Robotics and Autonomous Systems 160 (2023) 104321

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Skill generalization of tubular object manipulation with tactile sensing and Sim2Real learning

Yongqiang Zhao [a,b,1], Xingshuo Jing [a,b,1], Kun Qian [a,b,*], Daniel Fernandes Gomes [c], Shan Luo [d,**]

[a] School of Automation, Southeast University, Nanjing 210096, China
[b] The Key Laboratory of Measurement and Control of CSE, Ministry of Education, Southeast University, Nanjing 210096, China
[c] Department of Computer Science, University of Liverpool, Liverpool, L69 3BX, United Kingdom
[d] Department of Engineering, King's College London, London, WC2R 2LS, United Kingdom

ARTICLE INFO

Article history:
Available online 1 December 2022

Keywords:
Skill generalization
Sim-to-real learning
Tactile sensing
Reinforcement learning
Unsupervised domain adaptation
Robot manipulation

ABSTRACT

Tubular objects such as test tubes are common in chemistry and life sciences research laboratories, and robots that can handle them have the potential to accelerate experiments. Moreover, it is expected to train a robot to manipulate tubular objects in a simulator and then deploy it in a real-world environment. However, it is still challenging for a robot to learn to handle tubular objects through single sensing and bridge the gap between simulation and reality. In this paper, we propose a novel tactile–motor policy learning method to generalize tubular object manipulation skills from simulation to reality. In particular, we propose a Sim-to-Real transferable in-hand pose estimation network that generalizes to unseen tubular objects. The network utilizes a novel adversarial domain adaptation network to narrow the pixel-level domain gap for tactile tasks by introducing the attention mechanism and a task-related constraint. The in-hand pose estimation network is further implemented in a Reinforcement Learning-based policy learning framework for robotic insert-and-pullout manipulation tasks. The proposed method is applied to a human–robot collaborative tube placing scenario and a robotic pipetting scenario. The experimental results demonstrate the generalization capability of the learned tactile–motor policy toward tubular object manipulation in research laboratories.

1. Introduction

Robotic chemists [1] have been applied to accelerating the investigation of catalyst formulations, as what they do in one week is more than a student could typically study in 4 years. The capability of manipulating tubular objects is highly expected for robotic chemists that operate in a conventional, unmodified laboratory. For these robots, it is imperative that the learned contact-rich manipulation skills generalize well to environmental changes, task variations, as well as different tubular objects to be manipulated. To improve the task's generalization ability, data-driven skill learning is more desirable than a hand-engineered control paradigm.

In this regard, one challenge is the perception of tubular objects' in-hand poses in the skill learning. There are abundant contact-rich manipulations of different and sometimes transparent objects in chemical experiments, such as placing the test tubes in the reagent racks, and manipulating the droppers for reagent transferring, as shown in Fig. 1. It is challenging to detect these in-hand transparent objects using vision only, not only due to heavy occlusions but also because of the reflection and lack of salient features in transparent objects [2]. In contrast, in-hand tactile sensing is appropriate to compensate for the limitation of vision. In particular, optical tactile sensors such as GelSight [3] provide high-resolution tactile images and are extensively utilized in tactile-guided tasks.

In addition, positional uncertainty in robotic grasping as well as variations in tubular objects being manipulated indicates unseen in-contact surfaces. As a particular example, a robot chemist that collaborates with human chemists may receive different tubular objects from its human counterpart and is required to place them into a rack. Due to the inaccurate human handover motion [4], it is difficult for the robot to ensure a fixed grasp position, resulting in the possibility of different parts of an object being held, and sometimes the in-contact textures are even unseen. Therefore, the tactile–motor policy in this context should generalize well toward the unseen surface texture of objects.

* Corresponding author at: School of Automation, Southeast University, Nanjing 210096, China.
** Corresponding author.
E-mail addresses: kqian@seu.edu.cn (K. Qian), shan.luo@kcl.ac.uk (S. Luo).

[1] Yongqiang Zhao and Xingshuo Jing contributed equally to this work.

https://doi.org/10.1016/j.robot.2022.104321
0921-8890/

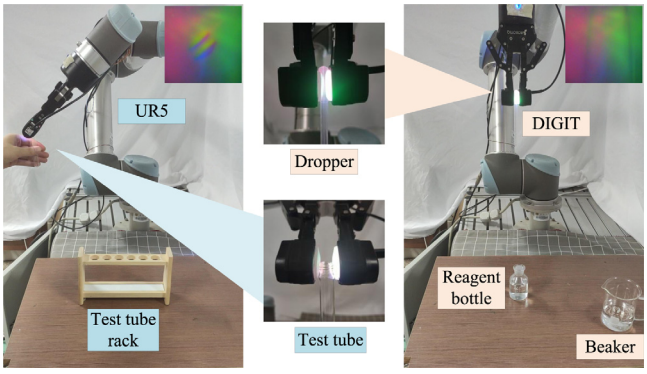

Fig. 1. Tactile-guided human–robot collaborative tube placing (left) and robotic pipetting (right). There are a UR5 robot and a Robotiq 2F-85 gripper equipped with two DIGIT sensors in both fingers.

Similar to visuomotor policies, RL-based tactile–motor policies call for end-to-end in-hand pose estimation methods with high generalization capability.

Another concern is learning the tactile–motor policy via sim-to-real transfer. The soft elastomer of optical tactile sensors is brittle and subject to wear-and-tear during robotic explorations. In addition, large-scale real-world data with annotation are costly for robot manipulation learning [5]. Therefore, transferring robot manipulation skills learned in simulation to reality is an effective way to reduce training costs [6]. When the real-world scenarios and manipulated objects are unseen in the simulator, zero-shot cross-domain policy transfer is expected to reproduce the robotic manipulation skills in the real world without further training in the real-world domain. Although tactile simulators [7–9] have been proposed to produce high-resolution tactile images, the simulation models are unable to capture the imperfect details (e.g., shadows, scratches) and object deformation in realistic tactile images for tubular objects. Because of the inherent domain gap, the generalization performance of the model trained by synthetic tactile images will be significantly reduced when deployed in the real-world environment.

Recent advances in unsupervised domain adaptation have exhibited the capability of sim-to-real perceptual transfer in visual or tactile based robotic tasks [10–12]. By the adversarial training with paired [10] or unpaired [11,12] simulated and real-world data between generator and discriminator, the probability distribution between the simulation domain and the real-world domain is learned autonomously. However, for those optical tactile sensors (e.g., the DIGIT sensor) with hard silicone elastomer, their tactile images are less sensitive to the deformation of the object caused by pressing. This means that pressing with moderate force will result in less distinctive features in the tactile images. In this case, the generation network based solely on Convolutional Neural Networks (CNNs) may not be competent for perceiving the texture features. To handle this problem, applying the attention mechanism and a task-related constraint can focus on the in-contact regions in the tactile images, enhance the global perception ability of the network, and thus transfer tactile images with higher quality.

In this paper, we consider different robotic grasp positions on the tubular objects, which will produce seen or unseen in-contact textures. At the same time, sim-to-real gaps lead to difficulty for transferring manipulation policy from simulation to reality. To this end, we propose a method for generalizing to unseen in-contact textures and bridging sim-to-real gaps. In particular, we propose CTF-CycleGAN (CNNs and Transformer Fusion-CycleGAN), which is trained with unpaired data collected from simulation and reality. We optimize the design of network modules by introducing the attention mechanism and constructing a task-related constraint to reduce the domain gap of tactile images in estimating the in-hand pose of tubular objects. This ensures higher reliability in manipulating unseen tubular objects when the learned tactile–motor policy is transferred to a real-world setting.

In summary, the contributions of this work are three-fold:

- A novel tactile–motor policy framework is proposed for learning robotic skills of tubular object manipulation that are generalizable toward Sim2Real transfer and different tubular objects. This is achieved via decoupling reinforcement learning-based skills and pixel-level tactile image transfer, which is also a general paradigm extendable to a variety of tactile–motor policy learning tasks.
- We propose a Sim2Real in-hand pose estimation method for tubular objects using tactile images. The key of the method is CTF-CycleGAN, a novel pixel-level unsupervised domain adaptation network that cascades CNNs and Transformer. We introduce the attention mechanism and a novel task-related constraint to ensure high-quality tactile image transfer. The proposed method ensures high generalization capability of in-hand pose estimation toward unseen surface textures of tubular objects.
- The proposed approach is implemented for learning Sim2Real transferable robotic insert-and-pullout actions, which are further evaluated in a human–robot collaborative tube placing scenario and a robotic pipetting scenario. Experiment results demonstrate the effectiveness of our approach in dealing with domain gaps as well as unseen in-contact textures.

The rest of this paper is organized as follows: the related work is reviewed in Section 2 and the proposed methods are introduced in Section 3. Section 4 illustrates the experimental setup and data preparation, followed by a description of the experimental results in Section 5, and Section 6 gives the conclusion.

