# Transferring Virtual Surgical Skills to Reality: AI Agents Mastering Surgical Decision-Making in Vascular Interventional Robotics

Ziyang Mei, Jiayi Wei, Si Pan, Haoyun Wang, Dezhi Wu, Yang Zhao, Gang Liu, and Shuxiang Guo, *Fellow, IEEE*

***Abstract*—Vascular interventional surgery offers advantages, such as minimal invasiveness, quick recovery, and low side-effects. Performing automatic guidewire navigation on vascular surgical robots can effectively assist doctors in performing surgery. Deep learning and reinforcement learning methods have been widely used for guidewire navigation tasks. However, the challenge remains in making delivery decisions for complex and extended pathways, with real-time images being the only data source. The development of network architecture, coupled with the formulation of an efficacious training regimen for this network is of significant importance and holds substantial meaning for the advancement of autonomous systems in vascular surgical robots. Therefore, this research proposes a virtual training environment that incorporates real vascular projections to create virtual environment. In this environment, the approach is enhanced by incorporating guidewire tip-to-target distance in the reward function, using real-time images as input states. This article also employs a multiprocess proximal policy optimization algorithm to accelerate training process and a multistage training approach to reduce the training difficulty. Results demonstrate the effectiveness in virtual automated guidewire navigation and improves success rates. This research proposes a method, which generates effective inputs for the reinforcement learning agent, and enables the pretrained agent to accomplish delivery tasks in real-world scenarios.**

***Index Terms*—Guidewire navigation, reinforcement learning (RL), reward function, vascular interventional surgery.**

Manuscript received 1 December 2023; revised 1 May 2024; accepted 17 June 2024. Date of publication 15 July 2024; date of current version 18 April 2025. Recommended by Technical Editor R. Carloni and Senior Editor S. Katsura. This work was supported in part by Fujian Science and Technology Project under Grant 2022I0003, in part by Shenzhen Science and Technology Program under Grant JCYJ20220530143217037, and in part by the National Natural Science Foundation of China under Grant 52075464. *(Corresponding authors: Yang Zhao; Shuxiang Guo.)*

Ziyang Mei, Jiayi Wei, Si Pan, Haoyun Wang, Dezhi Wu, and Gang Liu are with Xiamen University, Xiamen 361102, China (e-mail: umeko@stu.xmu.edu.cn; weijiayi@stu.xmu.edu.cn; pansi@stu.xmu.edu.cn; wanghaoyun@stu.xmu.edu.cn; wdz@xmu.edu.cn; gangliu.cmitm@xmu.edu.cn).

Yang Zhao is with Xiamen University, Xiamen 361102, China, and also with the Department of Shenzhen Research Institute, Xiamen University, Shenzhen 518000, China (e-mail: zhaoy@xmu.edu.cn).

Shuxiang Guo is with the Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen 518055, China, and also with the Key Laboratory of Convergence Medical Engineering System and Healthcare Technology, Ministry of Industry and Information Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: guo.shuxiang@sustech.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMECH.2024.3420954.

Digital Object Identifier 10.1109/TMECH.2024.3420954

## I. Introduction

In RECENT years, robot-assisted surgery has garnered significant attention in the field of medicine [1], [2], and the domain of interventional medicine is no exception [3], [4], [5]. Using vascular interventional robotic systems, doctors can remotely control robots to perform complex interventional procedures outside the operating room, effectively reducing the time doctors are exposed to radiation and wearing heavy lead aprons [6]. Moreover, current advancements have introduced features like tactile feedback and force feedback, enhancing the safety of robot-assisted interventional surgery [7], [8]. The emergence of this technology has opened up new possibilities for vascular interventional procedures and greatly improved the working environment for doctors [9].

Mastering guidewire interventional surgery and precise robot operation require extensive training and a wealth of practical experience to achieve a high level of proficiency in catheter and guidewire manipulation [10]. Integrating artificial intelligence (AI) extensively into interventional surgical robots to achieve their intelligence can provide doctors with more accurate surgical assistance [11], [12], reducing training time, improving the success rate of surgeries and lowering the risk of complications. Automatic guidewire navigation technology has become a focal and trending area in the field of intelligent interventional surgical robots. Deep learning (DL) and reinforcement learning (RL) have become the main approaches in current research. DL, with its ability to automatically learn feature representations from raw data and strong pattern recognition capabilities, has provided effective solutions in various domains of artificial intelligence. Previous studies [13], [14], [15], [16], [17] have employed DL methods for guidewire navigation. However, DL lacks environmental exploration capabilities, necessitating the collection of new data for each different environment, making the training process complex and limiting the navigation capabilities of the intelligent agent.

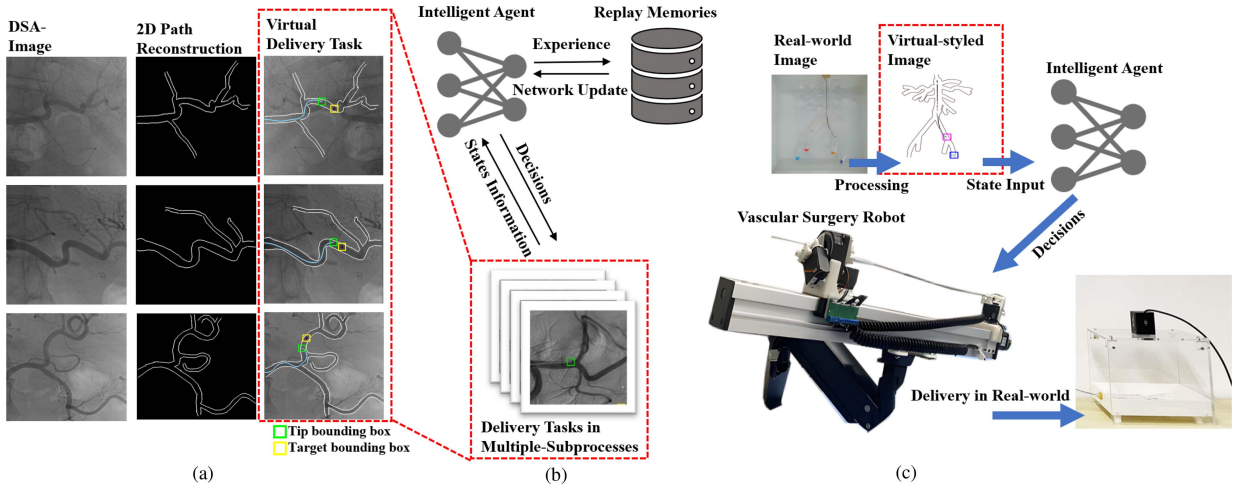


Fig. 1. Overall approach framework. Generating delivery tasks, training an RL agent, and utilizing the RL agent to control a vascular surgical robot. (a) Reconstructing vascular pathways from digital subtraction angiography (DSA) images and inputting them into the physics engine to generate delivery tasks. (b) Training the RL agent in virtual delivery tasks using a multiprocess method. (c) RL agent makes delivery decisions based on stylized images and controls the vascular surgical robot in real time to complete the delivery tasks.

Compared to DL, RL methods that seek long-term maximization of benefits through interaction with the environment are evidently more suitable for controlling surgical robots in uncertain environments that require long-term planning and decision-making. Therefore, RL has been widely adopted by researchers in this field [18], [19], [20], [21]. These research works present a scheme for decision-making in interventional delivery tasks using RL methods, and also present the possibility for employing artificial intelligence approaches in more complex and challenging conditions.

