A Bio-inspired Multi-functional Tendon-driven Tactile Sensor and Application in Obstacle Avoidance using Reinforcement Learning

Zhenyu Lu, *Member, IEEE,* Zhou Zhao, Tianqi Yue, Xu Zhu and Ning Wang, *Member, IEEE*

***Abstract*— This paper presents a new bio-inspired tactile sensor that is multi-functional and has different sensitivity contact areas. The TacTop area is sensitive and is used for object classification when there is direct contact. On the other hand, the TacSide area is less sensitive and is used to localize the side contact areas. By connecting tendons from the TacSide area to the TacTop area, the sensor is able to perform multiple detection functions using the same expression region. For the mixed contacting signals collected from the expression region with numerous markers and pins, we build a modified DenseNet121 model which specifically removes all fully connected layers and keeps the rest as a sub-network. The proposed model also contains a global average pooling layer with two branching networks to handle different functions and provide accurate spatial translation of the extracted features. The experimental results demonstrate a high prediction accuracy of 98% for object perception and localization. Furthermore, the new tactile sensor is utilized for obstacle avoidance, where action skills are extracted from human demonstrations and then an action dataset is generated for reinforcement learning to guide robots towards correct responses after contact detection. To evaluate the effectiveness of the proposed framework, several simulations are performed in the MuJoCo environment.**

***Index Terms*— Bio-inspired tactile sensor, Reinforcement learning, Deep learning, Multi-sensitivity, Obstacle avoidance.**

I. Introduction

TACTILE sensors are useful to improve the perceptual ability of robots, especially in contact manipulation, and have attracted much attention in recent years through the study of sensorimotor learning in cognitive systems. The typical tactile sensors are pressure sensor arrays, optical-based tactile sensors, and piezoresistive strain gauge rosettes [1]. Due to the low cost and biometric structure, such as soft contact surface, more and more research are focused on optical tactile sensors and their application in robot manipulation. Well-known optical tactile sensors include GelSight [2], [3], DIGIT [4] and its simulator Tacto [5], and TacTip family developed by Bristol Robotics Laboratory (BRL) [6]-[8] . The embedded camera can capture near-real-time images of the shadow images (e.g., Gelsight and DIGIT) or markers (e.g., TacTip and FingerVision [9]) and process the images using Deep Learning [10]-[12] or visual computing techniques to evaluate the haptic effect.

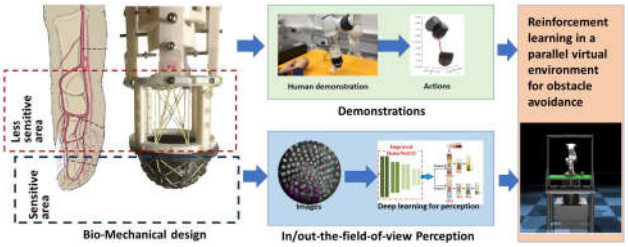


Fig. 1. A tendon-driven tactile sensor: MechTacTip and its application in obstacle avoidance with reinforcement learning

The tactile sensor based on optical measuring units can only detect images within their field of view and is constrained by resolution. Therefore, optical tactile sensor cannot be applied for large surface haptic detection, like electric skin. Some researchers have used various ways to extend visual detecting range of cameras. For example, Winstone et al. designed a Tac Cylinder [14], [7], as a speciality of the TacTip family that has a 3D-printed cylindrical skin and a catadioptric mirror system to achieve 360 -degree tactile sensing. Ma et. al [15] and Wang et. al [16] designed tactile sensors for robot grasping, named GelSlim 2.0 and GelSight Wedge respectively. Both designs have a mirror embedded in the sensor to reflect haptic images to the camera to enable the camera to enable it to view scenes outside the sensor's field of view. Although lenses and mirrors can partially solve the out-of-field-of-view detection problem, they are still limited by properties of the optical components.

In this paper, we propose a new tactile sensor, MechTacTip, which extends the detectable regions of the TacTip through a mechanical approach. It has tendons to connect the out-of-field-of-view areas with the skin and markers within the field of view. The working principle of MechTacTip is illustrated in Fig.2. The operator touches the area with crossing tendons, which are fixed at the bottom of the MechTacTip on one side, and crossed at the top of the MechTacTip on the other side. The connecting topology of tendons is detailed in Fig.4. The functional areas of MechTacTip inside and outside the field of view are named TacTop and TacSide, respectively. Some cubes are distributed

*This work was supported by the Horizon 2020 Marie Skłodowska-Curie Actions Individual Fellowship under grant 101030691. (Corresponding author: Ning Wang)

Zhenyu Lu, and Ning Wang are with the Faculty of Environment and Technology and Bristol Robotics Lab at the University of the West of England, Bristol, BS16 1QY, UK. {Zhenyu.Lu, Ning2.Wang}@uwe.ac.uk.

Zhou Zhao is with EPITA Research and Development Laboratory (LRDE), Le Kremlin-Bicetre, France. zz@lrde.epita.fr

Tianqi Yue and Xu Zhu are with the Department of Engineering Mathematics and Bristol Robotics Laboratory at the University of Bristol, Bristol, BS8 1TW, UK. {Tianqi.yue, xu.zhu.2021}@bristol.ac.uk

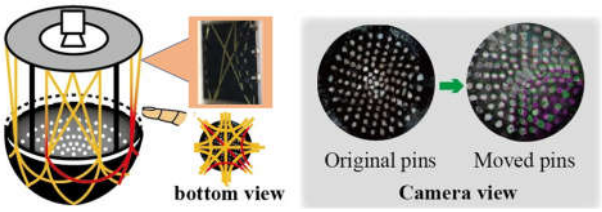


Fig. 2. Working principle of MechTacTip

in a circle on the TacTop with tendon crossing holes. When the TacSide are pushed, the crossing tendons will affect the front face of the TacTop and the pins' distribution on the back face of the TacTop (see the figure at the top right of Fig.3). By utilizing deep learning and a dataset of collected images of pins with labels, MechTacTip can achieve multiple functions, including object recognition through contact with TacTop and location determination through contact with TacSide. Since the unseen but perceptible area of TacSide can recognize the contact area, we further use MechTacTip for robot obstacle avoidance based on reinforcement learning (RL) using human demonstrations. Finally, we verify the effectiveness of the proposed diagram in Fig.1 through a simulation in MuJoCo.