2. Related work

2.1. Tactile–motor policy

Tactile provides the information to determine the states and properties of the manipulated objects, such as shape, pose, texture, material [13–18], etc. Compared with two conventional tactile sensors, the single-point contact sensor and the tactile array, the optical tactile sensor is able to present high-resolution tactile images. Typical optical tactile sensors like GelSight, GelTip, TacTip, and DIGIT [3,19–21], have been used for multiple robotics tasks, for example, sliding detection, texture recognition, surface tracking, object pushing, cable operation, plugging and screwing [10,22–26], etc.

In the works of [23,24], the tactile texture is used to extract uniform hand-engineered features, but the manipulated object with a single type and simple texture results in poor generalization ability of policies. In numerous experiments, it is difficult to extract hand-engineered features since the manipulated object's size, shape, and texture details are varied. The end-to-end tactile–motor policy learning methods [10,25,26] directly utilize tactile images as input, learn the features present in the tactile images via convolutional neural networks, recursive neural networks, or Transformer, etc., then maps the features to robot manipulation policies. The aforementioned methods improve the generalization ability of the tactile–motor policies for various manipulated objects via the gradient coherence of the neural network.

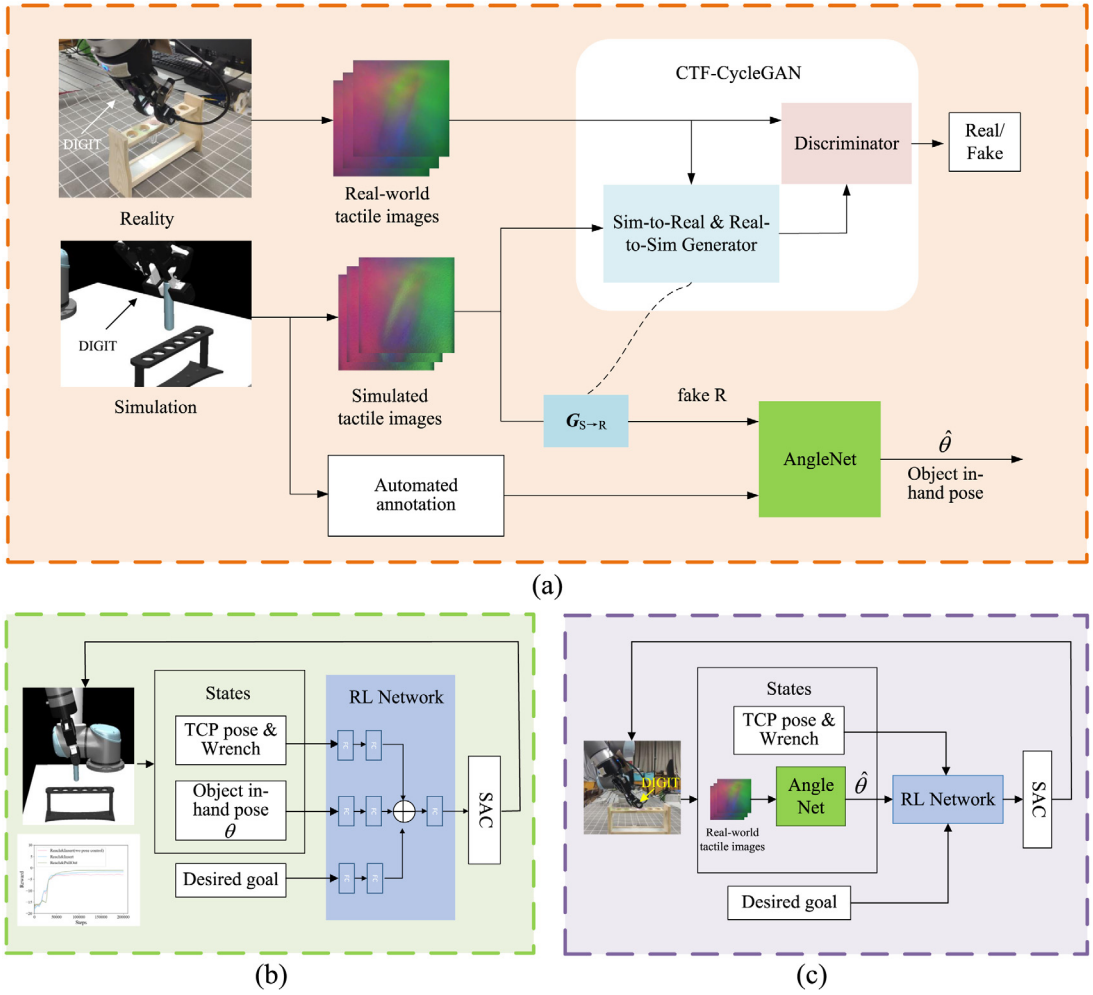

Fig. 2. An overview of our proposed framework. (a) includes training the CTF-CycleGAN and the AngleNet, and AngleNet is trained separately from the CTF-CycleGAN, (b) shows the robot manipulation policy learning in the simulator, and (c) shows the deployment of the model in the real-world setting.

2.2. Sim-to-real transfer of robot manipulation skills

Recent studies have tried to improve the data-efficiency by training the policy in simulation and transferring the acquired skills to the real-world setup. However, because of the domain gaps, the performance will degrade when the model trained in simulation is directly deployed to reality. Several methods have been proposed to mitigate this issue, which can be categorized into three types: domain randomization, domain adaption, and network distillation [27].

Domain randomization methods [28–31] provide enough simulated variability during training so that the model is able to generalize to reality. In particular, adding random noise, random color transformation, or geometric transformation to the simulated images helps data distribution get closer to the actual one as much as possible, which improves the image-based domain randomization method. Some dynamics parameters are randomized in [28], such as the mass of each part of the robot, the damping at the joint interface and so on. The method in [29] randomizes visual information, including the shape and number of sundries, the texture of the objects, etc. The ideas of methods mentioned above for Sim2Real problems are similar, which both narrow the gap by expanding the dataset.

In contrast, by adaptively learning from the images and acquiring the data distribution of the target domain, the domain adaptation methods [10–12,32] effectively handle the problem of the inconsistent probability distribution between the training samples and the testing samples. For instance, Patel et al. [32] propose a model for estimating the tactile images from the in-contact depth data. A work close to ours is [12], in which CycleGAN is utilized to narrow the gap between simulation and reality for GelSight. But our work is different from [12] in that we propose an improved CTF-CycleGAN and have applied the network to tactile-based robot manipulation tasks.

In this work, we focus on the pixel-level domain adaptation approach. For the benefit of the trained RL policy model adapting to reality better, we additionally perform a set of random operations to ensure that the simulator settings accurately reflect real scenarios. As a result, we are able to implement the sim-to-real transfer without training robot manipulation skills in reality or distilling a policy network using real-world data.

3. Proposed methods

Fig. 2 illustrates the overview of the proposed method. In the training stage, the tactile image transfer network CTF-CycleGAN is trained using unpaired simulated and real-world tactile images. This means that they can be captured in the simulator and the real world independently, at inconsistent poses and with distinct pressing forces. Then, the synthetic tactile images captured from the simulator are converted to generated tactile images by the sim-to-real generator $G_{S\to R}$ of CTF-CycleGAN. Using the generated tactile images together with the automatically generated labels of their simulation counterparts, we train the object in-hand pose

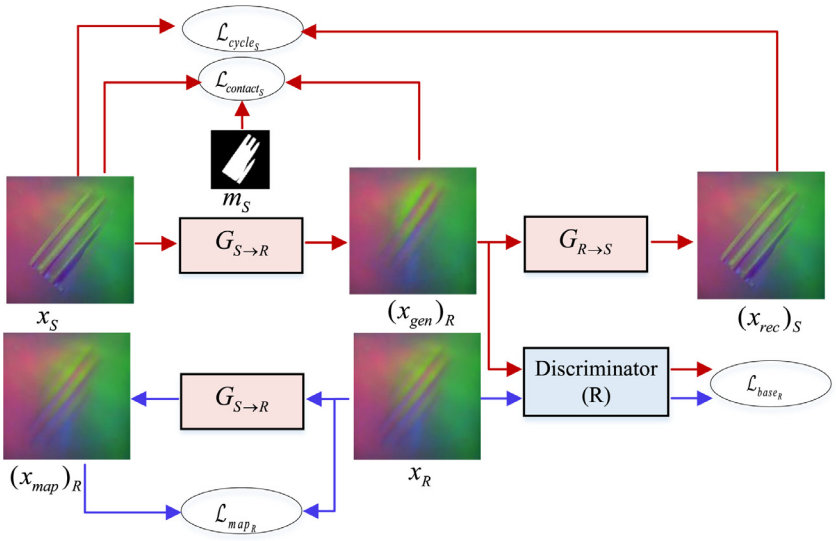

Fig. 3. The overall framework of the proposed CTF-CycleGAN. It includes both sim-real-sim and real-sim-real cycle generation pipelines, in which the generators and discriminators will be alternately and iteratively trained according to the weighted sum of all losses.

estimation network AngleNet to produce 1D in-hand poses of tubular objects. Meanwhile, the robotic insert and pullout manipulation policies are trained in the simulator using a reinforcement learning algorithm.