This research proposes an image-based RL intelligent decision-making method that involves recognizing and reconstructing vascular images, and then inputting the reconstruction results into the physics engine Pymunk to create interactive collision entities, as shown in Fig. 1(a). By establishing an interactive virtual delivery task and utilizing multiprocess technology to achieve synchronized simulation of multiple task groups, abundant training data is generated for the RL agent, expediting the training process as indicated in Fig. 1(b). Proximal policy optimization (PPO) algorithm is used for training, taking real-time images from the environment as inputs and designing dedicated reward functions for the training environment [22]. The agent is trained in stages to improve its robustness and adaptability in long-distance navigation and more complex environments. Upon convergence of the RL agent in virtual delivery tasks, the images from the real world are stylized. This enables the RL agent to apply the experiences learned from virtual tasks to real-world scenarios shown in Fig. 1(c). Consequently, the RL agent, guided by real-time images, can control the vascular surgical robot to complete delivery tasks in the vascular model. In addition, grad class activation mapping (Grad-CAM++) is employed to visualize the convolutional neural network (CNN) layers of the agent that to explain the important regions influencing the agent's decision-making, and evaluates the effectiveness of image stylization.

## II. Related Work

In vascular interventional surgery, achieving fully automated guidewire navigation for vascular interventional robots holds significant importance. Nowadays, guidewire navigation has made certain research advancements and can be mainly categorized into two directions: based on DL and RL.

### A. Based on DL

DL models have been widely applied in this field. Rafii-Tari et al. [14] from Imperial College London employed a hidden Markov model to capture the sequential relationship of each guidewire action, generate motion sequences, and predict future movements. They also utilized a Gaussian mixture model to jointly train motion models for the proximal and distal ends of the guidewire [13]. Zhao et al. [15] developed one-dimensional and two-dimensional CNNs to recognize the force states during guidewire manipulation and the guidewire actions based on the surgical images. They achieved automated control of the surgical robot within a closed-loop system. Wang et al. [16] used a more complex network structure in Yolov5s and added an attention mechanism, using real-time images of the ex vivo vascular model as input to improve the accuracy of the guidewire navigation. However, these approaches, which require large amount of data collection and long training for each environment, are difficult to deal with different, complex, and extended vascular pathways.

### B. Based on RL

Different from DL methods, guidewire navigation approaches based on RL exhibit strong self-exploration capabilities and adaptability to different environments. Guidewire navigation based on RL can be categorized into two directions. First, training guidewire navigation models directly on real vascular models. Chi et al. [18] imitated the surgical operations

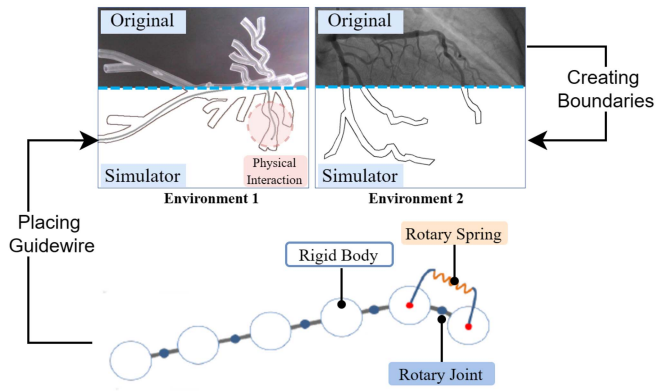


Fig. 2. Two-dimensional virtual modeling of interventional surgical environments. (a) Guidewire modeling process. (b) Complete modeling process of virtual environment and details of the collision between guide guidewire and vascular wall.

performed by expert surgeons and trained the model in the real vessel model environment using a combined PPO algorithm. However, direct training on real robots is time-consuming and limited to navigation tasks with simple environments and short distances. To address this issue, the second direction focuses on training and decision-making in virtual vascular environments, significantly reducing the time required for decision-making and movement. Meng et al. [19] used sofa to model catheter and guidewire insertion and employed the asynchronous advantage actor–critic (A3C) algorithm for training, reducing the average insertion time. Karstensen and Cho [20], [23] transferred the trained results from the virtual environment to the real extracorporeal vascular environment using the deep deterministic policy gradient (DDPG) algorithm, and You et al. [24] used dueling deep Q network (DQN). In particular, Cho et al. [23] utilized the time-series data of the guidewire tip and the corresponding environment images as input. They employed a behavior cloning approach based on their previous expert algorithm [25], followed by RL training. However, it only performs a Y-shaped branch in a virtual environment, which could not show the intelligence of its path selection in a complex vascular environment.

## III. Methods

### A. Simulation

In order to better simulate the characteristics of guidewire motion, it is necessary to consider its physical properties during the guidewire modeling process, as shown in Fig. 2. The guidewire is divided into multiple basic units connected by nodes. Rotational and elastic joints are introduced at each connection point to control the guidewire's rotation and simulate its elasticity, respectively. This modeling approach not only constrains the distances between guidewire basic units but also imparts a certain level of flexibility to the entire guidewire. Considering the force exerted on the guidewire's tail end during actual surgery, the force is applied to the tail end of the guidewire in the virtual environment. It should be noted that during guidewire rotation, torque is generated due to the connections between the basic units, and its calculation formula is as follows:

$$T = k \cdot \Delta\theta - c \cdot \omega \tag{1}$$

where $k$ represents the torsional spring elasticity coefficient, $c$ is the torsional spring damping coefficient, represents the change in angle between the centroids of two basic units, and $\omega$ is the current angle.

Virtual training environments, as shown in Fig. 2, are created by Pymunk. Pymunk is a 2-D physics engine based on Chipmunk [26], which enables realistic force and collision simulations [27]. Using Pymunk, the motion of the guidewire win blood vessel and its elastic collisions with the vessel walls can be simulated. Environment 1 was designed to mimic an ex vivo model, while Environment 2 was from automatic regionbased coronary artery disease diagnostics using x-ray angiography images (ARCADE) [27], which is a segmentation dataset under the DSA image containing expert annotations. The boundaries of the vasculatures were obtained, respectively, by the Canny [28] for Environment 1 and the expert-annotated segmentation labels for Environment 2, then translate them into collision boundaries in the physics engine.

### B. RL Algorithm

The RL algorithm adopted in this study is PPO. Compared to other RL algorithms, PPO has advantages, such as fast training speed, ease of implementation, and good convergence properties. In this algorithm, images of the current environment state are the input $s_t$ of the model. The tip of the guidewire and the target point are identified using bounding boxes, as shown in Fig. 1(a). Using images as state inputs, as opposed to traditional state representations, allows the agent to learn and discover relevant features better, such as colors and textures in the environment. Thus, during the iterative learning process, the agent could have a more comprehensive understanding of the environment. The input images have a size of 256 × 256 pixels. The action space $a_t$ in this environment is discrete, including actions, such as moving forward, moving backward, clockwise rotation, counterclockwise rotation, and pausing/waiting. After each decision made by the neural network policy $\pi(a_t|s_t)$, the agent selects an action from the five possible actions based on the action probabilities outputted by the neural network. At the end of each training round, the generalized advantage estimation (GAE) is calculated

$$\hat{A}_t^{\text{GAE}} = \sum_{y=0}^{T} (\gamma\lambda)^{T-t} \cdot \delta_t. \tag{2}$$

