Navigation of Tendon-driven Flexible Robotic Endoscope through Deep Reinforcement Learning

Chikit Ng[1], Huxin Gao[1], Tianao Ren[2], Jiewen Lai[1], Hongliang Ren[1]

***Abstract*— Robotic endoscopes play a crucial role in diagnosing gastrointestinal disease and performing tumor resections. While current research primarily focuses on autonomously controlling rigid robots, establishing control models for flexible robots remains challenging. To address this, model-free deep reinforcement learning (DRL) presents a promising approach for enabling agents to make decisions under uncertainty. In this paper, we investigate the control policy of flexible endoscope using Simulation Open Framework Architecture (SOFA) platform. We design a flexible tendon-driven robotic endoscope (TDRE) and develop a custom simulation environment within SOFA to train DRL agents. Our approach involves implementing the Proximal Policy Optimization (PPO) algorithm to approximate an optimal policy for trajectory planning. The optimal policy facilitates trajectory tracking tasks for the TDRE's end-effector, such as circle trajectories and action disturbances, without requiring fine-tuning policy network parameters. Experimental results demonstrate that our approach achieves near real-time performance (30 FPS). The feedforward neural network of the policy provides feedback, enabling closed-loop control of TDRE. Furthermore, our experiments show that the navigation success rate of TDRE exceeds 90% within a tolerant error of 3 mm in free space. Notably, compared direct training with contact, navigation task with contact retrained by pre-trained policy in free space exhibit enhanced navigation capabilities.**

I. INTRODUCTION

The widespread use of long flexible endoscopes in accessing and diagnosing hollow organs in the gastrointestinal (GI) tract underscores their importance in medical procedures. These flexible catheters, akin to transoral robots, navigate narrow spaces within the GI tract for diagnostic and therapeutic purposes [1], [2], [3], [4]. However, relying solely on preoperative information, like CT imaging, for developing path-planning algorithms is unreliable due to dynamic biological deformations and body contact between robots and tissues. In this paper, we consider both the dynamic behavior in trajectory planning and control method, and the factor of contacting environment.

Navigation is crucial for executing complex surgical tasks with endoscopes, yet autonomous navigation of flexible robots in soft tissues remains a challenge [5], [6]. We modeled the deformation of a flexible robot by finite element method (FEM) to balance processing speed and accuracy in approximating deformations [7], [8]. To avoid the impracticality of manually tuning hyperparameters for traditional nonlinear control systems and reduce time and cost associated with trial-and-error in real-world scenarios, using reinforcement learning in simulation to train the control policy of robot is a more efficient approach than model-based control [9]. The robot interacts with environments in SOFA and derives optimal control strategies through on-policy DRL training [10].

Current research primarily focuses on autonomously controlling rigid robots [11], [12], [13]. The key challenge lies in devising an appropriate control policy for flexible robots amidst both the deformation of the robot and the high variability and dynamic behavior of soft tissues [14].

Our simulated endoscope, equipped with flexible tendon-driven catheters, aims to learn the optimal trajectory for navigating on the inner wall of a simplified stomach model. To achieve autonomous navigation goals, we demonstrate the versatility of our approach by realizing random target navigation in free space, assessing the robustness of the trained control policy under disturbed actions. We deployed a retrain strategy to enhance the capability of navigation with contact in the environment.

To our best knowledge, it is the first attempt at using model-free DRL algorithms to control tendon-driven flexible robot in contact scenarios.

II. PROBLEM FORMATION

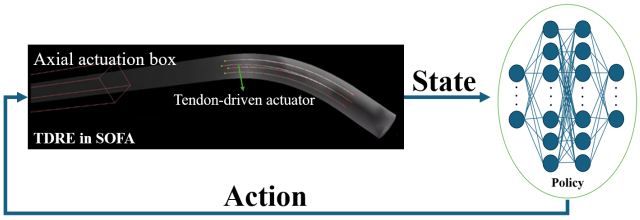

Fig. 1. Learning-based feedback control strategy. Pre-train Neural Network works as the brain of TDRE, the input is the instant state of robot. It provides an optimal action set for each time step to reach the destination.