In the deployment stage, real-world tactile images are directly fed into the trained AngleNet model. Since the AngleNet model is trained on the generated tactile images with minor domain gaps, it generalizes well to real-world tactile images in the pose estimation task. As the observations of states (e.g., object's in-hand pose, robot's end-effector pose) are domain-invariant, the tactile–motor policy network trained in the simulator can be applied to real-world settings without further fine-tuning.

3.1. Transfer of tactile images

The proposed CTF-CycleGAN, as shown in Fig. 3, is built on CycleGAN [33] with improvements in the generator and discriminator. Moreover, the mask information of the synthetic tactile images is utilized to constrain the generator to produce higher-quality details while retaining the structure information, which is important for the in-hand tubular object pose estimation.

Network Architecture: We introduce the self-attention mechanism to construct a novel generator network based on the encoder–decoder structure, as shown in Fig. 4(a). In the encoding stage, the high-level features of the raw tactile images with the resolution of $H \times W$ are firstly extracted using multi-layer convolution blocks. Then the self-attention Transformer encoder [34] connected with the hidden layer is leveraged to improve the feature extraction ability to adapt to the various texture changes of the salient areas in the tactile images. We consider this design mainly owing to that self-attention has a significant ability to capture global context information in tactile images, thereby focusing on regions with salient features. Besides, employing the pre-training model and introducing a self-attention encoder structure at the high level are capable of omitting the retraining process with large-scale data while promoting network convergence.

In the decoding stage, following the U-shaped structure design [35], the self-attention features extracted by the Transformer encoder are upsampled and concatenated with the high-resolution CNN features obtained from the convolution block of different layers to achieve high-quality image generation.

We also design a multi-layer connection network as a discriminator based on [36], which replaces the fully connected layer with the fusing layer, as is shown in Fig. 4(b). The structure is capable of smoothly connecting the discrimination information hidden in different scale layers and enhancing the feature extraction ability. In addition, the perceptual area of the discriminator for the tactile images is decreased to obtain the output in the form of a matrix instead of a point, which is able to improve the domain classification capability of the discriminator. Therefore, the discriminator will potentially stimulate the generator to maintain the clarity and details in tactile image generation.

Loss Optimization: We leverage the basic adversarial loss $\mathcal{L}_{\text{base}}$ and cycle-consistency loss $\mathcal{L}_{\text{cycle}}$ to maintain the consistency of the texture content of the source domain samples during the image generation process. $\mathcal{L}_{\text{base}}$ consists of $\mathcal{L}_{\text{base}_R}$ and $\mathcal{L}_{\text{base}_S}$, and $\mathcal{L}_{\text{cycle}}$ includes $\mathcal{L}_{\text{cycle}_R}$ and $\mathcal{L}_{\text{cycle}_S}$. LSGAN [36] is leveraged to calculate the basic adversarial loss. Meanwhile, we introduce the mapping consistency loss to regularize the generator, which can preserve the colors when the tactile image gets translated from one domain to the other and avoid superfluous translation [37]. For instance, the consistency loss guarantees that the sim-to-real generator will not map the input data that matches the identity of the real-world domain to other data domains.

$$\mathcal{L}_{\text{map}_S}(G_{R\to S}, X_S) = \mathbb{E}_{x_S\sim X_S}[\|G_{R\to S}(x_S) - x_S\|_1] \tag{1}$$

$$\mathcal{L}_{\text{map}_R}(G_{S\to R}, X_R) = \mathbb{E}_{x_R\sim X_R}[\|G_{S\to R}(x_R) - x_R\|_1] \tag{2}$$

where $\mathcal{L}_{\text{map}}$ consists of $\mathcal{L}_{\text{map}_R}$ and $\mathcal{L}_{\text{map}_S}$; $G_{S\to R}$ and $G_{R\to S}$ represent the sim-to-real and real-to-sim generators, respectively; X_R and X_S represent the datasets in the real-world and simulation domain, respectively; x_R and x_S represent the real-world image sample and the simulated image sample, respectively.

We introduce a task-related loss, which is the contact region consistency loss with a mask information to focus on the foreground region of the pressed object surface. Therefore, the generator has a greater capability to preserve the structural features in the contact region. Using the mask information conveniently obtained in the simulator, the mask-based Pairwise Mean Square Error (PMSE) is further calculated on the simulated images and the generated realistic style images. As a result, this enables the generator to retain the basic structural information and task-related information. In particular, the corresponding simulated depth image d_S is truncated according to the depth data without object contact to obtain a binary mask image m_S. The calculation

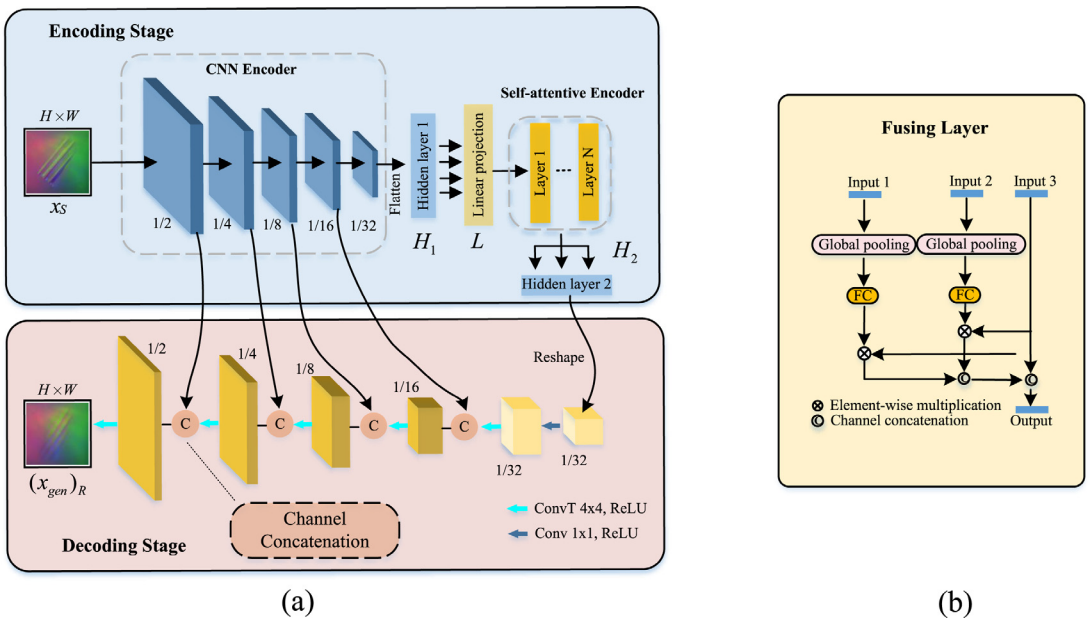

Fig. 4. (a) Network architecture of the generator. In the encoding stage, multi-layer convolution blocks are first utilized to extract the high-level features of the raw simulated image x_S, and then the features are followed by a self-attentive encoder with a hidden layer. In the decoding stage, the extracted self-attentive features are upsampled to be combined with different high-resolution CNN features skipped from the encoding path. (b) The Fusing Layer of the discriminator. Input1, Input2 and Input3 represent the output features of the first, third and fourth residual blocks, respectively.

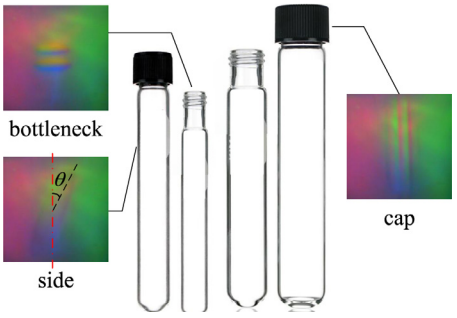

Fig. 5. Various types of test tubes. The test tubes are different in thickness, texture of caps, as well as thread of bottlenecks.

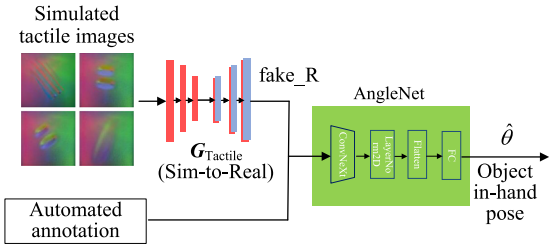

Fig. 6. The architecture of AngleNet.

formulation of contact region consistency loss is obtained as follows:

$$\begin{aligned}\mathcal{L}_{\text{contacts}}\ &(M_S, X_S, G_{S\to R})\\ =&E_{x_S\sim X_S}\left[\frac{1}{k}\left\|\left(x_S - G_{S\to R}\left(x_S\right)\right)\circ m_S\right\|_2^2\right.\\ &\left.-\frac{1}{k^2}\left(\left(x_S - G_{S\to R}\left(x_S\right)\right)^{\mathrm{T}} m_S\right)^2\right]\end{aligned} \tag{3}$$

where k represents the number of the input image pixels; $\circ$ represents Hadamard product.