$T$ is the total number of time steps in a single task. $t$ is the current moment. $\gamma$ is the reward discount factor that determines the balance between emphasizing immediate rewards and future total returns in advantage estimation. $\lambda$ is the GAE coefficient, and $\delta_t$ represents the TD advantage [22], which is calculated as follows:

$$\delta_t = r_t + \gamma \cdot V(s_{t+1}) - V(s_t) \tag{3}$$

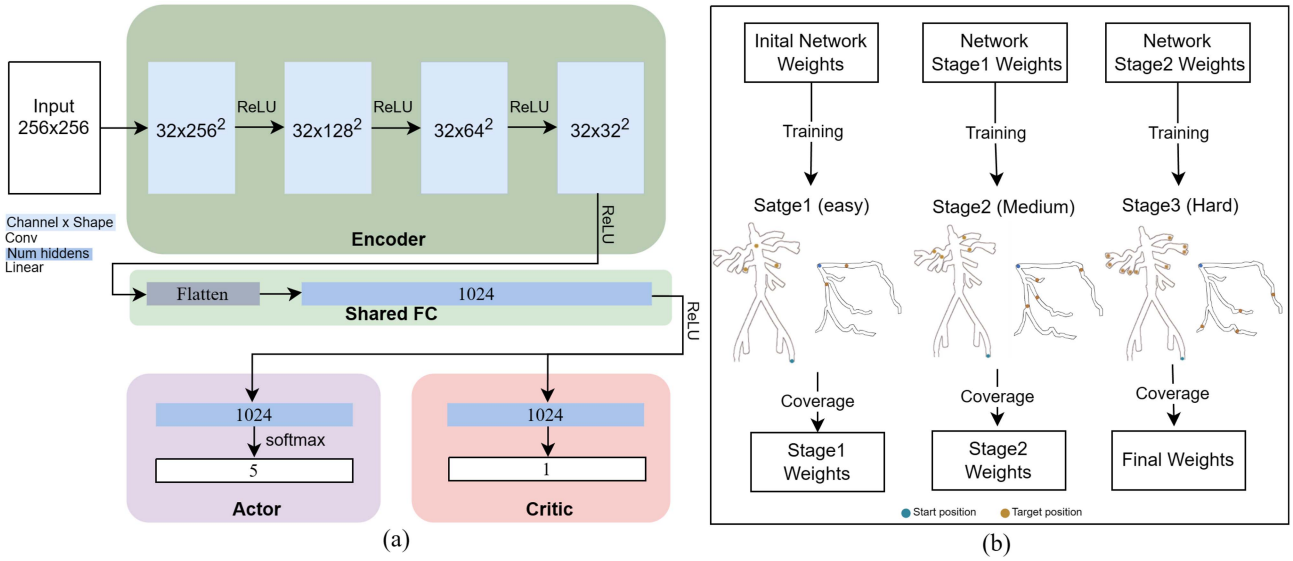


Fig. 3. Training framework. (a) Actor–critic agent network structure. (b) Illustration of multistage training method with inherited weights across different difficulty levels.

where $V(s)$ is the state value function, which represents the estimated future returns of the current task based on the current environmental state. Meanwhile, $r_t$ represents the current reward.

The architecture of the neural network is depicted in Fig. 3(a). CNN is chose for our image encoder network. Preliminary testing has indicated that, compared to structures, such as ResNet and VGG, CNN demonstrates superior training outcomes. Objective of the algorithm is to maximize the loss function

$$L^{\text{Net}} = L^{\text{PPO}}(\theta) + c \cdot L^{\text{Value}}(\theta) + \beta \cdot E(\theta) \tag{4}$$

$$L^{\text{PPO}}(\theta_\pi) = \frac{1}{N}\sum_{t=0}^{N}(\min(R_t(\theta_\pi)\cdot \hat{A}_t^{\text{GAE}} \; clip(R_t(\theta_\pi), 1-\epsilon, 1+\epsilon)\cdot \hat{A}_t^{\text{GAE}}))^2 \tag{5}$$

$$R_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)} \tag{6}$$

$$L^{\text{Value}}(\theta) = \frac{1}{N}\sum_{t=0}^{N}(\hat{A}_t^{\text{GAE}} + V_{\text{old}}(s_t) - V_{\theta_v}(s_t))^2 \tag{7}$$

$$E(\theta_\pi) = -\frac{1}{N}\sum_{t=0}^{N}\sum_{n=0}^{M}\pi(a_n|s_t)\cdot \ln(\pi(a_n|s_t)) \tag{8}$$

the overall loss function consists of the PPO loss function $L^{\text{PPO}}$, the value loss function $L^{\text{Value}}$, and the entropy loss $E(\theta)$. In the PPO loss function $L^{\text{PPO}}$, $R_t(\theta)$ represents the ratio between the current policy $\pi_\theta(a_t|s_t)$ and the policy $\pi_{\text{old}}(a_t|s_t)$ before optimization, which improves the stability of the convergence of the algorithm. The entropy value $E$ estimates the distribution of decision-making by the policy model, where a smaller entropy value indicates more concentrated decision-making, while a larger entropy value indicates more diversified decision-making. The weights $c$ and $\beta$ are used to adjust the influence of the value loss function and entropy loss function, respectively, fine-tuning the updating trend of the neural network model during training. $N$ represents the total number of data samples drawn in one training session, and $M$ is the number of all actions output by $\pi_\theta(a_t|s_t)$.

**Algorithm 1:** Multi-Process PPO Algorithm.

Initialize network parameters $\boldsymbol{\theta}$ of value and policy
**for** *episode* in $MAX\_EPISODE$ **do**
  Initialize all virtual environments and gets the initial state of those environments $\mathbf{s_0}$
  $s_t \leftarrow s_0$
  **for** *step* in $MAX\_STEP$ **do**
    $a_t \leftarrow Sample(\pi(a_t|s_t))$
    $(s_{t+1}, r_t, done_t) \leftarrow step(a_t)$
    experience replay $\leftarrow (s_t, a_t, \pi(a_t|s_t), V(s_t), s_{t+1}, r_t, done_t)$
  $s_t \leftarrow s_{t+1}$
  $\hat{A}_t^{GAE} \leftarrow \sum_{t=0}^{T}(\gamma\lambda)^{T-t}\cdot\delta_t$
  **for** $t$ in $RC$ **do**
    $L^{Net}(\theta) \leftarrow L^{PPO}(\theta) + c\cdot L^{Value}(\theta) + \beta E(\theta)$
    **Update network**
  Clear experience replay

To expedite the training process of the algorithm and enhance the efficiency of policy updates, a multiprocess PPO algorithm was implemented. In this approach, each process collects experience data, which is then consolidated into a shared experience buffer. In addition, the size of the experience pool is increased. The flowchart of the multiprocess PPO algorithm is shown as Algorithm 1 and detailed parameters of the algorithm can be found in Table I.

TABLE I
PARAMETERS OF MULTIPROCESS PPO

| Parameters | Description | Value |
|---|---|---|
| lr | Learning rate | $5 \times 10^{-5}$ |
| $\epsilon$ | PPO clip factor | 0.1 |
| $c$ | Critic coefficient | 1 |
| $\beta$ | Entropy coefficient | 0.01 |
| $\gamma$ | Reward decay factor | 0.99 |
| $\lambda$ | GAE coefficient | 0.95 |
| RC | Experience replay reuse count | 4 |
| $S$ | Input image size | $256 \times 256$ |

### C. Multistage Training

Since a single interventional surgery may involve entering multiple branches and have multiple target points, each target point may have a different distance from the starting point. Therefore, it is essential to enable the robot to have the capability of navigating to multiple target points. Currently, long-distance navigation in vessels poses significant challenges. The simple approach of setting a single starting point and a single target point does not allow the agent to converge successfully. To address this issue, a multistage training method was implemented. First, multiple subtarget points are marked along the path between the starting point and the target point. Since the distances between the target points and the starting point are different, the target points are divided into a subset of target points that are close to the radius from the starting point, and define a final target point. For example, as shown in Fig. 3(b), the training process is divided into three stages. After each stage converges, the model weights are used as the initial weights for the next stage, then the training continues. This approach transforms the overall long-distance navigation problem into multiple short-distance navigation problems. This training method not only facilitates the convergence of the agent but also exhibits good adaptability to the task of simulating vessel models from virtual to real-world scenarios.