The remainder of this paper is organised as follows: Section II shows the mechanical design and fabrication of MechTacTip. In Section III, we create a dataset of 10051 labelled images and improve the DenseNet121 network for object detection with the TacTop and position localization with the TasSide. Section IV introduces a reinforcement learning (RL) method based on the Gaussian Mixture Model (GMM) to extract various action features and generate an action dataset from human demonstrations. Section V reports on two experiments conducted to verify the perceptual accuracy of MechTacTip using the proposed network and the effectiveness of MechTacTip in obstacle avoidance. Specifically, we build a real platform with parallel simulation and training environment in Mujoco and use RL with a GMM-based action database extracted from human demonstrations. Finally, in Section VI, we conclude the paper and discuss future work.

II. Mechanical Design of MechTacTip

A. Mechnical design and working principles

The structure of MechTacTip is presented in Fig. 3, and it consists of a TacTop and four TacSides, a bracket, a base, a pair of shading shells and an endoscopy camera with LED lighting units (4 pcs) and a fixing unit of the camera. The TacTop has a hemispherical shape similar to that of the TacTip [7]. and it has several cubes distributed uniformly in eight directions on its

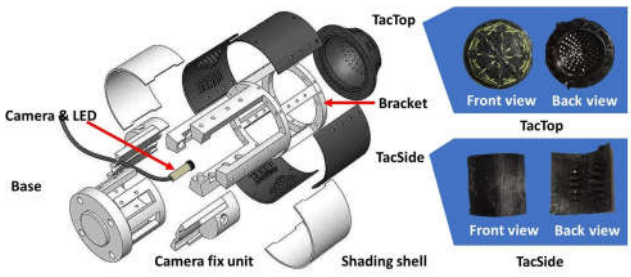


Fig. 3. Mechanical design of MechTacTip

outer surface. Each direction contains three cubes, as shown in the zoomed-in figure on the top right of Fig. 3 and Fig. 4. The inner surface of the TacTop has several pins with white top markers, which reflect the contact on the outer surface or the tension effect through the tendons. The outer and inner surfaces of the TacSide are also different. The outer skin is smooth, while the inner side has a black, thin, elastic membrane to fix the tendons' crossing topology, as shown in the bottom right of Fig. 3. Four TacSide skins are attached to the bracket and fixed by several M3 screws, and each TacSide occupies a quarter of the cylinder to sense touching on the side. As shown in Fig.4, there are several tendons waved at the inner side of TacSide. If any region on the TacSide is pressed, the tendons under this region will transfer the pressing action to the TacTop through tension and affect the current pin's distributions. The distance between the camera and TacTop can be manually adjusted to acquire clear images.

B. Fabrication

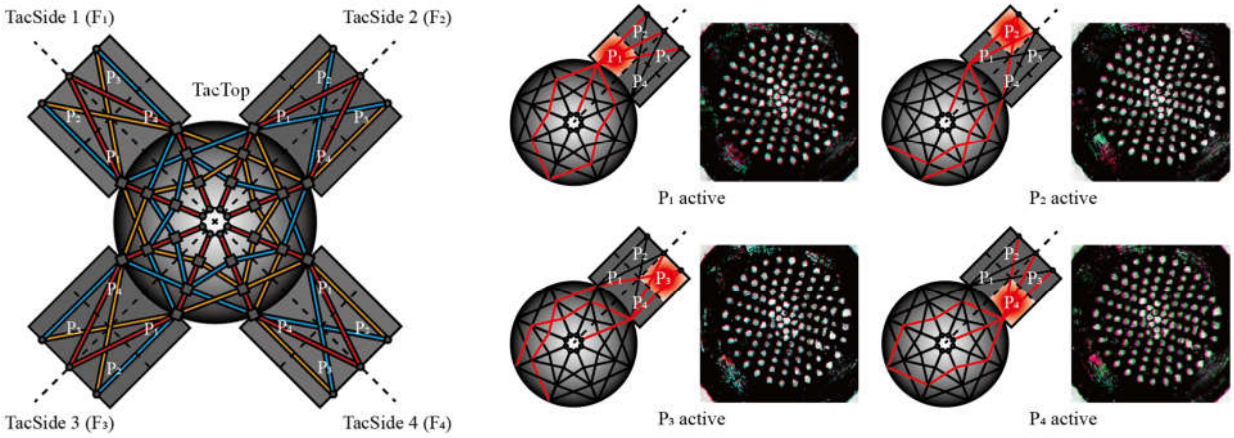
Different from 3D-printing TacTip sensors with filling gel in [4], we use 3D-printed pouring mould and polyurethane (PU) rubber (PT Flex 60, Polytek Development Corp, United States) to mould the TacTop and TacSide. The advantage of using PU rubber instead of Agilus materials is that PU rubber has a faster recovery and less viscoelasticity, stronger abrasion resistance and endurability. The white pins are made by adding a second layer of white PU rubber within the first PU rubber gelling time. Tendons are braided fishing lines (Polyethylene, 100lb, HERCULES), The white components in Fig. 3 are 3D-printed by PLA (Mega S, Anycubic, China). The camera is a medical endoscope module with 720p resolution, 10-100 mm focus range, and 4 pcs Hi-brightness white LEDs.

C. Connection of tendons

The connecting mechanism of the tendons is shown in Fig.4. Each tendon passes through the concealed membrane under the TacSides and cubes with crossing holes on the TacTop before being clamped at the center of the TacTop and fixed to the bracket by screws on the other side. Each TacSide covers three pairs of tendons, colored in blue, yellow, and red in Fig.4 (a). We weave the tendons into a small net for each TacSide that is further divided into four pieces of regions. Each region covers three tendons to generate distinct effects on the TacTop by different connection topologies. As illustrated in Fig.4 (b), we take TacSide 2 as an example to demonstrate connection of the tendons of each region and the changes in pin distribution after pressing these regions. We observe that, through the tendons, contacts on each region can be transmitted to the TacTop, and the influence spreads throughout the TacTop and further affects pin distribution under the TacTop.

D. Compare of sensitivity on the TacTop and TacSide

To quantify the difference in sensitivity between TacTop and TacSide under the same pressure, we have developed a tool consisting of a force sensor and a displacement sensor, whose signals are measured through serial port using Modbus protocol to the Laptop (in Fig. 5 (a)).