The overall loss includes four parts: basic adversarial loss, cycle-consistency loss, mapping consistency loss, and contact region consistency loss.

$$\begin{aligned}\mathcal{L}_{\text{overall}}&\left(G_{S\to R}, G_{R\to S}, D_S, D_R, X_S, X_R\right)\\ =&\lambda_{\text{base}}\left(\mathcal{L}_{\text{base}_R} + \mathcal{L}_{\text{base}_S}\right)\\ &+ \lambda_{\text{cycle}}\left(\mathcal{L}_{\text{cycle}_R} + \mathcal{L}_{\text{cycle}_S}\right)\\ &+ \lambda_{\text{map}}\left(\mathcal{L}_{\text{map}_R} + \mathcal{L}_{\text{map}_S}\right)\\ &+ \lambda_{\text{contact}}\mathcal{L}_{\text{contact}_S}\end{aligned} \tag{4}$$

where D_R and D_S represent the discriminator in the real-world and simulation domain, respectively; λ_{base}, λ_{cycle}, λ_{map} and λ_{contact} are the empirical coefficients of the basic adversarial loss, the cycle-consistency loss, the mapping consistency loss, and the contact region consistency loss respectively. Therefore, the optimal objective model $G^*_{R\to S}$ and $G^*_{S\to R}$ are obtained according to the following optimization problems.

$$G^*_{R\to S}, G^*_{S\to R} = \arg\min_{G_{R\to S}, G_{S\to R}, D_S, D_R}\max\mathcal{L}_{\text{overall}} \tag{5}$$

3.2. In-hand pose estimation of tubular objects

As shown in Fig. 5, we consider three possible parts of different test tubes to be held by the robot, including side, bottleneck and cap. We also consider variations in the tubes' diameter, the texture of caps, as well as thread of bottlenecks. The diversity in the in-contact surface texture makes it challenging to extract pose-aware tactile features using hand-engineered descriptors.

In this work, the pose of the manipulated tubular object in relation to the robot gripper is included in the observation data for the robot insert and pullout manipulation tasks. The pose θ describes the angle at which the manipulated tube is deflected from the tactile image's vertical main axis. For estimating θ, we design an object in-hand pose estimation network, AngleNet, which takes ConvNeXt [38] as the backbone and is added with a fully connected layer for the angle regression, as shown in Fig. 6.

By converting synthetic tactile images via the sim-to-real generator, we obtain the generated tactile images, which are used

to train AngleNet. The angle annotations are taken from the simulated samples, and thus no real-world annotations are required. In addition, the superior performance of CTF-CycleGAN's generator guarantees the AngleNet model trained on generated tactile images adapts well to real-world tactile images, without any fine-tuning using real-world data.

3.3. Manipulation policy learning

In this work, we use Soft Actor–Critic (SAC) [39,40] to learn the continuous control task of robotic tactile-based insert and pullout. In our case, the clearance between the tubes and the holes of the rack is large and the robot's gripper cannot hold the tubular object tight. Therefore, sensing the contact force during the manipulation using a force sensor attached to the end of the robot is unreliable. Instead of using force control, the insert and pullout policies are mainly implemented by adjusting the tube's pose using in-hand tactile sensing. In order to avoid jamming, we also construct a reward based on the wrench readings of the robot.

Observations: For encapsulating the diversity in the stiffness and friction of the contact, we use the pose of the end-effector, the sensed tactile, together with the wrench reading to model the state of the tasks.

- *Kinematic:* It provides the information about the position (three-dimensional coordinate) and posture (quaternion form) information of the Tool Center Point (TCP) of the end-effector in the world coordinate system. It is a one-dimensional array of 7 components: $O_K = [p_x, p_y, p_z, q_w, q_x, q_y, q_z]$
- *Tactile:* 1D deflection angle information provided by AngleNet: $O_T = [\hat{\theta}]$
- *Wrench:* The wrench reading of the robot is used to determine whether there are problems such as failure to insert and jamming: $Q_W = [w_x, w_y, w_z]$

Actions: The robot action array is composed of position and posture displacement of the end-effector in three-dimensional space: $A = [\Delta p_x, \Delta p_y, \Delta p_z, \Delta q_w, \Delta q_x, \Delta q_y, \Delta q_z]$

Reward: In this work, the robot manipulation skills include insert and pullout tasks. These two tasks are essentially composed of a sequence of goal reach and pose control actions.

The goal reach is to send the object to be manipulated from the initial position to the desired goal; The pose control is to ensure that there are no excess collisions or contact between the manipulated tubes and the hole during the insert and pullout process. By defining a set of way-points to separate the two types of actions, the success rates of insert and pullout tasks have increased significantly. Therefore, the reward settings during the training are as follows:

$$Reward = \alpha_0 * g_{diff} + \alpha_1 * a_{diff} + \alpha_2 * w_t \tag{6}$$

where g_{diff} represents the distance difference between the end position of the robot TCP and the target position in the world coordinate system, a_{diff} represents the angular deviation of the object in the world coordinate system concerning the vertical axis (the Z-axis we set as the world coordinate system), and w_t represents the wrench signal which is determined by thresholding the wrench reading from the robot controller. As the wrench reading is an estimation based on the required current in the motors, $w_t = 1$ indicates that the robot arm is stuck when performing the peg-in-hole action. α_0, α_1 and α_2 represent the corresponding weight, with $\alpha_0 = -1$, $\alpha_1 = -0.04$ and $\alpha_2 = -1$ in our experiment obtained by grid search.

Evaluation Metrics: To assess the effectiveness of the model, we primarily rely on three indications.

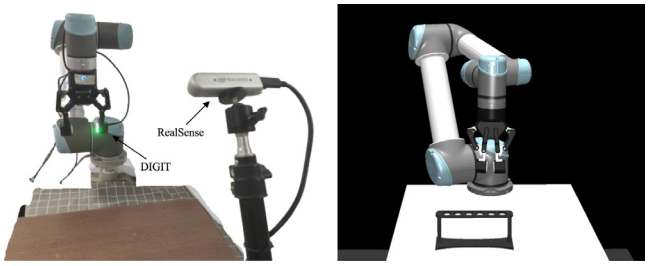

Fig. 7. The real-world experimental setup (left) and the simulator (right).

- *Reach Goal:* whether the center of mass of the manipulated object reaches the set target point;
- *Object Stability:* whether the container of the manipulated object is knocked down by the object;
- *Wrench Situation:* whether the actuators of the robot arm wrench during movement.

4. Experimental setup and data preparation

4.1. Real-world experiment setup

The real-world experimental system entails the use of a UR5 robot, a Robotiq 2F-85 gripper equipped with two DIGIT sensors in both fingers and a RealSense camera, as shown in the left part of Fig. 7.

The optical tactile sensors based on GelSight consist of a slab of transparent elastomer covered with a reflective coating membrane. When an object is pressed on the elastomer, the membrane stretches to conform to its surface while maintaining a constant reflectance. A camera placed in the sensor records the image of this distortion, using illumination from light sources in different directions. A photometric stereo algorithm [41] is then used to reconstruct the depth map of the surface.

4.2. Simulation experiment setup

In our previous work [7], we have presented a Gazebo-based GelSight simulator, in which the depth image of the object in contact with the elastomer is captured in the field-of-view of a simulated camera in the physical engine. To simulate the smooth contact edges and the bump contouring around the in-contact region of the real sensor, we process the obtained elastomer deformation height map by Gaussian filtering, followed by Gaussian difference. After that, the Phong illumination model [42] is applied to compute the internal illumination of the elastomer and therefore render the tactile image. See [7] for more details.

In this work, we build an RL-based robot manipulation skill learning platform using MuJoCo [43], as shown in the right part of Fig. 7. In order to integrate the tactile sensor simulator into the platform, we use a modified version of MuJoCo-based GelSight simulator instead of the PyBullet-based DIGIT simulator [8]. We modified the GelSight simulator to produce DIGIT-like tactile images by replacing the background image. Due to its curved surface, the in-touch regions of a tube in DIGIT tactile images are in an ellipse shape, compared to a rectangle shape in GelSight tactile images, as shown in the left and middle columns of Fig. 9. To handle this problem, we replace the flat elastomer in the simulator with a curved one.

However, a new problem arises with the curved elastomer. Since the non-touch region of the elastomer is not flat, the depth reading of non-touch region is not a fixed value, as shown in Fig. 8. This results in difficulty in capturing the contact region in

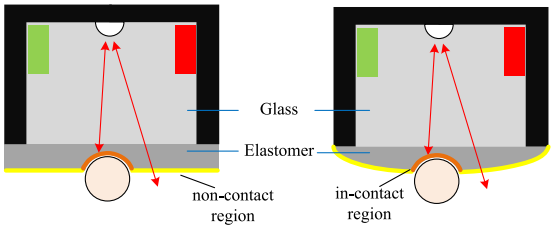

Fig. 8. The flat (left) and curve (right) elastomer in DIGIT simulator.