### D. Reward Function

The design of the reward function plays a crucial role in RL tasks, and different environments require different reward function settings. Current research has considered the minimization of decision steps, where each additional step taken by the guidewire receives a negative reward. When the guidewire reaches the target point, a positive reward is given [20], [29]. To guide the guidewire toward the correct branching points, some studies have also imposed restrictions on the guidewire's path selection by setting forbidden passage locations, and if the guidewire reaches these locations, it incurs a negative for complex environments. The current reward function settings, which simply penalize each step, may reduce the agent's exploration motivation and fail to effectively guide the guidewire to reach distant target positions. To achieve a balance between constraining the maximum number of steps and ensuring adequate exploration, we propose a distance-based incentive approach. The agent is only rewarded when it reaches the target point or the steps reaches our preset limit. This approach allows the agent to explore extensively in environments where exploration is necessary, preventing the agent from getting lost. Therefore, at the end of each training iteration, the algorithm calculate the distance between the guidewire tip and the target point. Using this distance information, higher reward is assigned to the agent as the guidewire tip gets closer to the target point, thereby motivating the guidewire to move toward the target point.

In conclusion, the final formula for the reward function is as follows:

$$r(s,a) = -w \cdot \ln(D) + b \tag{9}$$

$$D = c \cdot \sqrt{(x_{\text{tip}} - x_{\text{target}})^2 + (y_{\text{tip}} - y_{\text{target}})^2} \tag{10}$$

where $c$ is a constant gain coefficient, $D$ represents the distance from the guidewire tip to the target point at the end of each training iteration, which can be calculated based on the corresponding coordinates. $(x_{\text{tip}}, y_{\text{tip}})$ and $(x_{\text{target}}, y_{\text{target}})$ are the coordinates of the guidewire tip and the target point, respectively. $w$ and $b$ are reward adjustment coefficients, which are used to expand the reward space.

## IV. EXPERIMENTS

### A. Simulation

During the training process, the maximum number of steps was restrict to 400 per round for the agent to reach the goal. In delivering procedures, precisely rotating the guidewire tip at vessel bifurcations is a major challenge. Even for seasoned experts with keyboard control, steering the guidewire accurately into the curved branches often requires multiple attempts. However, the trained agent is capable making multiple attempts at the vessel's bifurcation, until entering the correct branch. Using the multiprocess PPO algorithm, final stage achievement is shown in Fig. 9. As the task difficulty increases, the convergence speed slightly decreases. The first stage reaches convergence in 250 rounds, the second stage converges in 450 rounds, and the third stage converges in 1800 rounds. At each stage, training for the next stage begins once convergence becomes stable. After 10 000 rounds of testing, the average number of steps taken by the guidewire to reach success rates of 99.30%, 99.26%, and 98.34% for each stage, respectively. Due to the increasing task difficulty, occasional errors may occur in the third and final stage.

Data from 1000 rounds of interact in the virtual environment have been collected by using pretrained agents. A total of 952 were successful and 48 were failed, resulting in a total of 133 106 decision inputs and corresponding decision outcomes. An analysis is performed on this data. Each set of decision outcomes is represented as $P = \pi_{a_1}, \pi_{a_2}, \ldots, \pi_{a_n}$, where $\pi_{a_n}$ represents the probability of action $a_n$ being executed. Fig. 4(a) presents box plots and bar charts for these decision inputs. The figures clearly demonstrate significant class imbalance when this task is approached as a classification task. The distribution of the training data exhibits a significant class imbalance problem. For example, during the guidewire delivery process, forward action is required mostly, while backward action is required in very few instances. However, the backward movement is critical important. In many situations, only performing a backward

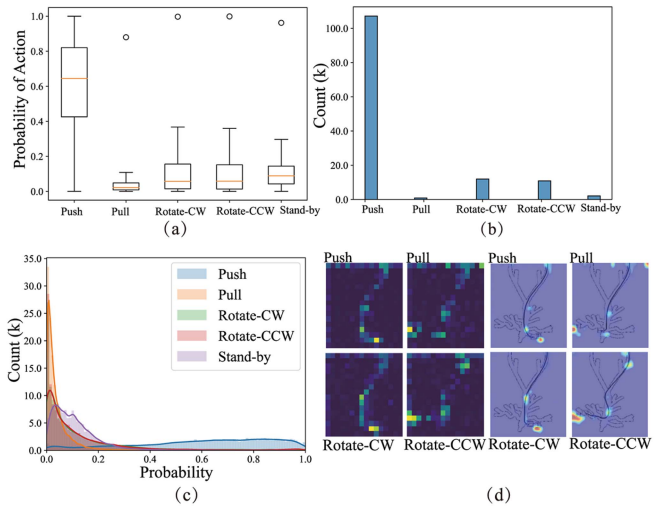


Fig. 4. Experiment results in virtual environment. (a) Probability distribution box plot of the agent's output decision. (b) Distribution histogram of agent's decisions. (c) Probability distribution diagram of agent execution decision. (d) CAM visualization under different decisions.

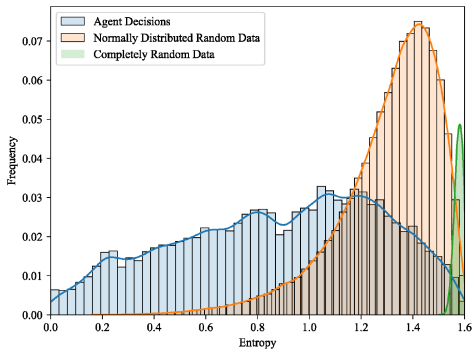


Fig. 5. Comparing entropy values from 133 106 sets of output decisions with control groups: normally distributed random data and completely random data.

movement can correct the direction of guidewire. The class imbalance problem introduces difficulty to the neural network learning and convergence. However, the RL agent output high confidence decision probabilities in such imbalanced categories and complete most of the delivery tasks successfully.

The Smooth Grad-CAM++ class activation map method proposed by Omeiza et al. [30] is utilized to visualize the regions that are crucial for category discrimination during the decision making process of the network. Decision images and their corresponding confidence levels for different decisions made by the agent are obtained from the aforementioned decision data. These images are then reinputted into the neural network, and the gradients of the class during forward propagation are recorded to visualize the class activation maps. Class activation visualization was performed on the last layer of the intelligent agent's CNN. Fig. 4(d) demonstrates the visualization results of class activation maps when the intelligent agent makes different decisions in a virtual environment. The visualization of class activation maps reveals that annotating the target position in the image plays a crucial role in the network's decision-making. In addition, the shape and entry position of the guidewire also have a significant impact on the network's decision. This provides important insights for the application of RL agent models trained and converged in virtual environments to real interventional robotic instruments.