(a) (b)

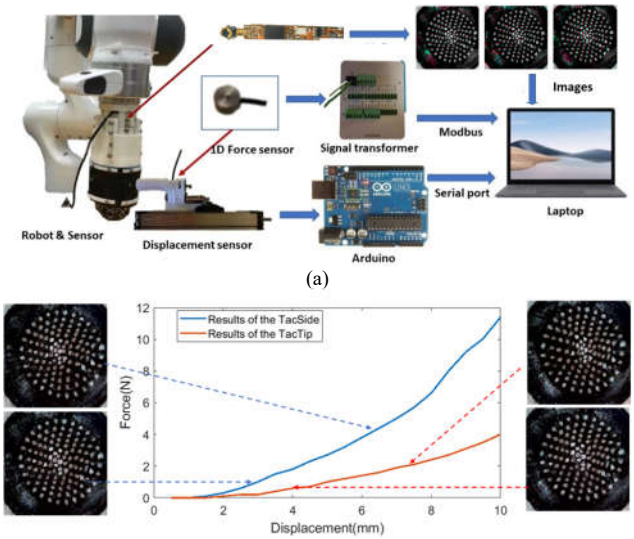
Fig. 4. Connection mode from TacSides to the TacTop. (a) Connecting topology from the TacSide (F1~F4) to the TacTop, Under each TacSide, there are three pairs of crossing tendons coloured in red, yellow and blue. The tendons are clamped at three points at the top of the TacSide on one side and fixed on the TacTop on the other side. Tendons on the TacTop are woven and coupled with each other. (b) Take TacSide 2 as an example, after touching different regions (P1 ~P4) , different regions on the TacTop will be activated.

Human operators can manipulate the slider and fix the body of the displacement sensor to apply varying degrees of pressure on the TacTop or TacSide. During this process, images, contact forces, and displacements of the slider with a pressing tool are collected. These measurements are taken while contacting the TacTop and TacSide, and are illustrated in Fig. 5(b). It can be observed that with increasing pressing depth, the force on the TacSide increases much more quickly than that on the TacTop, indicating that the TacTop is much softer than the TacSide. However, the impact of the pins' distribution on the TacSide is less obvious than on the TacTop. Therefore, contacting on the TacTop is more easily distinguished, making it more sensitive than the TacSide.

III. Position Location and Object Recognition using Improved DenseNet 121

A. Task description

The typical usage of TacTip Family sensor is to recognize the contact object in manufacturing and in-hand manipulation[7]. In this paper, we consider that the TacTop of MechTacTip has a similar structure to the TacTip Sensor and use the TacTop for object recognition. In contrast, we only use the TacSide for contact region localization since it is less sensitive and has low accuracy in object recognition. Recognitions of the TacTop and TacSide can be performed as separate tasks or cooperatively in a single task. The key challenge is how to address different perceptual challenges in a Deep Learning network. The pre-step in Deep Learning is to collect data and create a dataset. We use the embedded camera to capture the images of the distributions of pins with different contact cases. According to the perceptual requirements, we obtain the images of contact with the TacTop in different directions and depths or pressure on different areas of the four TacSides, as well as the joint effect of contact with both TacTop and one area of the TacSides. These images are cropped to the same size and assigned different names to represent tasks and contact states of MechTacTip. We elaborate on the data acquisition steps and the dataset creation process in Section V.I.



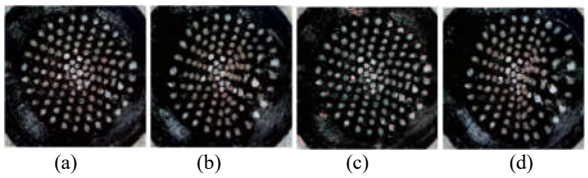
(a)

(b)

Fig. 5. Accessing the sensitivity of difference areas on the MehTacTip (a) Setup of pressing equipment with MehTacTip, displacement sensor and force sensor; (b) Tactile images and pressing force varying with depth.

B. Improved Densetwork 121 for perception and locating

First, the captured images are continuous in time, allowing us to stack three consecutive 2D images from the database to obtain colored images with temporal information. The stacked images serve as inputs to the learning framework and provide a representation of the dynamic motion of the papillae pins. Each composite 2D color image has three color channels (R, G, B).



(a) (b) (c) (d)

Fig. 6. Images of different contact situations/ (a) original image without any contact; (b) contact the TacTop; (c) contact the TacSide and the pressure is transferred through tendon and reflected on the pins' distribution, and the displacement of markers, compared with (a), are highlighted in red (d) the overlay effect of contacting both TacTop and TacSide simultaneously

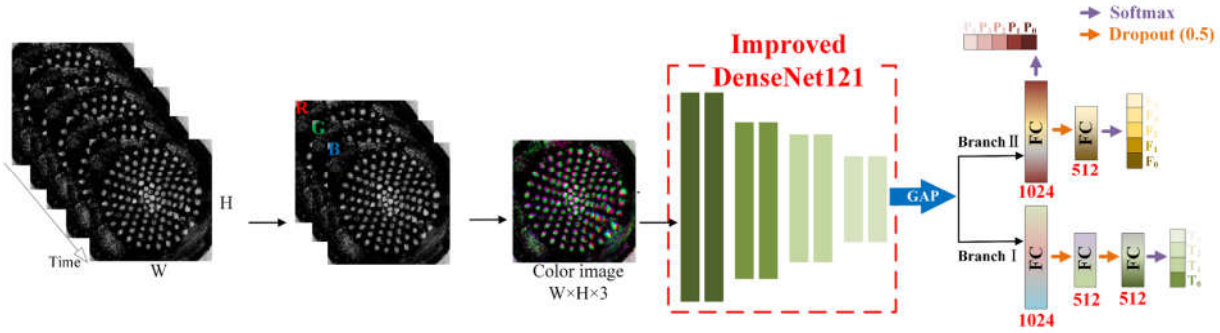


Fig. 7. Framework of improved DenseNet121 for contact state classification. First, we stack three 2D images to achieve images with temporal information and three color channels (R, G, B) for the input of the improved DenseNet121. We remove the all the fully connected (FC) layers and add a global average pooling (GAP) layer. We use two branches: Branch I is for object perception on the TacTop and Branch II is for localization of touch regions on the TacSide.