Fig. 9. Result of the modified simulator. The left is the original simulated GelSight image, the middle is the real DIGIT image, and the right is the simulated DIGIT image with our modified simulator.

Table 1
The parameters setting of our optical tactile simulator.

	$i_{m,s}$, $i_{m,d}$	$\hat{L}_m$	k_d	k_s	k_a
Red	(255,130,115)	(0.0, −1.0, 0.25)	0.3	0.4	1.0
Green	(120, 255, 153)	(0.87, 0.5, 0.25)	0.1	0.4	1.0
Blue	(108, 82, 255)	(−0.87, 0.5, 0.25)	0.1	0.4	1.0

tactile images by thresholding. To solve this problem, we subtract the depth map with object contact from the one without object contact, and then we can project the subtracted depth map to a plane to yield a height map, from which the contact region can be easily extracted. The right column of Fig. 9 shows the final synthetic tactile image captured from the modified simulator, which is more similar to the real one. Detailed parameters setting of the simulated tactile sensor is shown in Table 1.

4.3. Data collection

For training CTF-CycleGAN, the real-world dataset D_{GAN}^{Real} consists of 2790 tactile images, in which 1980 were captured on 20 objects (see [7] for printable object models), and 810 (270 × 3) were taken from the side, cap and bottleneck of different test tubes. Similarly, the simulation data D_{GAN}^{Sim} also includes 1980 + 270 × 3 images, which is unpaired with the real-world data. Besides, the training and testing samples are randomly divided according to 10:1 on both D_{GAN}^{Real} and D_{GAN}^{Sim}.

For training AngleNet and evaluating the relationship between the performance of AngleNet and dataset size, we have divided our dataset into four sub-sets with different sizes, which contain 891 (*Tiny*), 2673 (*Small*), 8019 (*Base*) and 24 057 (*Large*) generated tactile images of different simulated test tubes, respectively. We also collected 540 unseen real-world tactile images of new tubular objects (test tubes with different diameters, droppers) as the testing dataset. These testing samples were collected by mounting a DIGIT sensor to a robot's flange and controlling the robot to touch a fixed object at a predefined sequence of angles. Therefore, the angle ground truth of the testing data can be obtained automatically. It is worth mentioning that collecting real-world test data is time-consuming. A case in point is that we spent almost 12 h collecting 1350 real-world images. In contrast, the time we used to collect 8019 simulated images is just 30 min. The details of the test tubes used in our experiments are explained in Section 5.2.

Table 2
Comparison of CTF-CycleGAN with other methods.

Method	Datasets	FID↓	KID(x100)↓	SSIM↑
Direct	X_S&X_R	110.820	4.159	0.764
CycleGAN	X_R&$(X_{gen})_R$	65.137	2.437	0.660
TGN	X_R&$(X_{gen})_R$	54.541	2.091	**0.797**
Abl.1	X_R&$(X_{gen})_R$	58.233	2.057	0.778
Abl.2	X_R&$(X_{gen})_R$	51.444	1.200	0.780
CTF-CycleGAN	X_R&$(X_{gen})_R$	**46.724**	**0.936**	0.787

4.4. Typical tasks

As we consider robot chemists to handle thousands of test tubes and other tubular objects, the following two scenarios are designed to contain the tactile–motor policies mentioned in Section 3.3, i.e., "Reach&Insert" and "PullOut&Reach". The two scenarios also involve the manipulation of both typical rigid objects (e.g., different parts of test tubes) and typical deformable objects (e.g., the rubber tip of a dropper).

Human–robot Collaborative Tube Placing: The purpose of this experiment is to test the generalization capability of the learned skills toward uncertain grasp positions and unseen textures of objects. When receiving test tubes from a human counterpart repeatedly, robots can hardly regrasp at the same position. In this experiment, we consider the possibility of grasping at three different parts of test tubes as well as variations in the textures of the contact surface, as shown in Fig. 5.

Robotic pipetting: The robot continuously conducts a reagent transfer task containing four steps: insert the dropper into the vial ("Reach&Insert"), press the dropper to suck up the liquid ("SuckUp"), pull out the dropper, move above a goal beaker ("PullOut&Reach"), and press the dropper's tip again to squeeze out the liquid ("SqueezeOut").

5. Experimental results

5.1. CTF-CycleGAN transfer results and ablation analysis

We first compared CycleGAN, TGN (Texture Generation Network in [11]) and CTF-GycleGAN in the sim-to-real pipeline. As shown in Fig. 10, CTF-GycleGAN outperforms CycleGAN and TGN in the quality of generated tactile images. For example, the "cylinder_side" image has obvious generation defects in shape, or the texture of the image has leaked to the background area. In contrast, our method generates tactile images with higher quality. This is because the constraints on contact region consistency suppress the texture leakage phenomenon, and therefore it is possible to better keep the edge structure and texture information in the in-contact region of tactile images.

Then the ablation study compares CTF-CycleGAN with CycleGAN, Abl.1 and Abl.2 for testing the effectiveness of the main components of CTF-CycleGAN. Abl.1 represents the method of ablating the contact region consistency loss and optimized discriminator, and Abl.2 represents the method that only ablates the contact region consistency loss. In addition, "Direct" indicates the evaluation of the difference between the raw simulated dataset X_S and the raw real dataset X_R.

As shown in Table 2, CycleGAN has a significant improvement in the FID and KID indicators in the sim-to-real pipeline compared with "Direct". CycleGAN pays more attention to the transfer of features whilst neglecting the intuitive structural details, resulting in a decrease in the SSIM indicator. Compared with CycleGAN, Abl. 1 has a 6.9 and 0.4 decrease in the FID and the KID metric, respectively. This also verifies that the improved generator can generate higher-quality images to meet the needs of tactile image generation. The comparison between Abl. 1 and Abl. 2 validates

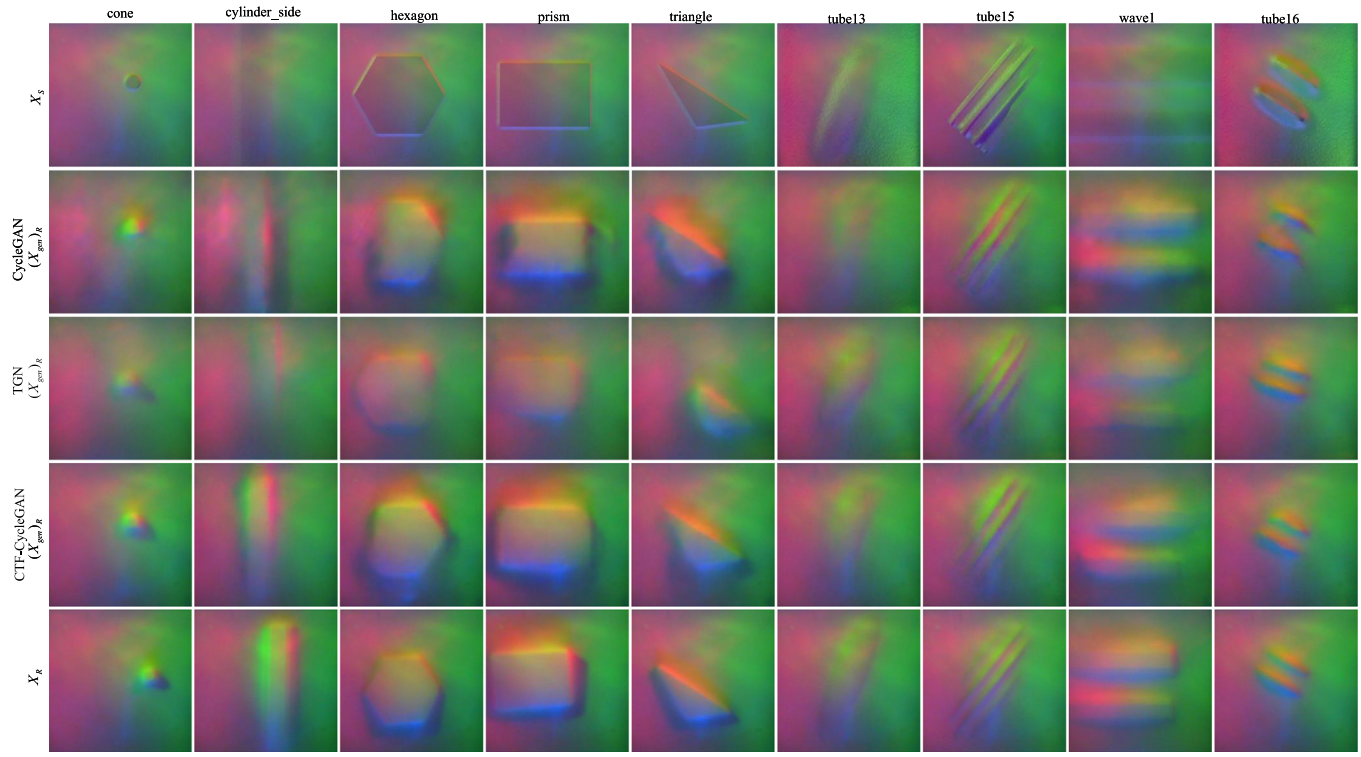

Fig. 10. The comparison of sim-to-real effect using different algorithms. Wherein the first row represents an example of the raw simulated image dataset X_S. The second and third rows represent examples of adaptive real-world image datasets $(X_{gen})_R$ generated by different methods, and the last row represents examples of real-world image dataset X_R.

that the discriminator stimulates the generator to generate more realistic tactile images. With the task-related loss incorporated, CTF-CycleGAN achieves the highest performance in all indicators.