In some scenario of simpler paths, it may only require executing forward and rotational movements in certain proportions to reach the target point consistently. This single random decision-making process may solve some straightforward path decision problems, but it fails to demonstrate the intelligence of neural networks and lacks the ability to handle complex paths, correct manipulator postures, and make sophisticated decisions. The entropy function is used to evaluate the quality of the agent's decision-making. The entropy function can be represented as follows:

$$E(P) = \frac{1}{A}\sum_{n=1}^{A}(-\pi_{a_n} \cdot \ln(\pi_{a_n})) \tag{11}$$

where $\pi$ represents a set of decision probabilities and $A$ represents the length of the action space. In this case, $\sum_{n=1}^{A} \pi_{a_n} = 1$. The entropy function can effectively measure the quality of a set of decisions. A set of uncertain decisions will have a high entropy value, while decisions with strong determinism will have a lower entropy value. Fig. 5 presents the entropy values obtained from 133 106 sets of output decisions in the aforementioned experiment and compares them with an equal number of control groups: one consisting of normally distributed random data, and the other consisting of completely random data. The normally distributed random data simulate simple decisions generated based on statistical information, while the completely random data simulate the scenario where no action probability distribution has been learned. In this case, both the normally distributed random numbers and completely random numbers are processed through the softmax function to form a set of probabilities, satisfying $\sum_{n=1}^{A} \pi_{a_n} = 1$. The decision sets made by RL agent generally have lower entropy values, indicating higher determinism compared to completely disordered actions or simple statistical induction. In this study, the magnitude of decision entropy is regarded as one of the important indicators to evaluate whether the agent converges during the training process.

## B. Applying Agent Into Real-World Robot

A comprehensive vascular interventional robot system is designed to achieve autonomous operation of guidewire navigation, as shown in Fig. 6. This system consists of a mechanical structure capable of manipulating the interventional instruments, a lower level subsystem that receives and translates operational commands into corresponding motor control signals, and an upper level interface for programmable logic control.

The mechanical structure of the robot comprises friction wheel modules, a rotating platform module, and a linear motion module. The friction wheel module simulates the finger delivery

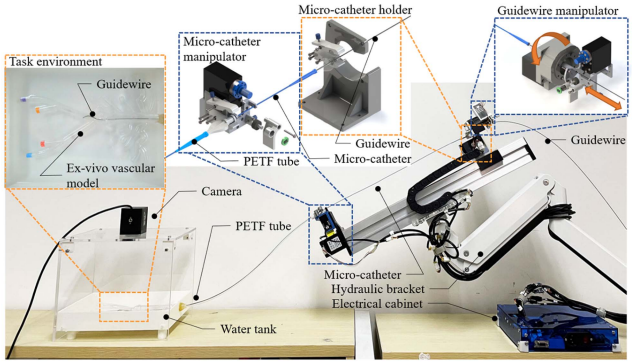


Fig. 6. Vascular interventional robot with ex vivo vascular model setup.

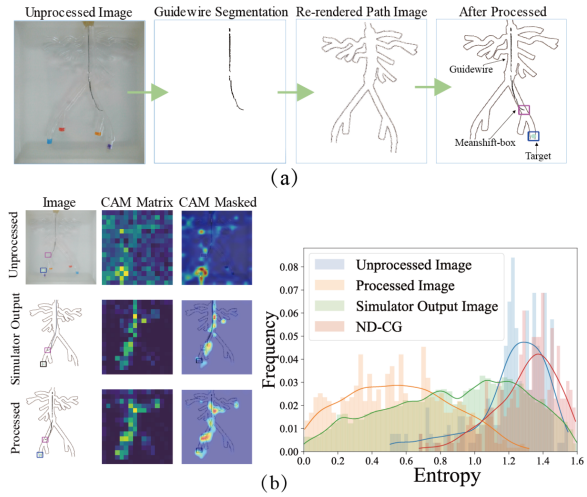


Fig. 7. Image stylization and visualization of experimental results. (a) Image stylization of ex vivo vessel model task. (b) CAM-visualizations comparison (left) and decision entropy distributions comparison (right).

action of a human hand on the interventional guidewire, enabling smooth and precise delivery and providing clamping force on the guidewire. The rotating platform, driven by the friction wheel module along the axial direction, enables the rotation of the interventional guidewire. Through the combined motion of the friction wheel and rotating platform modules, the robot platform can achieve independent and noninterfering actions of guidewire delivery and rotation. The linear motion module controls the forward and backward movement of the microcatheter wrapped around the guidewire. To avoid bending of the catheter and provide greater pushing force for the microcatheter during delivery, an additional set of friction wheels is added to the front end of the linear platform, assisting in stable microcatheter delivery. The friction wheel modules of the robot utilize the HX8-U28 bus servo motor from Shenzhen Huaxin Technology, while the rotating platform and linear motion platform employed ASM34AK stepper motors from Oriental Motor Company.

The lower level control system of the robot is designed based on ESP32. It utilizes the pulsewidth modulation waveform controller in the ESP32 chip to generate variable-frequency pulse signals, which are converted into drive sequences to control the movement of the stepper motors through motor drivers. The universal asynchronous receiver transmitter (UART) module in the ESP32 chip is used for communication with the bus servo motors and setting the motor's speed. The lower level system integrates an RS232 communication module that converts the ESP32's UART 3.3V-TTL signal to a 12 V RS232 signal, allowing industrial-grade remote communication by transmitting the signal through a cable to PC. The control logic of the lower level system was developed using FreeRTOS, which analyzes the command signals from the PC and drives the corresponding motors accordingly.

The upper level control code of the robot is written in Python. It establishes a serial communication channel between the PC and the robot system using the PySerial library. Based on mechanical structure of the robot, it calculates the relationship between the operational speed of the interventional instrument and the motor control speed, driving the robot to perform the required instrument movements. These motion methods are encapsulated as callable program interfaces.

The skills learned by the RL agent are solely derived from a two-dimensional virtual environment. However, as indicated by Akkaya et al.'s [31] research, there are inevitable differences between virtual simulations and real environments. There are significant disparities between two-dimensional virtual environments and real three-dimensional environments. The real world exhibits more complex force relationships, and the motion of the guidewire is harder to predict. Consequently, applying the experiences gained from the two-dimensional virtual environment to the real world presents considerable challenges. To address this issue, experiments were conducted using a transparent vessel model, as illustrated in Fig. 6. A polytetrafluoroethylene (PTFE) catheter is installed at the front end of the robot and connected it to an ex vivo vessel model. Above the vessel model, a camera module was positioned to capture images of the guidewire's motion within the transparent model.

The images captured by the camera are processed and fed into the neural network of our RL agent. The entropy value output by the agent is utilized to guide the image optimization strategy. Lower entropy value in the decision probabilities signifies more definitive choices by the agent, whereas higher entropy suggests a more dispersed and less certain decision-making process. To reduce the entropy value, the captured images were rerendered, as illustrated in Fig. 7(a). The "unprocessed" images, subjected only to gray-scale conversion and brightness adjustment, are distinct from the "simulator output" representing the ideal input for the agent because it was trained on simulators with this kind of images. The "processed" images, synthesized using the method illustrated in Fig. 7(a) are designed to emulate these ideal conditions. Fig. 7(b) presents the effects of our synthesized images from two different perspectives. The left side demonstrates the class activation mapping (CAM) responses of the neural network from three different input images. CAM matrix represents the visualization result, while the CAM mask is the effect of smoothed CAM result semitransparently overlaying on the original image. While the "unprocessed" images displayed significantly different activation patterns compared to the "simulator output," the "processed" images exhibited

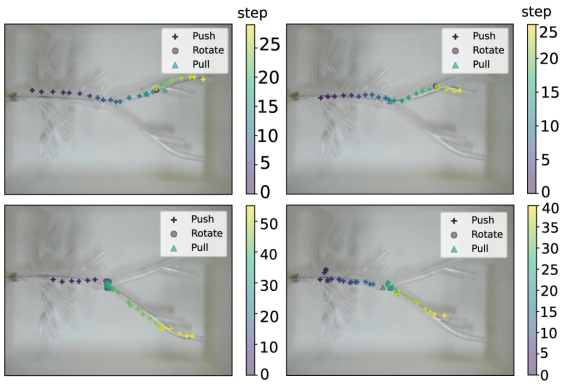


Fig. 8. Demonstration of the guidewire delivery process in ex vivo vessel model.The position of patterns represents the front end of the guidewire at a certain moment, the shape of the pattern represents the action taken at this moment, and the color change of the pattern represents the step order of this position.