Navigation in endoscopy procedures in the GI tract relies on controlling degrees of freedom and reacting to collisions with the organ's inner wall to reach target areas. A more uniform collision can enhance safety by reducing the endoscope's pressure on the organ surface [7], [15]. We anticipate that TDRE can learn environmental information through trial and error using model-free DRL. By leveraging

This work was supported by Hong Kong Research Grants Council (RGC) Collaborative Research Funds (CRF C4026-21G) and HK RGC Research Impact Fund (RIF R4020-22).

[1]C. Ng, H. Gao, J. Lai, H. Ren are with the Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong, China hlren@ee.cuhk.edu.hk (Corresponding Author: Hongliang Ren)

[2]T. Ren is with the Department of Mechanical Engineering, Stanford University, CA, USA

Manuscript 43 submitted to 2024 IEEE International Conference on Advanced Robotics and Its Social Impacts (ARSO). Received February 16, 2024.

this information, TDRE aims to approach unreachable target areas of the organ wall, utilizing front-end tendon-driven actuation, back-end reaction force, and swinging higher in free space like a pendulum to reach the maximum bending inaccessible region.

The algorithms of RL are used to find an optimal policy and predict action pairs. For modeling decision-making problems, the Markov Decision Process (MDP) is a framework for formulating problems in RL.

The problem of navigation for TDRE is a finite-horizon discounted MDP. MDP consists of five components, $< S, A, R, P, \gamma >$, where $s_t \in S$ is the state with respect to time t, $a_t \in A$ is the action in time t, $r_t = R(s_t, a_t)$ is the reward function. Given state transition function P, $P(s_{t+1}|s_t, a_t)$ is the probability for the state in the next time step under the current state-action pair.

Neural net-based DRL approaches can provide predictors of dynamics [9]. A trained agent can perform real-time action selection by forward passing observation state input to a pre-trained Neural Network. As shown in Fig. 1, the feedback control strategy can select an optimal action set for instant observation.

III. Methodology

A. Computer-aided Design

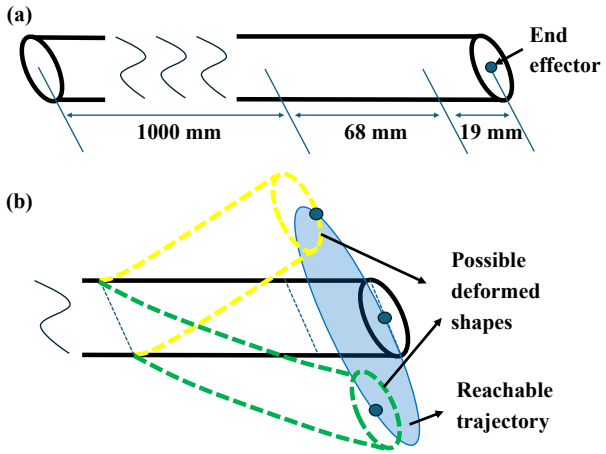

Fig. 2. Computer-aided design for TDRE. (a) TDRE consists of three parts: the back end (1000 mm), the tendon-driven part (68 mm), and the front end (19 mm). (b) Tendon-driven bending capability. Assumes that the back end is fixed and only the tendon drive is considered. The blue-shaded trajectory shows the reachable trajectory of the end effector of TDRE. Green and yellow dotted lines illustrate possible deformed shapes of TDRE.

1) Mechanical Structure: The endoscopic robot is designed to have 3 parts as shown in Fig. 2(a). The back end is the longest part that passively interacts with the environment. The length of this part is 1000 mm. The second part is the tendon-actuated part that can be controlled by peripheral devices, such as using a keyboard in simulation. In reality, it is expected to bend the front end of the robot, based on the control from the forces executed on the back end and actively controlling with tendon actuators. In SOFA, cable-driven module is applied to simulate the behavior of tendon [8]. The length of this part is 68 mm. The third part is on the rightmost session. The length of this part is 19 mm. This part is reserved to add functionality on the top of the endoscope. The most popular application is to attach a camera and use rigid material in this part.