Fig. 5 shows the framework, which was achieved by improving the DenseNet121 [17]. We removed all fully connected layers (FC) and added global average pooling (GAP) at the end of the sub-network of the DenseNet121. The improved DenseNet121 is initialized from a pre-trained model on ImageNet [18]. Then, we added a GAP layer [19] to robustize the spatial translations of the extracted features. Considering that the proposed task involves two subtasks for the TacTop and the TacSide, but both represented by the same papillae pins under the TacTop, a key challenge is distinguishing the differences between two contact features.

Fig.6 depicts the tactile images of the contact on the TacTop and TacSide. As shown in Fig. 6 (c), the contact on the Tacside does not obviously affect on the distribution of pins, only in the areas highlighted in red. But, the effect is weaker compared to the direct pressure on the TacTop shown in Fig. 6 (b). So the signals generated by the TacTop can easily override the effect of the contact with the TacSide, as demonstrated in Fig. 6 (d).

To address this issue, we designed two branches after the GAP layer. Branch I is responsible for object perception of TacTop and consists of three FC layers with 1024, 512, and 512 units. Branch II is for the localization of touch regions in TacSide, which has two outputs and comprises two FC layers with 1024 and 512 units. The outputs are used to distinguish different contact areas ($F_1 \sim F_4$) and regions ($P_1 \sim P_4$) on the TacSide (the meanings of P_i and $F_i, i = 1,2,3,4$ can be found in Section V.I). This improved Densetwork121 is trained by Adam optimizer [20] using a focal loss [21] for 200 epochs. Finally, to address the issue of overfitting, we employ regularization techniques and data expansion strategies. To reduce overfitting, we introduce a dropout layer [22] with a rate of 0.5 and an L2 regularization [23] after each FC layer. Additionally, we adopt a data augmentation strategy that includes random rotation by ±1 degree, random horizontal and vertical shifts of the image's width and height by ±0.1, and random zooming within a range of ±20%.

IV. Reinforcement Learning Based on Human Demonstrations

In this section, we introduce a reinforcement learning (RL) framework for navigating continuous state-action spaces in an unknown environment using probabilistic density estimates. To approximate the density observations learned from human demonstrations, we introduce a Gaussian mixture model. This non-parametric approach enhances the generalizability of the learning results. We also evaluate the performance of four RL algorithms: DQN, Vanilla_PG, PPO_Penalty and PPO_CLIP in the same target approximation task, where the agent modifies its actions after colliding with obstacles.

A. Gaussian Mixture Model (GMM)

Gaussian Mixture Model (GMM) is a general algorithm to extract several action features from human demonstrations that consists of several probability density functions with unknown parameters in multi-dimensional space. The general probability distribution of a GMM is

$$p(\mathrm{x},\Theta) = \sum_{i=1}^{K} \alpha_i N(\mathrm{x};\mu_i,\Sigma_i) \tag{1}$$

where K represents the number of Gaussian functions, α_i is the weight of the ith Gaussian functions and the prior probability, and $N(\mathrm{x};\mu_i,\Sigma_i)$ represents the probabilistic density of the ith Gaussian distribution, in which μ_i is a mean vector and Σ_i is a covariance matrix. $\Theta = \{\{\alpha_1,\mu_1,\Sigma_1\},...,\{\alpha_K,\mu_K,\Sigma_K\}\}$ represent the whole set of uncertain parameters in (1) to be estimated by methods e.g. Expectation-Maximization (EM) [26]. Following the procedure shown in [26], for the samples $\mathrm{X}=\{x_t \,|\, t=1,...,N\}$ of demonstration, the factors μ_i, Σ_i and α_i are calculated by:

E-step:

$$w_{t,k} = \frac{\alpha_k N(x_t;\mu_k,\Sigma_k)}{\sum_{i=1}^{K} \alpha_i N(\mathrm{x};\mu_i,\Sigma_i)} \tag{2}$$

where $t = 1,...,N$ and $k = 1,...,K$.

M-step:

$$\mu_i = \sum_{t=1}^{N} w_{t,k} x_t \Big/ \sum_{t=1}^{N} w_{t,k} \ , \ \alpha_i = \frac{1}{N} \sum_{t=1}^{N} w_{t,k} \ , \text{ and}$$

$$\Sigma_i = \frac{\sum_{t=1}^{N} w_{t,k} \left(x_t - \mu_i\right)\left(x_t - \mu_i\right)^T}{\sum_{t=1}^{N} w_{t,k}} \tag{3}$$

B. Reinforcement learning

In this section, we present a RL scheme for the approaching task using MechTacTip. Unlike existing methods, like the one proposed in [24], which combines RL and GMM, our approach leverages the unique features of tactile sensors that can detect contact directions and regions. We extract actions from human demonstrations of encountering obstacles in similar directions and regions, and then use GMM to learn a probabilistic density estimate of these actions. Define $S = \{s_1, s_2, ..., s_t\}$ as the set of possible states, $A = = \{a_1, a_2, ..., a_t\}$ as the set of possible actions and $\pi(s_t, a_t | \theta)$ as the parametric policy. For this approaching task, after detecting a conflict between a region on the TacSide and an obstacle in a specific direction, the agent is enforced to take a definite action based on human demonstrations learned by GMM. The reward function will be enforced with negative feedback as the punishment either. Then the agent will explore the environment from current position until reaches the target or exhausts all the exploring steps.

In this paper, we choose three typical RL methods: DQN[25], vanilla Policy Gradient [26] and proximal policy optimization (PPO) [27] with the same training hyperparameters in TABLE I to compare the differences of the value-based, policy-based and actor-critic strategies.

TABLE I
TRAINING HYPERPARAMETERS IN RL ALGORITHMS

Parameters	Values
Optimizer	Adam
Learning rate	0.001
Weight decay	0.0001
Max steps	1000
Batch size	32

Here, the "Max steps" restricts the maximum length in an episode and "Batch size" defines the size of a mini-batch that is realized by experience replay in DQN and PPO. To compare the performance of different RL methods, we use the following three evaluation metrics:

- Mean episode reward: It measures the average cumulative reward per episode, which represents the optimization target in the training process;
- Mean episode steps: It shows how fast the agent can reach the target position, by measuring the average number of steps taken per episode;
- Mean number of collisions: It evaluates the avoiding obstacles ability of the agent, by measuring the average number of collisions per episode;

In the experiment, we will use the above metrics to compare the performance difference using three RL strategies.