TABLE II
ACTIONS OF REAL WORLD EXPERIMENT

| Action | Parameter |
|---|---|
| push | 10mm |
| pull | 10mm |
| rotate-cw | 45° |
| rotate-ccw | 45° |
| wait | 0.5s |

activation patterns closely resembling those of the "simulator output." This resemblance suggests that the agent's focus areas are more consistent with the features recognized during its virtual environment training. On the right side of Fig. 7(b), the entropy distribution of the decision probabilities, as generated by the neural network across different image inputs, is presented. The "ND-CG" data consists of random numbers drawn from a normal distribution N(0, 1) and processed by softmax, used for reference purposes. When input into the network, the unprocessed images resulted in entropy values approximating those of the random number test group, while the processed images produced substantially lower entropy values, closely mirroring the agent's performance within the simulator.

In real-world delivery experiments, such image processing techniques yielded remarkable results. The guidewire achieve a high success rate in robot operation tasks. The demonstration of the guidewire delivery process is illustrated in Fig. 8. To evaluate the efficacy of the RL agent approach within the context of real-world interventional surgical robotics. An interventional surgeon was invited to be part of an expert group in our comparative experiments. The experimental design allowed interventional surgeons to make decisions on whether to move forward, retract, or rotate (clockwise or counterclockwise) based on the image information from the current frame. Each action had the same duration, and a single action was executed per decision output. The specific parameter settings are shown in Table II. Upon surgeon making a decision, the robotic system will execute the corresponding action, moves the guidewire, then updates the image to the surgeon. In this scenario, a total of 20 delivery

TABLE III
PERFORMANCE COMPARISON

| Target | Item | Expert decision | Expert clone | RL agent |
|---|---|---|---|---|
| A | Success rate | 100% | 0% | 82.50% |
| | Average steps | 31 | - | 41.61 |
| B | Success rate | 100% | 0% | 87.50% |
| | Average steps | 26.95 | - | 36.8 |
| C | Success rate | 100% | 0% | 82.50% |
| | Average steps | 34.05 | - | 45.94 |
| D | Success rate | 100% | 0% | 72.50% |
| | Average steps | 41.65 | - | 54.10 |

experiments were conducted for each target point. At the same time, the complete decision sequence made by the interventional surgeon was recorded. After retracting the guide wire back to the starting point, the surgeons' decisions were replicated according to the recorded sequence to explore the importance of real-time decision-making in the experimental environment. Finally, the RL agent was programmed to make decisions based on image information, completing 40 delivery experiments for each target point. A criterion is set that if the decision-making process exceeded 100 steps without successfully delivering to the target location, the delivery attempt would be deemed a failure. The results of these experiments are presented in Table III. From the results, it can be observed that the decisions made by the interventional surgeons could achieve 100% completion rate for the delivery tasks to all target points, with the fewest number of decision steps. The RL agent also demonstrated a high success rate, and the average number of steps consumed in successful delivery tasks was slightly higher than that of expert decisions. Notably, the success rate of expert clones was 0%. This result indicates that the delivery task has a certain level of complexity and randomness introduced by the environment, and real-time decision-making is crucial at each moment; relying solely on recording and replicating historical decisions are not sufficient to complete such delivery tasks. Although the RL agent did not match the experts in terms of success rate and number of steps consumed, its performance showed the stability and reliability of the algorithm. Moreover, because it can continuously learn and adapt, it constantly optimizes its strategy to cope with various challenges. Therefore, the agent has the potential to further improve the success rate and decision-making efficiency, making it worthy of further research and application.

### C. Evaluation

*Multistage training:* In RL tasks, when the difficulty level of subtasks is similar, the agent is more likely to converge. Therefore, the staged training approach divides the subtasks with different difficulty levels within an environment, gradually increasing the task difficulty and allowing the agent to adapt to the environment step by step. Through this multistage training strategy, the success rate of the agent in each stage is improved, and the performance of guidewire navigation is gradually enhanced throughout the training process. Although multistage training may slightly reduce the convergence speed, it significantly improves the success rate.

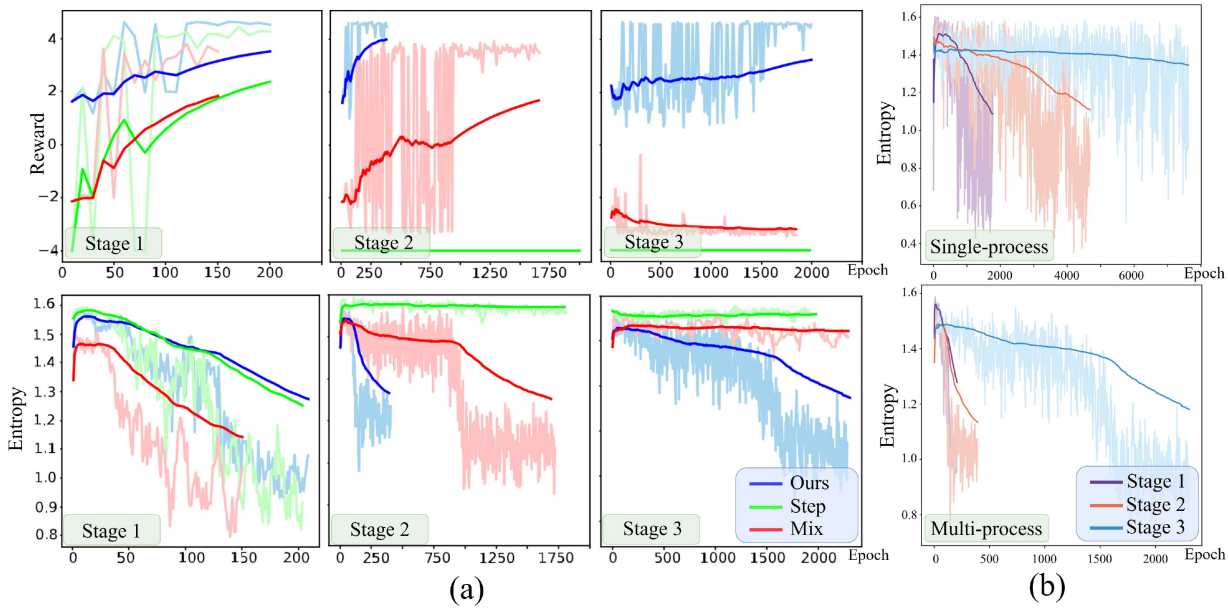


Fig. 9. Training curves during the evaluation process. (a) Reward and entropy curves during different stages with different reward functions. (b) Entropy curves during different stages of single-process and multiprocess.

TABLE IV
SUCCESS RATE OF TWO TRAINING WAYS

| Training method | Targets S | Targets D |
|---|---|---|
| One training | 89.27% | - |
| Multitraining | 98.65% | 98.34% |

Two branches are removed from the target point set of the stage 3 in Fig. 3 that are duplicates from the stage 1, leaving target points with distance similar (Targets S) to the guidewire starting point. These selected target points, along with the target points with different distance from the stage 3 (Targets D), are used separately as final target points for comparative experiments. When the final target point set is targets S, both training approaches can converge. However, when the final target point set is targets D, one-shot training fails to converge. This is because the task distances from the starting point to the target points in Targets S are relatively similar, while in Targets D, the radii of target points differ significantly, leading to increased task difficulty. Detailed data charts for targets D are available in the supplementary materials. Table IV provides a comparison of success rates between one-shot training and multistage training. To reach the final target point, one-shot training has a slightly faster convergence rate than multistage training. However, after 10 000 rounds of testing, the average success rate of multistage training for Targets S is 9.38% higher compared to one-shot training. The multistage training approach demonstrates more stable performance during testing.