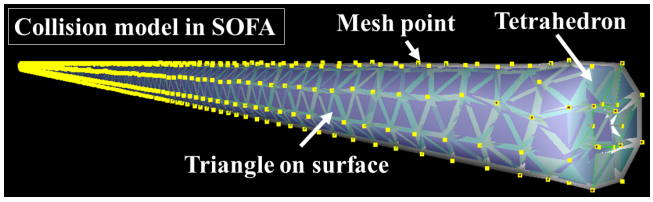

Fig. 3. Mesh generation for simulating deformable object in SOFA by Finite Element Method. The yellow dots represent points in the topology container in SOFA.

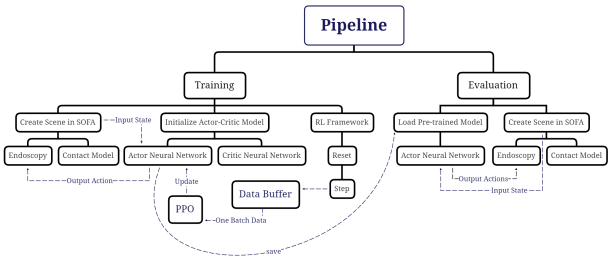

Fig. 4. The DRL training and evaluation pipelines in SOFA.

2) Mesh Generation: The collision model of TDRE in SOFA has a cylindrical shape as shown in Fig. 3. To simulate its deformable behavior by Finite Element Method (FEM), the mesh is generated by Gmsh [16]. To accurately simulate the deformable behavior of TDRE, we generated 1920 points. There are 3540 triangles on the surface and 4728 tetrahedra in the cylinder. We set Young's Module = 1000 Nm^{-2} and Poisson's Ratio = 0.1 for using the module of tetrahedron FEM forcefield in SOFA.

3) Tendon-driven Actuators Design: The diameter of the surface of the robot tip is 11.8 mm. We designed four cable-driven actuators. In SOFA, the flexible endoscopic robot will twist its direction from the central axis because the cable length can be modified by inputting the value index to decrease the length of the cable.

B. DRL

State-of-the-art DRL algorithms, including PPO, Trust Region Policy Optimization (TRPO), Deep Deterministic Policy Gradient (DDPG), Twin Delayed DDPG (TD3), and Soft Actor-Critic (SAC), have been evaluated for continuous control tasks in [17]. Among these, PPO demonstrates consistent performance across tasks, leading to its selection for training our agents.

Reward R_t is calculated immediately after each action has been executed to TDRE. To make immediate rewards be

more significant, the discount factor $\gamma \in (0,1)$ is introduced. The expected discounted episode reward

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots = \sum_{k=0}^{T} \gamma^k r_{t+k}. \quad (1)$$

The goal for agents in DRL training is to take actions to maximize G_t.

We deploy an on-policy DRL algorithm PPO. It collects a set of roll-out data C, stored as a sequence

$$Q = \{S_t, A_t, R_t, S_{t+1}, A_{t+1}, R_{t+1}, \ldots\}. \quad (2)$$

The policy function π_θ is optimized by PPO to maximize the expected return, which is an optimization problem,

$$\max_\theta F(\theta) = \max_\theta E_{\pi_\theta}(G_t | s_t = s). \quad (3)$$

The trajectory planning for navigation tasks are end-to-end DRL approach in simulation scenarios. We want the end-effector of TDRE to autonomously process the state information obtained from the SOFA environment and adjust the posture for reaching targets. The goal is to reach the target as close as possible with the smallest number of steps.

Two neural networks $\pi_\theta(A_t|S_t)$ and $V_\phi(S_t)$ are used to approximate the functions of actor (policy function) and critic (value function) respectively. $V_{\theta(s_t)}$ is an estimate state value from $\pi_\theta(A_t|S_t)$ to determine whether the output action set is commending.

We deploy PPO-Clip to maximize the expected return [18]. The clipping coefficient C_c is set to prevent large policy changes for one update process. For the value function, we encourage exploration by penalizing overly deterministic policies. The loss functions for $\pi_\theta(A_t|S_t)$ and $V_\phi(S_t)$ are clipped surrogate objective $L_t(\theta)$ and entropy regularization, where A_t is the estimator of the advantage function at timestep t. $L_t(\theta)$ is defined as

$$L_t(\theta) = \min(r_t(\theta)A_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)A_t. \quad (4)$$

1) State: The state set S_t in continuous state space can be represented as

$$S_t = \{P_e, C_t, M, P_t, S_e\}, \quad (5)$$

where $P_e = (X_e, Y_e, Z_e)$ is the current positions of end-effector, $C_t = C_{t1}, C_{t2}, C_{t3}, C_{t4}$ is the values of constraints of the tendons, M is the value of axial motion of last time step and $P_t = (X_t, Y_t, Z_t)$ is the target position. S_e is the number of steps of the current episode.