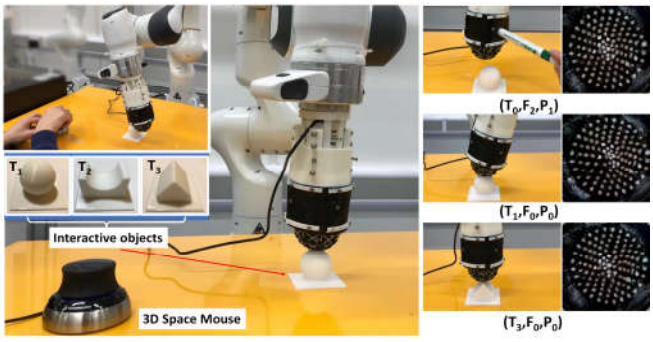


Fig. 8. Experimental system for data collection, containing a Franka robot equipped with the MechTacTip and a 3Dconnexion space mouse to teleoperate the robot arm. The left zoomed figure shows the operational process and three contact objects T1~T3, and the right three figures show the contact situations and the corresponding images of papillae pins.

V. EXPERIMENTS

In this section, two experiments are conducted to verify the effectiveness of the proposed deep learning and RL method for contact perception and obstacle avoidance. The following parts belong to two issues: Sections IV. A to IV. C deal with object recognition based on deep learning, and Sections IV. D to IV. F are for obstacle avoidance based on the RL and GMM.

A. Data collection and creating dataset

Data is collected using the experimental setup shown in Fig. 8, whcih comprises a Franka robot arm, a 3Dconnexion space mouse, and the designed MechTacTip sensor. During the data collection, one operator uses the space mouse to manipulate the TacTop to press objects T1 to T3 in Fig.8 in various positions and orientations. Concurrently, another operator uses a pen (Φ ~1.5cm) to touch different regions of the TacSide. The positions and orientations of the robot end are recorded, along with the corresponding images about the papillae pins' distributions. The tactile images of the are sampled frequency of 10 Hz.

Combining the indications of F_j and P_k, $j,k = 1,...,4$ in Fig. 4, we define a dataset as $Data\,(T_i, F_j, P_k)$, $i = 0,..,3; j,k = 0,...,4$ to classify and label the tactile images, where T_0 represents no contact with TacTop, T_1 to T_3 represent contact with the objects in the form of a sphere, a saddle and a sharp cone in Fig. 8. F_j represent a contact with the jth TacSide, P_k represent a contact with the kth region on the jth TacSide. The condition $P_k = 0$ is satisfied if $F_j = 0$, which means there is no contact on the TacSide. By setting different values of T_i, F_j and P_k, we can obtain a dataset covering all the cases of object recognition on the TacTop and contact area perception on the TacSide described in Section III.B. The number of collected images for all the situations are shown in (4), totally 10051 images.

$$Data\,(T_i, F_j, P_k) = \begin{cases} 150, & Case\,1: 1 \le i \le 3, j = k = 0 \\ 150, & Case\,2: i = 0, 1 \le j,k \le 4 \\ 150, & Case\,3: 1 \le i \le 3, 1 \le j,k \le 4 \\ 1, & Case\,4: i = j = k = 0 \end{cases} \tag{4}$$

The first two functions in (4) represent contact on the TacTop

and TacSide only. The third case represents when both TacTop and TacSide are in contact. The last case illustrates that there is no contact with MechTacTip, and only one original image is collected. We label these images using (4) to represent different contact cases and divide them into a training set (8041 images), a validation set (1005 images), and a testing set (1005 images) using the following processing steps.

1) Crop the images from their original size of 875×656 pixels to 520×520 pixels to remove the extra pixels that do not contain information about the papillae pins.
2) Downsample the cropped images to 224×224 pixels to retain only the essential information about the distribution of papillae pins for learning.
3) Normalize the pixel values by subtracting the mean of each image. This helps to reduce the effect of grey value variation across the dataset.

B. Evaluation standard

To evaluate the classification results, we choose two metrics: classification accuracy, representing the percentage of correct predictions out of total number of predictions:

$$Accuracy = \frac{Number\ of\ correction\ predictions}{Number\ of\ all\ predictions}, \qquad (5)$$

and confusion matrix, which is a cross-tabulation representing the corresponding rate between the actual classification and a prediction. An example is shown in Fig. 9, the column labels correspond to the predicted classifications while the row labels represent the true classifications. The values on the diagonal of the matrix represent the correct rate of classifications for each class. The off-diagonal values represent the misclassifications. Ideally, a confusion matrix should have diagonal values equal to one, and off-diagonal values equal to zero, indicating perfect classification. The values in the confusion matrix can be used to calculate various metrics such as the precision of model's performance.

C. Multi-purpose perception

The situations presented in (4) can be split into two cases:

1) Mono-functional perception

This case contains two perceptual tasks: object perception by the TacTop and localization by contacts on the TacSide. We use the labelled dataset in Section V. A to train the classification network in Fig.7, achieving an 98% average classification accuracy in diagonal for mono-functional perception in Fig.9. The confusion matrix involves a 16x16 square and a 3x3 square. The former lists all the situations of contacting four areas on four TacSides from F_1P_1 to F_4P_4. The classification accuracy for each situation is 100%, indicating that the features of contacting on the TacSide can be entirely separated due to the different topology of tendons woven on the surface of the TacTop. However, in the 3x3 matrix, there are some misclassifications between T1 and T3, and the prediction accuracy of T1 is only 0.86, as highlighted in the red square in Fig. 9. The mismatch is primarily due to the images taken during the initial contact phase of objects T1 and T3, as the distribution of the pins differs significantly as the press-in depth increases.