*Reward function:* The effectiveness of the proposed reward function is validated through comparative experiments on reward functions. In Fig. 9(a), the changes in reward values and entropy under three reward functions were compared: "ours," "step," and "mix." In the "step" function, the agent receives positive rewards upon reaching the target point and negative rewards for each step taken. The "mix" function combines elements of both "ours" and "step" reward functions. In the stage one, "mix" exhibits slightly faster convergence and more deterministic strategies. However, in the stage two, where the task complexity increased, "ours" reward function show significantly faster convergence, reaching convergence around 250 rounds. The convergence speed is approximately four times faster than the approach that combined two reward functions. In the entropy curve comparison of the second stage, it is evident that "ours" reward function also outperforms the other two reward functions in terms of strategy determinism. In the final stage, only using our reward function successfully completed the task, and the curve converged.

*Multiprocess:* A multiprocessing training approach is used to enhance the training efficiency of RL agents in delivery tasks. To validate the effectiveness of the multiprocessing training method, multistage delivery task training is undertaken using both single-process and multiprocess methods in Environment 1. The entropy of decision outputs is utilized to observe the convergence efficiency of neural networks under different conditions, as illustrated in Fig. 9(b). The performance of the multiprocess method with four processes is compared against the single-process method. It can be observed that compared to the single-process approach, the multiprocess method demonstrates significantly higher training efficiency at each training stage, and exhibits better training outcomes particularly in the most challenging stages. The analysis primarily attributes this to several factors. First, the training algorithm framework involves network training and clearing the replay buffer after each epoch, resulting

TABLE V
GUIDEWIRE TIP AND TARGETS WITHOUT BOUNDING BOXES

| Environment | Stage | Convergence epochs | Success rate | Average step |
|---|---|---|---|---|
| Environment 1 | stage 1 | 350 | 99.25% | 102.56 |
| | stage 2 | 800 | 97.81% | 122.85 |
| | stage 3 | 2600 | 96.43% | 127.60 |
| Environment 2 | stage 1 | 150 | 100% | 38.21 |
| | stage 2 | 700 | 98.93% | 70.28 |
| | stage 3 | 2400 | 97.68% | 86.35 |

TABLE VI
GUIDEWIRE TIP AND TARGETS WITH BOUNDING BOXES

| Environment | Stage | Convergence epochs | Success rate | Average step |
|---|---|---|---|---|
| Environment 1 | stage 1 | 250 | 99.30% | 99.70 |
| | stage 2 | 450 | 99.26% | 115.93 |
| | stage 3 | 1800 | 98.34% | 121.68 |
| Environment 2 | stage 1 | 100 | 100% | 36.58 |
| | stage 2 | 550 | 99.43% | 66.33 |
| | stage 3 | 2250 | 99.17% | 79.03 |

in a significantly larger volume of interaction data stored in the replay buffer during multiprocess training compared to single-process training. Second, during multiprocess training, the replay buffer concurrently contains interaction data from different tasks. The presence of interaction data from different tasks in the replay buffer helps diversify the training data, allowing the network to learn from a wider range of situations and scenarios. This diversity enhances the network's robustness in each training iteration, enabling it to adapt to various inputs and environments more effectively. Finally, while theoretically achieving similar results by programming RL agents to undergo training after multiple epochs is possible, the multiprocessing approach maximizes the utilization of CPU multicore resources, thereby improving the efficiency of RL algorithm execution and consequently reducing the time required for training neural networks.

*With bounding boxes:* The convergence speed, success rate, and average number of steps in the three stages of the navigation task in the two environments were compared between the cases with and without bounding boxes in Fig. 1(a). Each stage was tested for 10 000 rounds. In the stage one, a relatively simple task, the guidewire tip with bounding boxes exhibited a slightly higher success rate and fewer average steps compared to without bounding boxes. However, as the task difficulty increased, the advantage of using bounding boxes become more prominent. In the stage 3, as shown in Tables V and VI, the success rate increases by 1.91% and 1.49%, and the average step decreases by 5.92 and 7.32 when bounding boxes are employed in two environments. When bounding boxes are used, the network tended to focus more easily on crucial areas, such as the guidewire tip and target point. This facilitated an accelerated training convergence process, aiding the guidewire in reaching the target point quickly and enhancing navigation accuracy.

### D. Discussion

This research employs the multiprocess PPO algorithm for training in guidewire navigation tasks. Real-time images are used as input states to enable the neural network to extract deeper environmental features, allowing the agent to have a more comprehensive understanding of the entire training environment.

The reward function is improved by abandoning the idea of penalizing every step and instead incorporating the distance between the guidewire tip and the target point. The purpose is to encourage the agent to explore the environment more boldly in complex environments and gradually approach the target point, thereby accelerating the training process. Compared to commonly used reward functions, our approach improve the convergence speed and save training time. In addition, a multistage training method is employed, dividing the complex target path into multiple subtarget points rather than training all at once. Although the training time is comparable to one-time training, the success rate of the guidewire reaching the final target point is significantly increased. This method enhances the navigational stability of the guidewire.

Through these optimization method, favorable results are achieved in this study. The agent is trained in a virtual environment using an improved reward function and a multistage training strategy, enabling faster reaching of the target point and exhibiting more stable navigational capabilities. These improvements have the potential to enhance the efficiency and accuracy of guidewire operations in practical applications.

## REFERENCES

[1] F. Cepolina and R. P. Razzoli, "An introductory review of robotically assisted surgical systems," *Int. J. Med. Robot. Comput. Assist. Surg.*, vol. 18, no. 4, 2022, Art. no. e2409.
[2] L. Qian, J. Y. Wu, S. P. DiMaio, N. Navab, and P. Kazanzides, "A review of augmented reality in robotic-assisted surgery," *IEEE Trans. Med. Robot. Bionics*, vol. 2, no. 1, pp. 1–16, Feb. 2020.
[3] V. M. Pereira et al., "First-in-human, robotic-assisted neuroendovascular intervention," *J. Neurointerventional Surg.*, vol. 12, no. 4, pp. 338–340, 2020.
[4] Z. Yang, L. Yang, M. Zhang, C. Zhang, S. C. H. Yu, and L. Zhang, "Ultrasound-guided catheterization using a driller-tipped guidewire with combined magnetic navigation and drilling motion," *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 5, pp. 2829–2840, Oct. 2022.
[5] J. Hwang, J.-Y. Kim, and H. Choi, "A review of magnetic actuation systems and magnetically actuated guidewire-and catheter-based microrobots for vascular interventions," *Intell. Service Robot.*, vol. 13, pp. 1–14, 2020.
[6] C. Yang et al., "A vascular interventional surgical robot based on surgeon's operating skills," *Med. Biol. Eng. Comput.*, vol. 57, pp. 1999–2010, 2019.
[7] C. Lyu et al., "Flexible tactile-sensing gripper design and excessive force protection function for endovascular surgery robots," *IEEE Robot. Autom. Lett.*, vol. 8, no. 10, pp. 6171–6178, Oct. 2023.
[8] X. Bao, S. Guo, C. Yang, and L. Zheng, "Haptic interface with force and torque feedback for robot-assisted endovascular catheterization," *IEEE/ASME Trans. Mechatron.*, vol. 29, no. 2, pp. 1111–1125, Apr. 2024.
[9] S. Miyachi, Y. Nagano, R. Kawaguchi, T. Ohshima, and H. Tadauchi, "Remote surgery using a neuroendovascular intervention support robot equipped with a sensing function: Experimental verification," *Asian J. Neurosurgery*, vol. 16, no. 2, pp. 363–366, 2021.
[10] J. Guo, M. Li, Y. Wang, and S. Guo, "An image information-based objective assessment method of technical manipulation skills for intravascular interventions," *Sensors*, vol. 23, no. 8, 2023, Art. no. 4031.
[11] Y. Zhao et al., "Remote vascular interventional surgery robotics: A literature review," *Quantitative Imag. Med. Surg.*, vol. 12, no. 4, 2022, Art. no. 2552.