2) Action: TDRE has two types of actuation: axial forward, and tendon-driven actuation. TDRE can make one DoF of axial forward motion, and four DoFs of tendon actuation. Values of five actions are defined in an action set

$$A_t = \{C_{t1}, C_{t2}, C_{t3}, C_{t4}, M\}. \quad (6)$$

The actions are continuous. In each time step, the output values of $\pi_\theta(A_t|S_t)$ are constrained from -1 to 1 by adding hyperbolic tangent activation function $tanh^{-1}$ in the final layer. The values in A are sampled by multivariate normal distribution, whose mean is from the vector output of $\pi_\theta(A_t|S_t)$. One variance value V_a is set as a hyperparameter to make sure the agent can explore the environment in the training process. To simulate the bending limit of front-end tendon-driven behavior as a real endoscope, we limit the accumulated constraint value for each tendon in between -7 and 7 in simulation.

3) Reward: The reward function R consists of three factors. When the distance between the end-effector and target point is within the predefined threshold distance, TDRE finishes this episode and gains a fixed reward R_f. R_f is the most significant factor. We set a negative reward R_d denoting the two-point distance of P_e and P_t. R_e is a punishment reward for the number of steps of one episode. The more steps have been taken, the more negative value is set to the reward of one action. Hence, the reward function is represented as

$$R = R_f - \mu_1 R_d - \mu_2 R_e. \quad (7)$$

C. Pipelines

Training and evaluation pipelines are shown in Fig. 4. The Main Scene contains object nodes in SOFA, a physical engine for DRL training and evaluation, providing real-time updated object parameters in SOFA. Our designed TDRE is the agent for DRL. Additional models can be loaded as an object node to provide contact with TDRE. Reinforcement learning class (RL class) includes compulsory components (Reset function and Step function) for RL training [19]. After finishing training, the trained actor Neural Network is saved and it is loaded as a feedforward Neural Network for evaluation.

IV. Experiment

We conducted two experiments that followed the training and evaluation pipelines as shown in Fig. 4. The first experiment is to investigate the effect of action error disturbance for navigation tasks. The second experiment is to demonstrate the ability of the trained model to conduct a pre-planning trajectory task and its navigation capability in an environment with conduct.

Based on the interaction between the endoscopic robots TDRE and constructed environments of free space (E_f) and simplified model (E_s), the data collected from each step is transmitted as input data for training and evaluation. Agents can learn from the data and optimize the weights of neural networks by clip-PPO.

Using a feedforward network of trained policy to output action vectors, it can perform navigation tasks in real-time (near 30 FPS). All the simulation experiments, including DRL training and evaluations in SOFA, ran on 20 cores 12^{th} Gen Intel(r) Core(TM) i7-12700K CPU.

A. Disturbance for Actuators

We trained the policy $\pi^o_\theta(A_t|S_t)$ for 52634 episodes in environment E_{f1} as shown in Fig. 5(a). We define maximum disturbance error (MDE) as the maximum difference between each output value from $\pi^1_\theta(A_t|S_t)$ and the input value to actuate TDRE. For example, the values in action set A_t are