5.2. In-hand pose estimation results

In our experiment, we consider test tubes with 5 different diameters, 5 different textures of caps, as well as 5 different types of bottleneck threads. Examples of them are shown in Fig. 5. For each of the three aspects, we consider 3 types as "Seen" and the other 2 types as "Unseen" in dividing the training and testing dataset. For example, the "Seen" diameters are 13 mm, 15 mm and 16 mm and the "Unseen" diameters are 12 mm and 18 mm.

The estimation results of the object in-hand pose estimation network AngleNet, trained on *Tiny*, *Small*, *Base* and *Large* sub-sets, are shown in Table 3. In the table, "Sim2Sim" means training and testing exclusively on simulated data, and "Direct" denotes training on simulated data and testing directly on real-world data. "Sim2Real" means training on data generated by the CTF-CycleGAN sim-to-real generator which we freeze, and testing on real-world data. As the table shows, in general, the predicted angle accuracy of AngleNet rises with the increase of the dataset size. Particularly, the evaluation performance trained on the *Base* and the *Large* sub-sets are very close. Hence, we use the *Base* sub-set with 8019 simulated images in other comparison and ablation experiments of AngleNet.

For each sub-set and each method in Table 3, it is comprehensible that the accuracy of the "U" row is lower than that of the "S" row, as the "U" row indicates the generalization performance of the models on unseen objects. According to Table 3, the high accuracy of "Sim2Sim" implies that AngleNet ensures accurate in-hand object pose estimation. In addition, by feeding the pose estimator with transferred tactile images, "Sim2Real" outperforms "Direct" on all sub-sets, which indicates that our method is capable of reducing the domain gap and accurately estimating the in-hand object pose, even though the pose estimator is trained only in a simulator.

We also conducted some comparison and ablation experiments in testing the accuracy of AngleNet to demonstrate the performance of our method, as is shown in Table 4. The meanings of TGN, Abl.1 and Abl.2 have been claimed in Section 5.1. Besides, the models of those experiments are all trained on the data generated by the sim-to-real generator of CTF-CycleGAN and other methods for comparison or ablation, and tested on real-world data. As the table shows, CTF-CycleGAN has the best performance compared with other methods.

5.3. Robot policies learning results

In our work, the length of each tube is not known as a prior, but it is assumed that the lengths of all tubes are within the range of 10 cm–15 cm. In order to solve the problem of unknown tube length, we design two way-points, i.e., approach position and end position (see Fig. 13), and make sure that the vertical distance between the two points is greater than the maximum length of the tubes. Therefore, there is enough space between the approach position and the end position to adjust the pose of various tubes with different lengths. Denote the reagent rack depth as y. For the tube length $\leq y$, the insert is completed when the gripper reaches the end position. For the tube length $> y$, the insert is terminated if a collision between the tube and the bottom of rack is detected according to the wrench reading.

The "Reach&Insert" and "PullOut&Reach" policies are trained in the simulator. We also implemented another version of the "Reach&Insert" without pose control. The "Reach&Insert (without pose control)" aims to verify the importance of pose observation in the "Reach&Insert" task by omitting the observation of object in-hand pose. Fig. 12 shows the reward curves and success rate

Table 3
The quantitative accuracy evaluation results of the object in-hand pose estimation network AngleNet, trained on *Tiny*, *Small*, *Base* and *Large* sub-sets, separately. "S" means seen, and "U" means unseen.

Sub-set		Sim2Sim	Direct	Sim2Real
Tiny	S	95.37% ± 1.88	27.46% ± 8.03	54.71% ± 4.25
	U	59.79% ± 3.65	12.50% ± 8.77	24.67% ± 5.10
Small	S	96.49% ± 1.02	40.35% ± 6.72	80.70% ± 2.58
	U	68.42% ± 2.53	26.32% ± 7.89	57.02% ± 3.04
Base	S	**98.67%** ± 0.89	55.96% ± 5.32	85.92% ± 2.25
	U	71.36% ± 1.12	**34.18**% ±5.87	60.98% ± 2.65
Large	S	98.02% ± 1.08	**56.63%** ±5.01	**86.70%** ±1.99
	U	**72.50%** ±1.67	33.35% ± 5.60	**62.33%** ±2.48

Table 4
The quantitative Sim2Real accuracy evaluation results of AngleNet trained on *Base* sub-set using the sim-to-real generator of CTF-CycleGAN and other methods for comparison or ablation.

Method	Seen	Unseen
CycleGAN	72.85% ± 3.06	44.82% ± 4.11
TGN	74.67% ± 3.20	47.02% ± 3.54
Abl.1	80.13% ± 2.97	55.98% ± 3.17
Abl.2	82.88% ± 2.55	58.76% ± 2.99
CTF-CycleGAN	**85.92%** ± 2.25	**60.98%** ± 2.65

Table 5
The success rate of seen, unseen and all test tubes in Real-world Scenario 1 (row1, row2 and row3). The success rate of the dropper in Real-world Scenario 2 (row4).

Type	Success Rate
Tube_Seen	83.33%
Tube_Unseen	70.00%
Tube_All	78.00%
Dropper	90.00%

curves of the three task training processes. The "Reach&Insert (without pose control)" only has an 18.50% success rate during the training phase in the simulated environment, as can be observed. In contrast, the success rate of the "Reach&Insert" reaches 92.70%. For the "PullOut&Reach" task, the success rate reaches 100.00%. Meanwhile, to improve the robustness of the learned insert and pullout policies against contact and friction, we performed random disturbance to the deflection angle of the manipulated object in the observation of object in-hand pose during training.

To compare with the decoupled training strategy, we also attempted to train the AngleNet and the policy as a whole, i.e., AngleNet is updated according to the loss of the policy. However, the training becomes difficult to converge while the training failure rate increases. The reason is that the "Reach" part of the "Reach&Insert" policy is independent of tactile perception, and thus the two parts cannot benefit from each other during the training. In contrast, by decoupling the training of policy from that of tactile image transfer, the transferable skills are much easier to be trained.

5.4. Real-world Scenario 1: Human–robot collaborative tube placing

Fig. 13(a.0-7), (b.0-7), and (c.0-7) show three continuous tube-placing episodes, in which a human participant passes the test tube to the robot and the robot holds sequentially the tube at the bottleneck, the side and the cap.

To provide the desired goal for the "Reach&Insert" action, we detect and locate the round holes on the wooden reagent rack using RealSense (see the left of Fig. 11) and then transform the position of each hole to the robot base coordinate system. As shown in Fig. 13(a.0-7), (b.0-7), and (c.0-7), although the robot

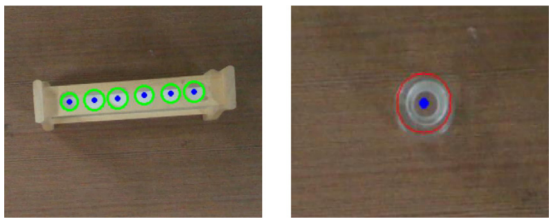

Fig. 11. Detection and location of the reagent rack (left) and the reagent vial (right).

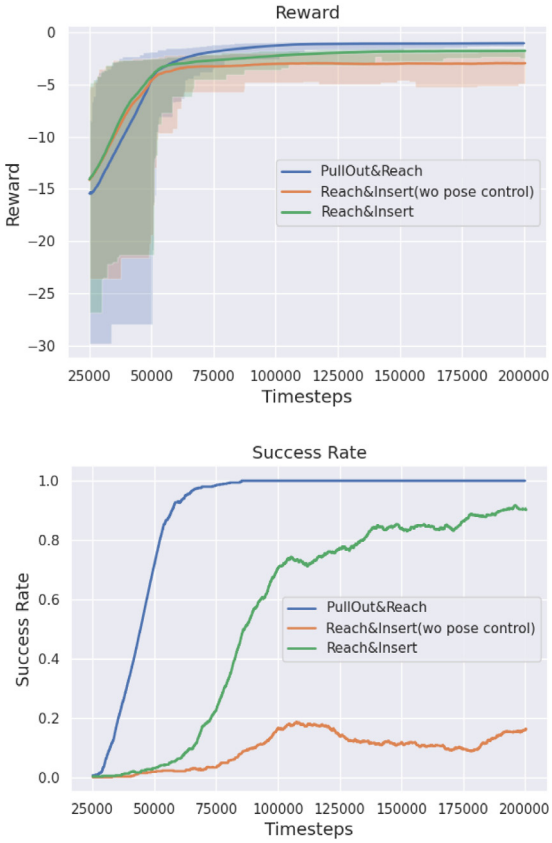

Fig. 12. Reward curves (top) and success rate curves (bottom) of the three task training processes in the simulation domain.

holds the test tubes of various lengths with different unknown poses, it adjusts the gripper's angle to vertically insert the test tubes into the three holes of the reagent rack.