2) Multi-functional perception

Multi-functional perception aims to distinguish contact on both the TacTop and TacSide simultaneously, which is useful for tasks like online object perception and conflict localization in obstacle avoidance. The primary data used in this case is from Case 3 in (4), while Case 1 and Case 2 are treated as special cases of Case 3 to expand the training and testing dataset. The confusion matrix shown in Fig. 10 has a classification accuracy of 99% in averarge. The confusion matrix is a 19x19 square to represent the mixed perceptions of the TacTop and TacSide. Compared to Fig.9, the corresponding rate between T1 and T3 increases to 1, indicating that contacts on objects T1 and T3 can be distinguished using multifunctional perception. However, some misclassifications still occur, as indicated by the red square in Fig.10 with a value of 0.04, which is mainly affected by the TacTop's object perception of class T1.

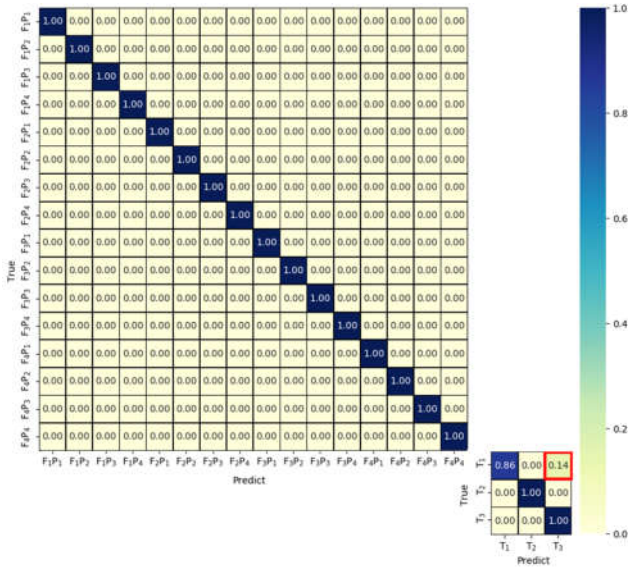


Fig. 9. Confusion matrix of mono-functional perception.

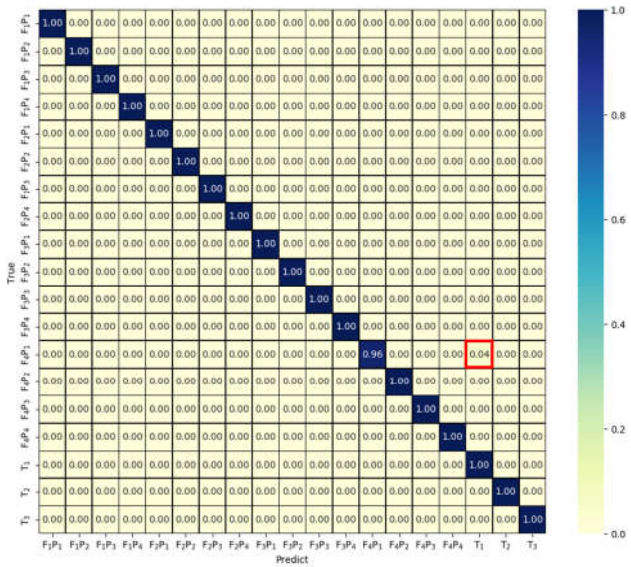


Fig. 10. Confusion matrix of multi-functional perception.

D. Human demonstration system

Section V. A to V. C utilize the MechTacTip sensor to realize perception detection. Considering collecting accurate data and performing human demonstrations in a virtual environment is more feasible than on the physical platform, a parallel system to the real platform is established for human demonstrations, as depicted in Fig.11. The platform involves a Touch phantom joystick and a laptop for interaction and simulation. The blue lines in Fig.11 denotes signals transmitted from the human side through the haptic device to interact with the virtual robot, and red lines represents force rendering of robot movement and the conflict via the haptic device and a simple interaction interface. Therefore, the interface can provide force and visual feedback to the operators and display the start, target, and vivid trajectory of the agent. Finally, the movement of both humans and the virtual robot, as well as the conflict state, are recorded with time information.

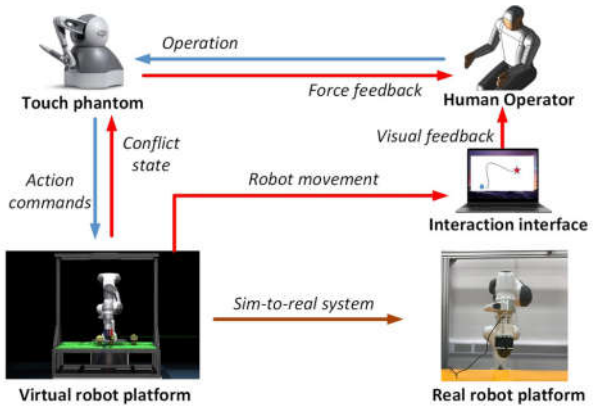


Fig. 11. Human demonstration platform.

E. Virtual environment building

The virtual manipulation environment replicates the platform in real, encompassing URDF models of the Franka robot, the tactile sensor, the obstacles, and a robot manipulation platform in the MuJoCo physics engine. The robot is controlled using 'mocap control', an interface provided by the MoJuCo simulator that computes the inverse kinematics of the robot's position and orientation. Furthermore, we utilize the OpenAI Gym, a Python library offers an API for reinforcement learning, to encapsulate the experimental environment in Fig. 12.

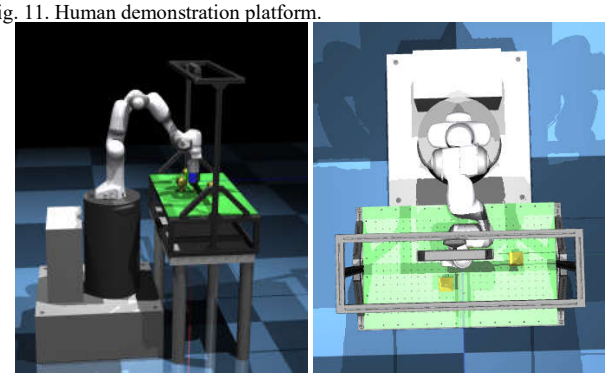


Fig. 12. Virtual training environment built in the MuJoCo physics engine

The settings of RL in the Gym are:

- Action_space: Actions are executed based on current state and conflict status. For the next step, we choose a random offset from the current position and enforce a specific action learned from GMM after colliding with obstacles.
- Observation_space: The agent can only observe the target, start, current position, and contact force, as the positions and sizes of the obstacles are unknown. We set eight force -sensing units on the surface of the MechTacTip to record force values and directions as the force sensor in MuJoCo that can only detect one-dimensional signals.
- Reward: we consider three types of rewards in the task: a large positive reward is granted if the agent reaches the goal, a negative reward is given if the agent collides with obstacles, and a zero reward is given otherwise.
- Step: update the MuJoCo environment and then transmit the state information to the Gym environment.