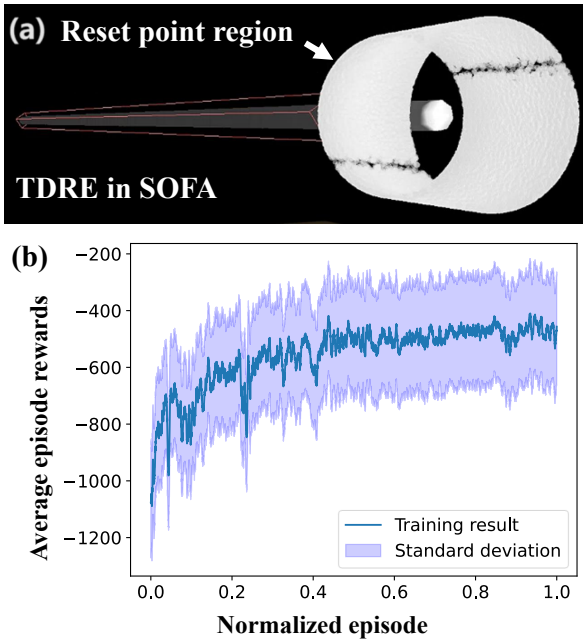

Fig. 5. (a) The training and evaluation environment for experiment 1. The white side surface of the cylinder is the reset target region for each episode. The red rectangular box mimics the actuation box in reality, so that only axial movement is allowed in the box, neither deformable behavior nor swinging effect by inertia. (b) The training result of policy $\pi_\theta^1(A_t|S_t)$, it converges at very begining. We trained for 52634 episodes.

constrained in the range between -1 and 1. If MDE = 10%, the errors that are randomly sampled from -0.1 to 0.1 are added to all the values in A_t. To investigate MDE, the reset target region is the side surface of the cylinder, the white region as shown in Fig. 5(a). All the reset target points are reachable without the swinging effect by inertia. The red rectangular box mimics the actuation box in reality which provides the axial movement. The part of TDRE in the red box neither has deformable behavior nor makes a swinging effect by inertia.

We defined tolerant error (TE) that the navigation task is treated as success within TE. For each point, TDRE can take a maximum of 128 steps. If the step is over 128, the task is treated as fail. Fig. 6(a) indicates that when TE exceeds 3 mm, the success rates of navigation tasks are over 90%, regardless of the presence of disturbance. Fig. 6(b) shows the point density of distance of the final step (DFS) under different maximum disturbance for actions (MDA). We tested 5000 points with a maximum step of 128 when MDA = 15%, 30%, 45% and 60%.

The results in Fig. 6(a-b) demonstrate the robustness of the DRL control strategy for tendon-driven flexible robot, even when there is a discrepancy between the action value output by the model and the actual value required for actuation in reality. Despite this discrepancy, with accurate state information S_t provided by Eq. (5), the control policy is capable of generating an optimal trajectory to effectively approach the target in the subsequent steps.

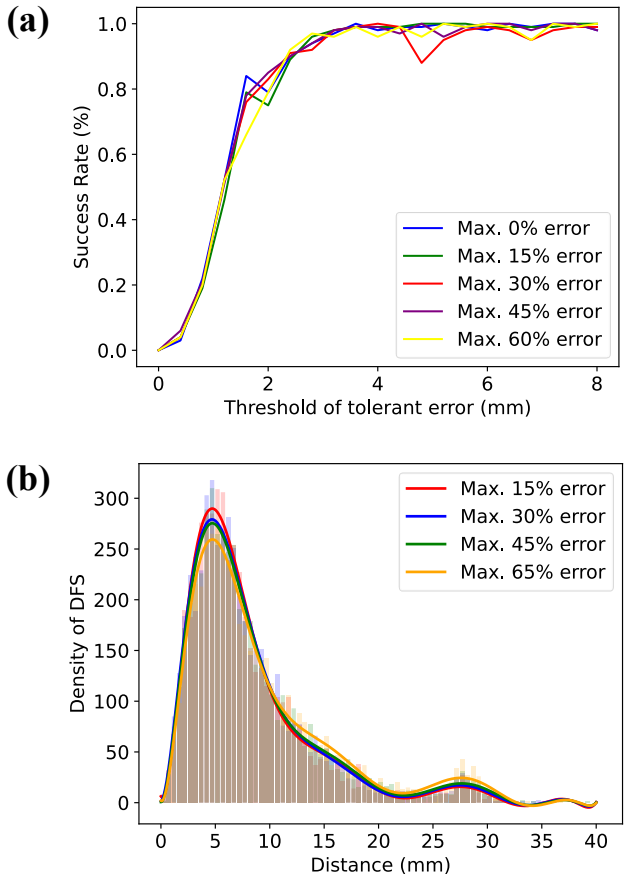

Fig. 6. Evaluation of control policy $\pi_\theta^1(A_t|S_t)$. (a) The success rates of navigation with actions with different maximum noise. (b) The points density of distance of the final step (DFS) under different maximum disturbances for actions. We tested 5000 points with a maximum step of 128 for each case.

