, pp. 1587–1596.

[14] S. Kim and A. Rodriguez, "Active extrinsic contact sensing: Application to general peg-in-hole insertion," in *Proc. IEEE Int. Conf. Robot. Automat.*, Philadelphia, PA, USA, 2022, pp. 10241–10247.

[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 Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6145–6152, Oct. 2020.

[16] Y. S. Narang, B. Sundaralingam, K. Van Wyk, A. Mousavian, and D. Fox, "Interpreting and predicting tactile signals for the syntouch biotac," *Int. J. Robot. Res.*, vol. 40, no. 12–14, pp. 1467–1487, Dec. 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," in *Proc. IEEE Int. Conf. Robot. Automat.*, Xi'an, China, 2021, pp. 6444–6451.

[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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Las Vegas, NV, USA, 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," *Soft Robot.*, ahead-of-print, Nov. 2021. [Online]. Available: https://www.liebertpub.com/doi/full/10.1089/soro.2020.0213

[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 Robot. Automat. Lett.*, vol. 6, no. 3, pp. 5761–5768, Jul. 2021.

[21] Z. Ding, N. F. Lepora, and E. Johns, "Sim-to-real transfer for optical tactile sensing," in *Proc. IEEE Int. Conf. Robot. Automat.*, Paris, France, 2020, pp. 1639–1645.

[22] P. Paoletti, D. F. Gomes, and S. Luo, "Generation of gelsight tactile images for sim2real learning," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 4177–4184, Apr. 2021.

[23] Z. Si and W. Yuan, "Taxim: An example-based simulation model for gelsight tactile sensors," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 2361–2368, Apr. 2022.

[24] A. C. Abad and A. Ranasinghe, "Visuotactile sensors with emphasis on gelsight sensor: A review," *IEEE Sensors J.*, vol. 20, no. 14, pp. 7628–7638, Jul. 2020.

[25] W. Yuan, S. Dong, and E. H. Adelson, "Gelsight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, no. 12, Nov. 2017, Art. no. 2762.

[26] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.

[27] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2015, pp. 234–241.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 25, pp. 1097–1105, 2012.

[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, Lille, France, 2015, vol. 37, pp. 448–456.

[30] J. Lloyd and N. Lepora, "Goal-driven robotic pushing using tactile and proprioceptive feedback," *IEEE Trans. Robot.*, vol. 38, no. 2, pp. 1201–1212, Apr. 2022.

[31] Blender Online Community, *Blender - a 3D Modelling and Rendering Package*. Amsterdam: Blender Foundation, Stichting Blender Foundation, 2018.

[32] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*. [Online]. Available: http://arxiv.org/abs/1707.06347

[33] A. Raffin, A. Hill, M. Ernestus, A. Gleave, A. Kanervisto, and N. Dormann, "Stable baselines3," 2019.

[34] N. Lepora and J. Lloyd, "Pose-based tactile servoing: Controlled soft touch using deep learning," *IEEE Robot. Automat. Mag.*, vol. 28, no. 4, pp. 43–55, Dec. 2021.