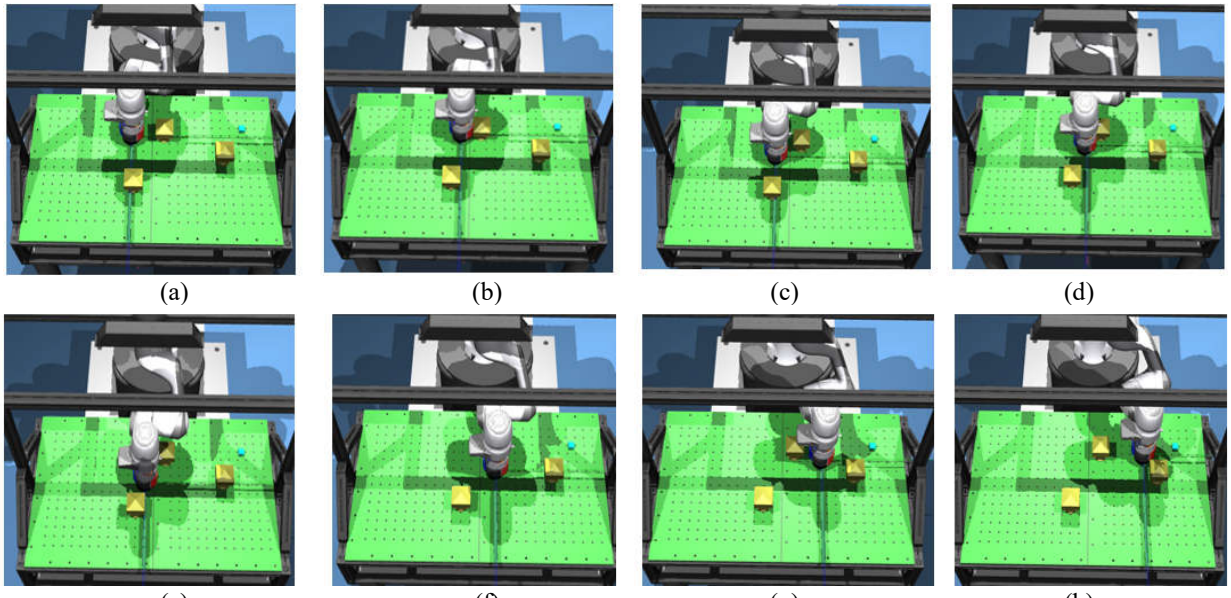


Fig. 13. Application of the RL and the developed MechTacTip sensor used in a collision-avoidance task

F. Simulation of obstacle avoidance

The experiment is conducted using the settings specified in Section V.E and the parameters and algorithms outlined in Section IV.B. The experiment is taken on a system with Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz, NVIDIA Geforce RTX2060 6GB, 12GB RAM, Ubuntu 18.04 LTS with Tensorflow 1.14 Framework. We compare the performance of the DQN, vanilla Policy Gradient (Vanilla_PG), PPO with adaptive KL penalty (PPO_ PENALTY) and PPO with clip function (PPO_CLIP). The curves of the exponential average training rewards are shown in Fig.14 and the evaluations based on the metrics in Section IV.B are shown in Table II.

The results in Table II indicate that the two PPO methods achieve the highest episode rewards, while the Vanilla_PG method yields the lowest value. In addition to the reward, the two PPO methods outperform the value-based and strategy-based methods in terms of the minimum episode steps and number of collisions. Fig. 15 compares the results obtained using different action spaces with the same PPO_CLIP method. It can be observed that training reward significantly improves when we incorporate the actions that are learned from human demonstrations, as opposed to using only continuous actions. Finally, we apply the learned network to a goal achievement task in a virtual environment, and the internal steps are depicted in Fig. 13.

TABLE II
COMPARE OF DIFFERENT RL LEARNING STRATEGIES

Algorithms	Mean episode reward	Mean episode steps	Mean number of collisions
DQN	6.74	342	6.55
Vanilla_PG	5.23	487	7.21
PPO_PENALTY	8.55	**248**	3.41
PPO_CLIP	**8.67**	259	**3.24**

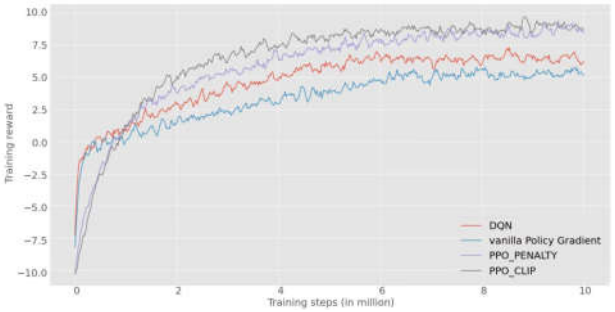


Fig. 14. Exponential average training rewards of different algorithms.

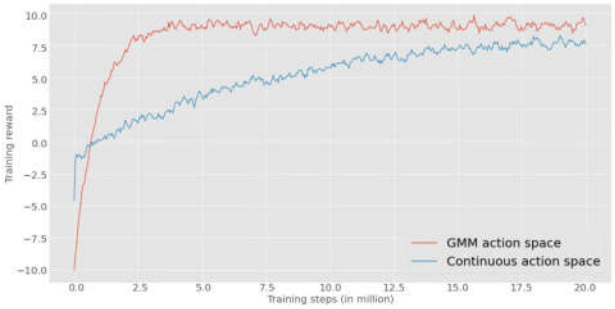


Fig. 15. Exponential average training reward of PPO_CLIP with different action space

VI. CONCLUSION

In this paper, a new optical tactile sensor - MechTacTip - is developed for tactile perception and contact position detection. The new sensor employs tendons to extend the sensing range of traditional tactile sensors, even outside of view field, and then transmit contact signals to TacTop via the distribution of pins. By utilizing this unique structure, MechTacTip can serve dual purposes of object detection and contact position localization on TacSide. However, overlapping signals generated by direct contact on the TacTop and indirect pulling on the TacSide, along with non-obvious changes in the pin distribution, pose challenges for accurate signal classification. To address this problem, We improve DenseNet121 network by removing the full connect layer and building a two-branch structure after a global average pooling layer connected to the subnetwork of DenseNet121 network. The experiments based on a dataset of over 10k images demonstrate that the proposed network could achieve high classification accuracy.