B. Navigation Task with Pre-planning Trajectory and in Environment with Contact

We trained three policies with the same reward settings. We set the parameters $R_f = 1000$, $\mu_1 = 8$ and $\mu_2 = 1$ as shown in Eq. 7. The hyperparameters of Neural Networks are the same for three policies. Neural networks consist of two fully connected layers for both actor and critic. The first layer and second layer for the actor neural network contain 128 and 64 neurons respectively. The two layers for critic neural networks contain 256 neurons. The batch size for updating the weights of neural networks = 512 and learning rates = 0.0003.

Fig. 7(a) shows the training result of $\pi_\theta^2(A_t|S_t)$, which is trained in E_{f2}, as shown in Fig. 7(e). The yellow cylinder is a reachable region without swinging by damping action actuated by cables. The white sphere region is a region for resetting target points.

Fig. 8 shows the pre-planning trajectory navigation of TDRE by using pre-train control policy $\pi_\theta^2(A_t|S_t)$. We set a circle trajectory containing 20 green points with radius = 50 mm. Red points indicate the real trajectory. The points

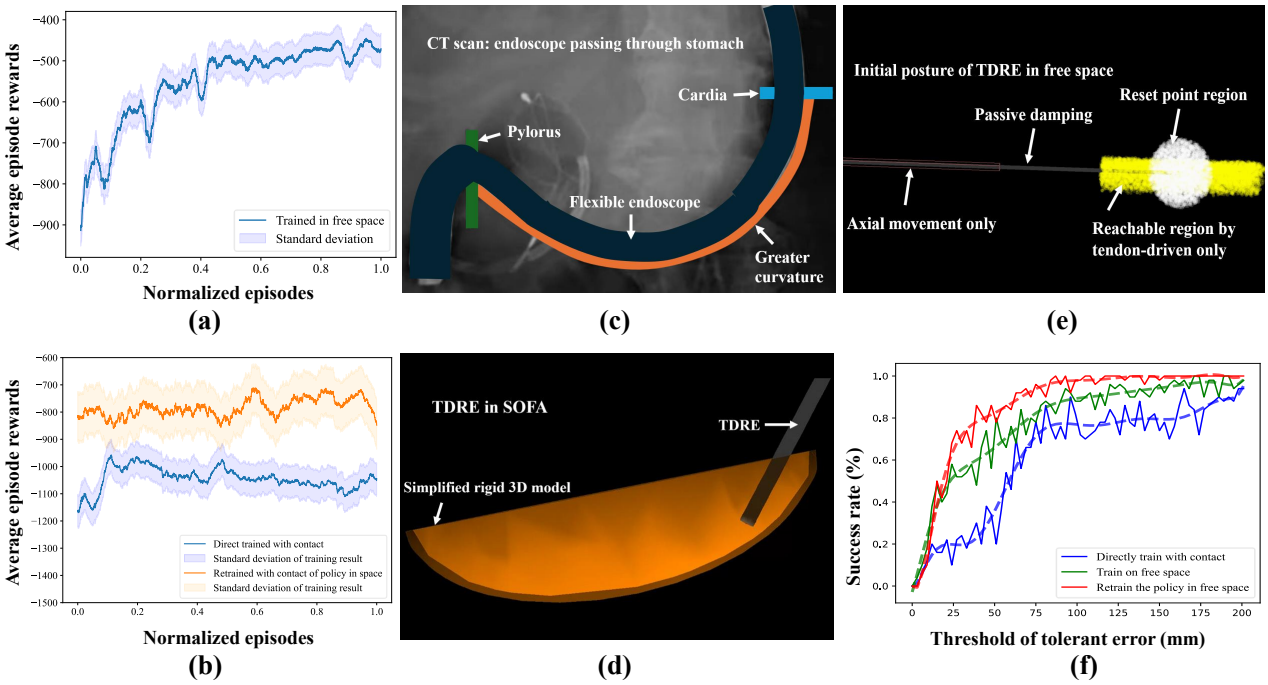

Fig. 7. (a)-(b) training results. The shaded regions are the standard deviation of samples. The curve is using an average window with size of 200 for the convolution process. (a) The training result of $\pi_\theta^2(A_t|S_t)$ in E_{f2}. (b) The training results of $\pi_\theta^3(A_t|S_t)$ and $\pi_\theta^4(A_t|S_t)$ in E_s. (c) The CT scan of the endoscope passes through a stomach in reality. (d) The simplified rigid model in environment E_s mimics the inner wall structure of the stomach, which provides collision and contact for TDRE. The model is in three-dimensional space. We make it transparent for better visualization. (f) Evaluation of control policy $\pi_\theta^2(A_t|S_t)$, $\pi_\theta^3(A_t|S_t)$ and $\pi_\theta^4(A_t|S_t)$ in E_s.

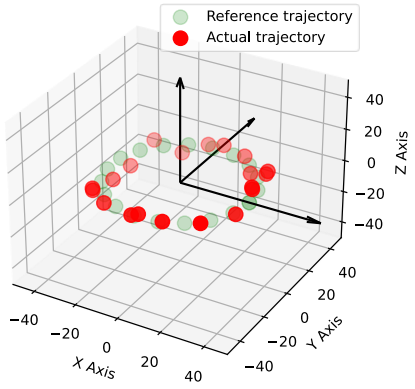

Fig. 8. Green points are pre-planning circle trajectory and red points are the trajectories of TDRE.

of end-effector coincide roughly with points of pre-planning trajectory because we trained the policy with TE = 20 mm.

Fig. 7(b) shows the training results of $\pi_\theta^3(A_t|S_t)$ and $\pi_\theta^4(A_t|S_t)$, which are trained in E_s, as shown in Fig. 7(d). $\pi_\theta^3(A_t|S_t)$ is retrained policy of $\pi_\theta^2(A_t|S_t)$ in E_s.

Fig 7(c) shows a CT of a real endoscope passing through the stomach. The orange curve indicates the inner wall of the stomach that contacts with a flexible endoscope. The green line indicates the gastric outlet and the blue line stands for the starting point where the endoscope collides with the inner wall of the stomach. Fig. 7(d) showcases the initial posture of TDRE for training and evaluation. The rigid model is a simplified model of the stomach that can represent the similar structure of the inner wall.