The first three rows of Table 5 present the task success rate of seen, unseen and all test tubes in this experiment. It shows that the success rates are 83.33%, 70.00%, and 78.00%, respectively. We

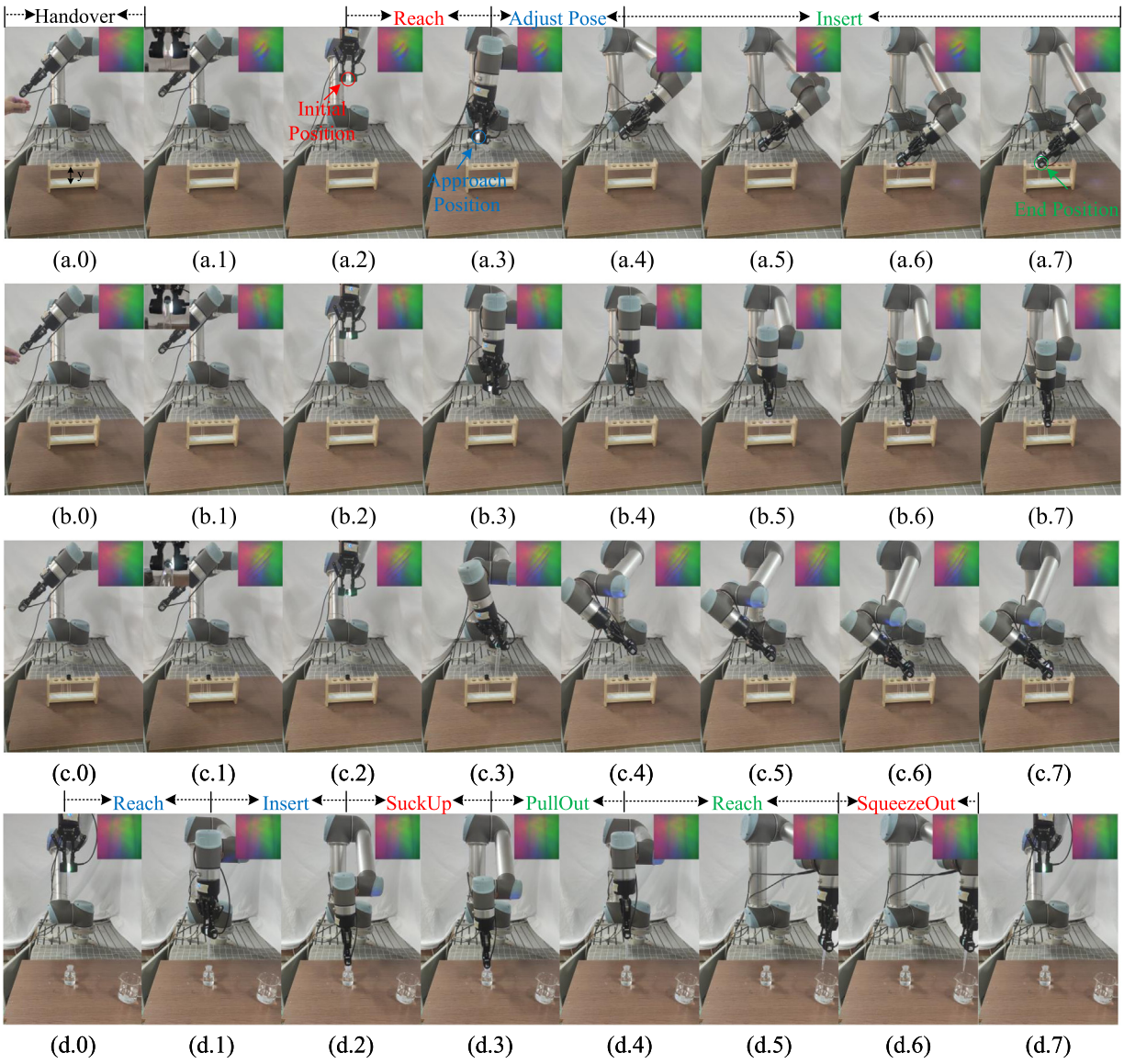

Fig. 13. Experimental snapshot in real-world scenarios. The first three rows are the "Human–robot Collaborative Tube Placing" task, and the last row is the "Robotic Pipetting" task.

also recorded the error bars of the deflection angle θ_Z between the test tube and the Z-axis of the world coordinate system for different test tube parts (e.g., side, bottleneck, and cap) at each key stage of the task (initial position, reaching above the hole, and insert completion), as shown in Fig. 14. The result in Fig. 14 demonstrates that our method ensures that the deflection angle gradually approaches zero degree by adjusting the gripper pose, until the insert task is completed.

5.5. Real-world Scenario 2: Robotic pipetting

Fig. 13(d.0-7) shows a snapshot of the robotic pipetting scenario, in which the robot sequentially conducts the four steps of "Reach&Insert", "SuckUp", "PullOut&Reach" and "SqueezeOut" to transfer reagent from a vial to a beaker. Details of the task can also be found in Section 4.4. In this experiment, the dropper's tip is constantly held by the robot's gripper, and we choose a simple and pure background of the environment to ease the localization of the reagent vial and the beaker using RealSense (see the right of Fig. 11). The "SuckUp" and "SqueezeOut" operations are performed by applying adequate grip force to the gripper according to the size of the in-contact region in the tactile image.

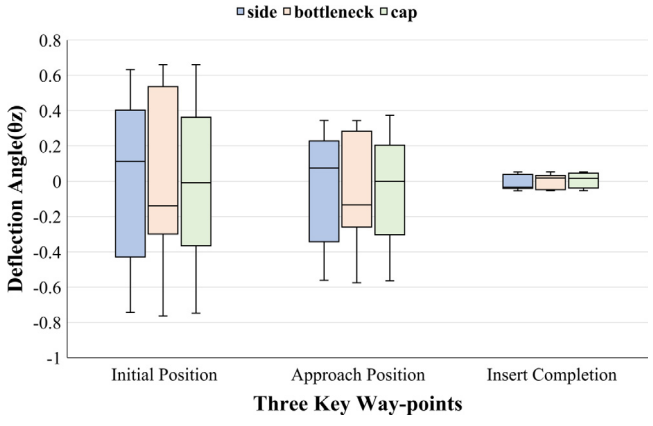

Fig. 14. The error bars of the deflection angle in Real-world Scenario 1. We recorded the angles in three key way-points for the side, bottleneck and cap of the test tube.

It is worth noticing that we applied light grip force to the dropper's tip and therefore the texture of the in-contact region in the tactile image is relatively indistinct. Nevertheless, thanks to the generalization ability of the AngleNet, the robot could still adjust the dropper's pose and complete the insert manipulation. As shown in row 3 of Table 5, after 30 consecutive episodes of experiments, the overall task success rate reaches 90.00%.

6. Conclusion

In this paper, we propose a skill generalization method for robotic tubular object manipulation tasks. In particular, we present a novel sim-to-real transferable in-hand pose estimation method using unsupervised adversarial domain adaptation of tactile images. Using the in-hand tubular object pose estimation results, the RL-based object insert and pullout policies learned in a simulator generalize well to unseen real-world in-contact object texture, as well as the sim-to-real domain gaps of the tactile sensor. The proposed method is further applied to a human–robot collaborative tube placing scenario and a robotic pipetting scenario for a robot chemist. Experimental results validate the performance of sim-to-real transfer of tactile–motor policy as well as the capability of handling unseen textures of tubular objects. The proposed method is independent of pose annotations for real-world tactile images, and only unpaired simulated/real-world tactile images are required for training the CTF-CycleGAN. In the future, we plan to focus on more experimental scenarios and construct a more universal experimental platform.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is sponsored by the Natural Science Foundation of Jiangsu Province, China (No. BK20201264), Zhejiang Lab (No. 2022NB0AB02), and the National Natural Science Foundation of China (No. 61573101).