With accurate conflict localization, we apply the developed tactile sensor to a target reaching task in an unknown multiple obstacle environment. A simulation environment same to the real robot platform is built based on MuJoCo Physical Engine and OpenAI Gym. A human operator interacts with the virtual agent through a Touch Phantom joystick and an interaction interface that provides force and visual feedback.

Human demonstrations are recorded and further processed by using GMM to generate reactive skills for different contact situations, which are then used to create RL's action space for the designed task. We compare the performance of three typical RL strategies - DQN, Vanilla Policy Gradient, and Proximal Policy Optimization (PPO) - with the same configurations and hyperparameters and find that PPO achieves the best results using the proposed three metrics for the proposed task. Finally, we verify the effectiveness of the proposed framework through simulations in a MuJoCo environment.

REFERENCES

[1] G. J. Garcia, J. A. Corrales, J. Pomares, and F. Torres, “Survey of visual and force/tactile control of robots for physical interaction in Spain,” Sensors, vol.9, no.12, pp.9689-9733, 2009.

[2] W. Yuan, S. Dong, and E. H. Adelson, “Gelsight: High-resolution robot tactile sensors for estimating geometry and force,” Sensors, vol.17, no.12, p.2762, 2017.

[3] C. Abad and A. Ranasinghe, "Visuotactile Sensors With Emphasis on GelSight Sensor: A Review," IEEE Sensors Journal, vol. 20, no. 14, pp. 7628-7638, 2020.

[4] M. Lambeta et al., "DIGIT: A novel design for a low-cost compact high -resolution tactile sensor with application to in-hand manipulation, " IEEE Robotics and Automation Letters, vol. 5, no. 3, pp. 3838-3845, 2020.

[5] S. Wang, M. Lambeta, P. -W. Chou and R. Calandra, "TACTO: A Fast, Flexible, and Open-Source Simulator for High-Resolution Vision-Based Tactile Sensors," IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 3930-3937, 2022.

[6] N. F. Lepora, Y. Lin, B. Money-Coomes and J. Lloyd, "DigiTac: A DIGIT-TacTip Hybrid Tactile Sensor for Comparing Low-Cost High-Resolution Robot Touch," IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 9382-9388, 2022.

[7] B. Ward-Cherrier, N. Pestell, L. Cramphorn, B. Winstone, M. E. Giannaccini, J. Rossiter, and N. F. Lepora, “The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies,” Soft robotics, vol. 5, no.2, pp.216-227. 2018.

[8] G. Soter, A. Conn, H. Hauser, N.F. Lepora, and J. Rossiter, “MultiTip: A multimodal mechano-thermal soft fingertip,” in IEEE International Conference on Soft Robotics (RoboSoft) , pp. 239-244. 2018.

[9] Yamaguchi and C. G. Atkeson, “Combining finger vision and optical tactile sensing: Reducing and handling errors while cutting vegetables, ” in 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), 2016, pp. 1045-1051

[10] N. F. Lepora and J. Lloyd, "Pose-Based Tactile Servoing: Controlled Soft Touch Using Deep Learning," IEEE Robotics & Automation Magazine, vol. 28, no. 4, pp. 43-55, 2021.

[11] M. Floriano Vázquez and N. F. Lepora, "Uncertainty-aware deep learning for robot touch: Application to Bayesian tactile servo control," in 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 1615-1621.

[12] N. F. Lepora and J. Lloyd, "Optimal Deep Learning for Robot Touch: Training Accurate Pose Models of 3D Surfaces and Edges," in IEEE Robotics & Automation Magazine, vol. 27, no. 2, pp. 66-77, 2020.

[13] Yamaguchi and C. G. Atkeson, “Combining finger vision and optical tactile sensing: Reducing and handling errors while cutting vegetables,” in IEEE-RAS International Conference on Humanoid Robots (Humanoids), 2016, pp. 1045–1051

[14] Winstone, T. Pipe, C. Melhuish, S. Dogramadzi, and M. Callaway, “Biomimetic tactile sensing capsule,” in Conference on Biomimetic and Biohybrid Systems. Springer, 2015, pp. 113–122.

[15] Ma, E. Donlon, S. Dong and A. Rodriguez, "Dense Tactile Force Estimation using GelSlim and inverse FEM," 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 5418-5424

[16] Wang, S., She, Y., Romero, B. and Adelson, E., 2021, May. Gelsight wedge: Measuring high-resolution 3d contact geometry with a compact robot finger. In 2021 IEEE International Conference on Robotics and Automation (ICRA) (pp. 6468-6475). IEEE.

[17] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in IEEE Conf. Comput Vis. Pattern Recog., 2017, pp. 4700–4708

[18] Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classifica- tion with deep convolutional neural networks,” Adv. Neural Inform. Process. Syst., vol. 25, 2012

[19] M. Lin, Q. Chen, and S. Yan, “Network in network,” arXiv preprint arXiv:1312.4400, 2013.

[20] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

[21] T.-Y. Lin, P. Goyal, R. Girshick, et al., “Focal loss for dense object detection,” in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.

[22] P. Baldi and P. J. Sadowski, “Understanding dropout,” Adv. Neural Inform. Process. Syst., vol. 26, 2013.

[23] C. Cortes, M. Mohri, and A. Rostamizadeh, “L2 regularization for learning kernels,” arXiv preprint arXiv:1205.2653, 2012

[24] Agostini, A. and Celaya, E., 2010, July. Reinforcement learning with a Gaussian mixture model. In The 2010 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[25] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, nature, vol. 518, no. 7540 2015, pp. 529–533, 2015

[26] J. Peters, and S. Schaal, Reinforcement learning of motor skills with policy gradients. Neural networks, vol.21, no. 4, pp.682-697. 2008.

[27] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347 2017, 2017