In the evaluation process, 50 target points are randomly chosen from the points on the surface of the simplified model for each TE. Fig. 7(f) shows the successful rates for reaching the targets within corresponding TEs for $\pi_\theta^2(A_t|S_t)$, $\pi_\theta^3(A_t|S_t)$ and $\pi_\theta^4(A_t|S_t)$ in E_s.

Loading a pretrained model to train the model in a new environment is our retrain strategy, which showcases the best performance, followed by a policy trained in free space. Directly training on the environment with contact gives the worst performance. E_s is a more complicated environment, compared with the environment in free space (E_{f1} or E_{f2}). TDRE is supposed to learn how to approach the target by walking along the inner wall of the simplified rigid model. Retrain strategy helps TDRE learn how to approach targets in free space and learn specific information such as how the contacted model can guide TDRE to targets by retraining the pre-trained model.

V. CONCLUSIONS

This paper introduces an autonomous control strategy for flexible robots using DRL. We evaluate the effectiveness of the trained policies through navigation missions and pre-

planned circle-drawing tasks. Specifically, our model-free control policy achieves a success rate exceeding 90% in free space experiments within a clinical tolerance error of 3 mm. Furthermore, by introducing action disturbance errors to the agent, we demonstrate its robustness in accurately reaching the target.

However, we observe that the navigation capability of the robot is significantly affected by contact and collision scenarios. To address this challenge, we advocate for training the agent in environments that closely mimic real-world contact and collision conditions and deploy retrain strategy, thereby enhancing the performance of the control policy.

Our findings showcase the potential of autonomous control for flexible robots. Future research directions could focus on improving the realism of contact models and devising training methods that efficiently incorporate information on deformation. By leveraging collision and contact data from high variability of soft tissue, we aim to enhance the agent's ability to generalize its autonomous control capabilities within the GI tract and beyond. Reducing the gap between simulation and reality to a reasonable level, the trained control policy in simulation can be deployed to reality. The in-vivo experiments could be introduced to validate the pre-trained control policies in the future.