References

[1] B. Burger, P.M. Maffettone, V.V. Gusev, C.M. Aitchison, Y. Bai, X. Wang, X. Li, B.M. Alston, B. Li, R. Clowes, et al., A mobile robotic chemist, Nature 583 (7815) (2020) 237–241.
[2] J. Jiang, G. Cao, T.-T. Do, S. Luo, A4T: Hierarchical affordance detection for transparent objects depth reconstruction and manipulation, 2022, https://arxiv.org/abs/2207.04907.
[3] W. Yuan, S. Dong, E.H. Adelson, Gelsight: High-resolution robot tactile sensors for estimating geometry and force, Sensors 17 (12) (2017) 2762.
[4] W. Yang, C. Paxton, M. Cakmak, D. Fox, Human grasp classification for reactive human-to-robot handovers, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2020, pp. 11123–11130.
[5] X. Jing, K. Qian, X. Xu, J. Bai, B. Zhou, Domain adversarial transfer for cross-domain and task-constrained grasp pose detection, Robot. Auton. Syst. 145 (2021) 103872.
[6] T. Bi, C. Sferrazza, R. D'Andrea, Zero-shot sim-to-real transfer of tactile control policies for aggressive swing-up manipulation, IEEE Robot. Autom. Lett. 6 (3) (2021) 5761–5768.
[7] D.F. Gomes, P. Paoletti, S. Luo, Generation of gelsight tactile images for sim2real learning, IEEE Robot. Autom. Lett. 6 (2) (2021) 4177–4184.
[8] S. Wang, M. Lambeta, P.-W. Chou, R. Calandra, TACTO: A fast, flexible, and open-source simulator for high-resolution vision-based tactile sensors, IEEE Robot. Autom. Lett. 7 (2) (2022) 3930–3937.
[9] Z. Si, W. Yuan, Taxim: An example-based simulation model for gelsight tactile sensors, IEEE Robot. Autom. Lett. 7 (2) (2022) 2361–2368.
[10] A. Church, J. Lloyd, N.F. Lepora, et al., Tactile sim-to-real policy transfer via real-to-sim image translation, in: Conference on Robot Learning, PMLR, 2022, pp. 1645–1654.
[11] T. Jianu, D.F. Gomes, S. Luo, Reducing tactile Sim2Real domain gaps via deep texture generation networks, in: 2022 International Conference on Robotics and Automation, IEEE, 2022, pp. 8305–8311.
[12] W. Chen, Y. Xu, Z. Chen, P. Zeng, R. Dang, R. Chen, J. Xu, Bidirectional sim-to-real transfer for GelSight tactile sensors with CycleGAN, IEEE Robot. Autom. Lett. 7 (3) (2022) 6187–6194.
[13] S. Luo, J. Bimbo, R. Dahiya, H. Liu, Robotic tactile perception of object properties: A review, Mechatronics 48 (2017) 54–67.
[14] R.S. Dahiya, G. Metta, M. Valle, G. Sandini, Tactile sensing—from humans to humanoids, IEEE Trans. Robot. 26 (1) (2009) 1–20.
[15] Z. Kappassov, J.-A. Corrales, V. Perdereau, Touch driven controller and tactile features for physical interactions, Robot. Auton. Syst. 123 (2020) 103332.
[16] F. Zhu, R. Jia, L. Yang, Y. Yan, Z. Wang, J. Pan, W. Wang, Visual-tactile sensing for Real-time liquid Volume Estimation in Grasping, 2022, arXiv preprint arXiv:2202.11503.
[17] T. Kelestemur, R. Platt, T. Padir, Tactile pose estimation and policy learning for unknown object manipulation, 2022, arXiv preprint arXiv:2203.10685.
[18] S. Xie, Y. Zhang, H. Zhang, M. Jin, Development of triaxis electromagnetic tactile sensor with adjustable sensitivity and measurement range for robot manipulation, IEEE Trans. Instrum. Meas. 71 (2022) 1–9.
[19] D.F. Gomes, Z. Lin, S. Luo, GelTip: A finger-shaped optical tactile sensor for robotic manipulation, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2020, pp. 9903–9909.
[20] B. Ward-Cherrier, N. Pestell, L. Cramphorn, B. Winstone, M.E. Giannaccini, J. Rossiter, N.F. Lepora, The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies, Soft Robot. 5 (2) (2018) 216–227.
[21] 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 Robot. Autom. Lett. 5 (3) (2020) 3838–3845.
[22] S. Luo, W. Yuan, E. Adelson, A.G. Cohn, R. Fuentes, Vitac: Feature sharing between vision and tactile sensing for cloth texture recognition, in: 2018 IEEE International Conference on Robotics and Automation, IEEE, 2018, pp. 2722–2727.
[23] L. Pecyna, S. Dong, S. Luo, Visual-tactile multimodality for following deformable linear objects using reinforcement learning, 2022, arXiv:2204.00117.
[24] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, E. Adelson, Cable manipulation with a tactile-reactive gripper, Int. J. Robot. Res. 40 (12–14) (2021) 1385–1401.
[25] S. Dong, D.K. Jha, D. Romeres, S. Kim, D. Nikovski, A. Rodriguez, Tactile-rl for insertion: Generalization to objects of unknown geometry, in: 2021 IEEE International Conference on Robotics and Automation, IEEE, 2021, pp. 6437–6443.
[26] Y. Han, R. Batra, N. Boyd, T. Zhao, Y. She, S. Hutchinson, Y. Zhao, Learning generalizable vision-tactile robotic grasping strategy for deformable objects via transformer, 2021, arXiv:2112.06374.
[27] J. Ibarz, J. Tan, C. Finn, M. Kalakrishnan, P. Pastor, S. Levine, How to train your robot with deep reinforcement learning: lessons we have learned, Int. J. Robot. Res. 40 (4–5) (2021) 698–721.
[28] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, D. Fox, Closing the sim-to-real loop: Adapting simulation randomization with real world experience, in: 2019 International Conference on Robotics and Automation, IEEE, 2019, pp. 8973–8979.
[29] H. Niu, J. Hu, Z. Cui, Y. Zhang, DR2L: Surfacing corner cases to robustify autonomous driving via domain randomization reinforcement learning, in: The 5th International Conference on Computer Science and Application Engineering, 2021, pp. 1–8.
[30] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, P. Abbeel, Domain randomization for transferring deep neural networks from simulation to the real world, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2017, pp. 23–30.
[31] G. Tiboni, K. Arndt, V. Kyrki, DROPO: Sim-to-real transfer with offline domain randomization, 2022, arXiv preprint arXiv:2201.08434.
[32] K. Patel, S. Iba, N. Jamali, Deep tactile experience: Estimating tactile sensor output from depth sensor data, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2020, pp. 9846–9853.
[33] J.-Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2223–2232.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).
[35] O. Ronneberger, P. Fischer, 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.
[36] X. Mao, Q. Li, H. Xie, R.Y. Lau, Z. Wang, S. Paul Smolley, Least squares generative adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2794–2802.
[37] H. Dou, C. Chen, X. Hu, L. Jia, S. Peng, Asymmetric CycleGAN for image-to-image translations with uneven complexities, Neurocomputing 415 (2020) 114–122.
[38] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A convnet for the 2020s, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11976–11986.
[39] T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, in: International Conference on Machine Learning, PMLR, 2018, pp. 1861–1870.
[40] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, et al., Soft actor-critic algorithms and applications, 2018, arXiv:1812.05905.
[41] R.J. Woodham, Photometric method for determining surface orientation from multiple images, Opt. Eng. 19 (1) (1980) 139–144.
[42] B.T. Phong, Illumination for computer generated pictures, Commun. ACM 18 (6) (1975) 311–317.
[43] E. Todorov, T. Erez, Y. Tassa, Mujoco: A physics engine for model-based control, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2012, pp. 5026–5033.

Yongqiang Zhao received his B.S. degree in robot engineering from Southeast University in 2021. He is currently a M.S. student at the School of Automation, Southeast University. His research interests include robotic manipulation, tactile sensing and reinforcement learning.

Xingshuo Jing received his M.E. degree in control engineering from Nanjing University of Science and Technology in 2019. He is currently a Ph.D. candidate at the School of Automation, Southeast University. His research interests include robotic grasp learning, transfer learning and reinforcement learning.

Kun Qian received his Ph.D. degree in control theory and control engineering from Southeast University in 2010. He is currently working as an associate professor at the School of Automation, Southeast University. His research interests include robot manipulation learning and robot vision.

Daniel Fernandes Gomes received his M.E degree in Informatics and Computing Engineering from the Faculty of Engineering of the University of Porto in 2017. He is currently a Ph.D. candidate at the University of Liverpool. His research interests include object manipulation and robotic learning.

Shan Luo received the B.Eng. degree in Automatic Control from China University of Petroleum, Qingdao, China, in 2012, and the Ph.D. degree in Robotics from King's College London, UK, in 2016. He is currently a Senior Lecturer (Associate Professor) at the Department of Engineering, King's College London. Previously, he was Lecturer at the University of Liverpool, and Research Fellow at Harvard University and University of Leeds. He was also a Visiting Scientist at the Computer Science and Artificial Intelligence Laboratory (CSAIL), MIT. His research interests include tactile sensing, robot learning and computer vision.