[12] A. Moglia, K. Georgiou, E. Georgiou, R. M. Satava, and A. Cuschieri, "A systematic review on artificial intelligence in robot-assisted surgery," *Int. J. Surg.*, vol. 95, 2021, Art. no. 106151.
[13] W. Chi, J. Liu, H. Rafii-Tari, C. Riga, C. Bicknell, and G.-Z. Yang, "Learning-based endovascular navigation through the use of non-rigid registration for collaborative robotic catheterization," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 13, pp. 855–864, 2018.
[14] H. Rafii-Tari, J. Liu, C. J. Payne, C. Bicknell, and G.-Z. Yang, "Hierarchical HMM based learning of navigation primitives for cooperative robotic endovascular catheterization," in *Proc. 17th Int. Conf., Med. Image Comput. Comput.-Assist. Intervention*, 2014, pp. 496–503.
[15] Y. Zhao et al., "A CNN-based prototype method of unstructured surgical state perception and navigation for an endovascular surgery robot," *Med. Biol. Eng. Comput.*, vol. 57, pp. 1875–1887, 2019.
[16] S. Wang, Z. Liu, X. Shu, Y. Cao, L. Zhang, and L. Xie, "Study on autonomous delivery of guidewire based on improved YOLOV5s on vascular model platform," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2022, pp. 1–6.
[17] L. Mekki et al., "Surgical navigation for guidewire placement from intraoperative fluoroscopy in orthopaedic surgery," *Phys. Med. Biol.*, vol. 68, 2023, Art. no. 215001.
[18] W. Chi et al., "Collaborative robot-assisted endovascular catheterization with generative adversarial imitation learning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 2414–2420.
[19] F. Meng, S. Guo, W. Zhou, and Z. Chen, "Evaluation of an autonomous navigation method for vascular interventional surgery in virtual environment," in *Proc. IEEE Int. Conf. Mechatron. Autom.*, 2022, pp. 1599–1604.
[20] L. Karstensen et al., "Learning-based autonomous vascular guidewire navigation without human demonstration in the venous system of a porcine liver," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 17, no. 11, pp. 2033–2040, 2022.
[21] J. Ritter, L. Karstensen, J. Langejürgen, J. Hatzl, F. Mathis-Ullrich, and C. Uhl, "Quality-dependent deep learning for safe autonomous guidewire navigation," *Curr. Directions Biomed. Eng.*, vol. 8, no. 1, pp. 21–24, 2022.
[22] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
[23] Y. Cho, J.-H. Park, J. Choi, and D. E. Chang, "Sim-to-real transfer of image-based autonomous guidewire navigation trained by deep deterministic policy gradient with behavior cloning for fast learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.* 2022, pp. 3468–3475.
[24] H. You, E. Bae, Y. Moon, J. Kweon, and J. Choi, "Automatic control of cardiac ablation catheter with deep reinforcement learning method," *J. Mech. Sci. Technol.*, vol. 33, pp. 5415–5423, 2019.
[25] Y. Cho, J.-H. Park, J. Choi, and D. E. Chang, "Image processing based autonomous guidewire navigation in percutaneous coronary intervention," in *Proc. IEEE Int. Conf. Consum. Electron.-Asia*, 2021, pp. 1–6.
[26] J. Hunt and J. Hunt, "Introduction to games programming," *Adv. Guide to Python 3 Program.*, pp. 121–123, 2019. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-25943-3_11#citeas
[27] J. Shen, E. Xiao, Y. Liu, and C. Feng, "A deep reinforcement learning environment for particle robot navigation and object manipulation," in *Proc. Int. Conf. Robot. Autom.*, 2022, pp. 6232–6239.
[28] L. Ding and A. Goshtasby, "On the canny edge detector," *Pattern Recognit.*, vol. 34, no. 3, pp. 721–725, 2001.
[29] J. Kweon et al., "Deep reinforcement learning for guidewire navigation in coronary artery phantom," *IEEE Access*, vol. 9, pp. 166409–166422, 2021.
[30] D. Omeiza, S. Speakman, C. Cintas, and K. Weldermariam, "Smooth grad-CAM : An enhanced inference level visualization technique for deep convolutional neural network models," 2019, *arXiv:1908.01224*.
[31] I. Akkaya et al., "Solving rubik's cube with a robot hand," 2019, *arXiv:1910.07113*.

**Ziyang Mei** received the B.E. degree in mechanical design manufacturing and automation and the M.S. degree in mechanical electronic engineering in 2019 and 2022, respectively, from Xiamen University, Fujian, China, where he is currently working toward the Ph.D. degree in intelligent science and technology with the Institute of Artificial Intelligence.

His primary research focus is artificial intelligence and robotics.

**Jiayi Wei** received the B.E. degree in computer science and technology from Tianjin University, Tianjin, China, in 2021. She is currently working toward the M.S. degree in artificial intelligence with the Institute of Artificial Intelligence, Xiamen University, Fujian, China.

Her primary research focus is intelligent guidewire navigation for vascular interventional surgical robots.

**Si Pan** is currently working toward the B.E. degree in mechanical design manufacturing and automation with the School of Aerospace Engineering, Xiamen University, Fujian, China.

His primary research focus is artificial intelligence and robotics.

**Haoyun Wang** received the B.E. degree in software engineering from Qufu Normal University, Shandong, China, in 2022. She is currently working toward the M.S. degree in artificial intelligence with the Institute of Artificial Intelligence, Xiamen University, Fujian, China.

Her primary research focus is video instance segmentation.

**Dezhi Wu** received the Ph.D. degree in measuring and testing technologies and instruments from Xiamen University, Fujian, China, in 2009.

He was a visiting scholar with the University of California, Berkeley, Berkeley, CA, USA, from 2015 to 2016 and is currently a Professor with Xiamen University. His primary research interests are intelligent soft robots, micro–nano manufacturing equipment and humidity sensors.

**Yang Zhao** received the B.Eng. degree in mechanical engineering automation, the M.Eng. and Ph.D. degrees in mechanical manufacturing and automation from Jilin University, Jilin, China, in 2003, 2006, and 2009, respectively.

He is currently an Associate Professor with Xiamen University, Xiamen, China, with primary research interests in intelligent medical robots and artificial intelligence.

**Gang Liu** received the Ph.D. degree in biomedical engineering from Sichuan University, Chengdu, China, in 2009.

He continued his research on nanomedicine and molecular imaging with the National Institutes of Biomedical Imaging and Bioengineering, National Institutes of Health. He is currently the Professor of the Center for Molecular Imaging and Translational Medicine, Xiamen University, Xiamen, China. His research interests include biomaterials, medical robotics, and molecular imaging.

**Shuxiang Guo** (Fellow, IEEE) received the Ph.D. degree in mechanical engineering science from Nagoya University, Japan, in 1995. He is currently the Chair Professor with the Southern University of Science and Technology, Shenzhen, China. He is also the Chair Professor with the Beijing Institute of Technology, Beijing, China.

Prof. Guo has a fellowship of the Engineering Academy of Japan.