REFERENCES

[1] H. Gao, X. Yang, X. Xiao, X. Zhu, T. Zhang, C. Hou, H. Liu, M. Q.-H. Meng, L. Sun, X. Zuo, *et al.*, "Transendoscopic flexible parallel continuum robotic mechanism for bimanual endoscopic submucosal dissection," *The International Journal of Robotics Research*, p. 02783649231209338, 2023.

[2] X. Yang, H. Gao, S. Fu, R. Ji, C. Hou, H. Liu, N. Luan, H. Ren, L. Sun, J. Yang, *et al.*, "A novel miniature transendoscopic telerobotic system for endoscopic submucosal dissection," *Gastrointestinal Endoscopy*, 2023.

[3] J. Lai, T.-A. Ren, W. Yue, S. Su, J. Y. Chan, and H. Ren, "Sim-to-real transfer of soft robotic navigation strategies that learns from the virtual eye-in-hand vision," *IEEE Transactions on Industrial Informatics*, 2023.

[4] J. Zhang, L. Liu, P. Xiang, Q. Fang, X. Nie, H. Ma, J. Hu, R. Xiong, Y. Wang, and H. Lu, "Ai co-pilot bronchoscope robot," *Nature communications*, vol. 15, no. 1, p. 241, 2024.

[5] O. M. Omisore, S. Han, J. Xiong, H. Li, Z. Li, and L. Wang, "A review on flexible robotic systems for minimally invasive surgery," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 1, pp. 631–644, 2020.

[6] J. Lawson, M. Veliky, C. P. Abah, M. S. Dietrich, R. Chitale, and N. Simaan, "Endovascular detection of catheter-thrombus contact by vacuum excitation," *IEEE Transactions on Biomedical Engineering*, 2024.

[7] Z. Zhang, J. Dequidt, J. Back, H. Liu, and C. Duriez, "Motion control of cable-driven continuum catheter robot through contacts," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1852–1859, 2019.

[8] P. Ferrentino, E. Roels, J. Brancart, S. Terryn, G. Van Assche, and B. Vanderborght, "Finite element analysis-based soft robotic modeling: Simulating a soft actuator in sofa," *IEEE Robotics & Automation Magazine*, 2023.

[9] J. Ibarz, J. Tan, C. Finn, M. Kalakrishnan, P. Pastor, and S. Levine, "How to train your robot with deep reinforcement learning: lessons we have learned," *The International Journal of Robotics Research*, vol. 40, no. 4-5, pp. 698–721, 2021.

[10] P. M. Scheikl, B. Gyenes, R. Younis, C. Haas, G. Neumann, M. Wagner, and F. Mathis-Ullrich, "Lapgym–an open source framework for reinforcement learning in robot-assisted laparoscopic surgery," *arXiv preprint arXiv:2302.09606*, 2023.

[11] E. Tagliabue, A. Pore, D. Dall'Alba, E. Magnabosco, M. Piccinelli, and P. Fiorini, "Soft tissue simulation environment to learn manipulation tasks in autonomous robotic surgery," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3261–3266, IEEE, 2020.

[12] S. Wang, X. Zheng, Y. Cao, and T. Zhang, "A multi-target trajectory planning of a 6-dof free-floating space robot via reinforcement learning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3724–3730, IEEE, 2021.

[13] H. Gao, X. Xiao, L. Qiu, M. Q.-H. Meng, N. K. K. King, and H. Ren, "Remote-center-of-motion recommendation toward brain needle intervention using deep reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8295–8301, IEEE, 2021.

[14] A. Attanasio, B. Scaglioni, E. De Momi, P. Fiorini, and P. Valdastri, "Autonomy in surgical robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 651–679, 2021.

[15] A. Gao, Z. Lin, C. Zhou, X. Ai, B. Huang, W. Chen, and G.-Z. Yang, "Body contact estimation of continuum robots with tension-profile sensing of actuation fibers," *IEEE Transactions on Robotics*, 2024.

[16] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities," *International journal for numerical methods in engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.

[17] N. Naughton, J. Sun, A. Tekinalp, T. Parthasarathy, G. Chowdhary, and M. Gazzola, "Elastica: A compliant mechanics environment for soft robotic control," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3389–3396, 2021.

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

[19] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.