This paper appears in the Journal of Medical Robotics Research, 2023.
https://doi.org/10.1142/S2424905X23400044

Robot Learning Incorporating Human Interventions in the Real World for Autonomous Surgical Endoscopic Camera Control

Yafei Ou, Sadra Zargarzadeh, Mahdi Tavakoli

Department of Electrical and Computer Engineering, University of Alberta, 9211-116 Street NW, Edmonton, AB, T6G 1H9, Canada
E-mail: yafei.ou@ualberta.ca (corresponding author); sadra.zar@ualberta.ca; mahdi.tavakoli@ualberta.ca

Recent studies in surgical robotics have focused on automating common surgical subtasks such as grasping and manipulation using deep reinforcement learning (DRL). In this work, we consider surgical endoscopic camera control for object tracking – e.g., using the endoscopic camera manipulator (ECM) from the da Vinci Research Kit (dVRK) (Intuitive Inc., Sunnyvale, CA, USA) – as a typical surgical robot learning task. A DRL policy for controlling the robot joint space movements is first trained in a simulation environment and then continues the learning in the real world. To speed up training and avoid significant failures (in this case, losing view of the object), human interventions are incorporated into the training process and regular DRL is combined with generative adversarial imitation learning (GAIL) to encourage imitating human behaviors. Experiments show that an average reward of 159.8 can be achieved within 1,000 steps compared to only 121.8 without human interventions, and the view of the moving object is lost only twice during the training process out of 3 trials. These results show that human interventions can improve learning speed and significantly reduce failures during the training process.

Keywords: Surgical Autonomy; Reinforcement Learning; Human-in-the-Loop Deep Reinforcement Learning; Learning from Demonstration.

1. Introduction

Several recent advances in surgical robotics focus on automating common surgical subtasks such as grasping, suturing, and tissue manipulation [1–5] to reduce the workload of the surgeons. Furthermore, these studies lay the groundwork for increasing the level of surgical robot autonomy, as the automation of these subtasks can serve as the low-level robot skills needed for achieving a higher level of autonomy such as task reasoning and planning. In this context, deep reinforcement learning (DRL) which uses deep neural networks as function approximators in reinforcement learning (RL), is becoming increasingly popular for learning to automate surgical subtasks, largely due to their high generalizability and less need for human knowledge.

DRL has helped achieve high-level autonomy in other fields, including general robot manipulation and unmanned vehicles. Typically, an RL agent explores the environment by starting with random actions and gradually improves its decision policy to take better actions, based on the reward feedback from the environment. Since it requires a large number of explorations before learning an effective policy, one typical procedure to train an RL agent is the "simulation-to-reality" (sim-to-real) technique where an agent is first trained in a simulated environment and then transferred to the real world. While this approach has achieved some promising results in surgical robot learning [2–5], the level of autonomy that has been achieved and the tasks that have been automated are still limited compared to other fields such as autonomous vehicles where research is advancing towards Level 5 autonomy [6]. One of the reasons is the lack of high-fidelity simulators for surgical environments in which an RL agent can be trained. Therefore, even after a successful policy is learned in the simulator, it can suffer from performance degradations when transferred to the real world due to the sim-to-real gap, such as inaccurate robot dynamics or registration [2].

Continuing the learning process in the real world, specifically fine-tuning a pre-trained policy, is a logical step when applying DRL to real-world tasks. Although some sim-to-real techniques such as domain randomization [7] and utilizing offline RL approaches [8] aim at eliminating the need for online exploration in the real world, fine-tuning with online experiences in the real world is still a common strategy for improving the performance of a simulator-trained policy. However, learning in the real world for surgical robots is usually considered impractical, since unsafe robot actions during the learning agent's exploration

can cause significant damage to the environment, which is particularly undesirable in surgeries. Furthermore, even though fine-tuning a pre-trained model requires less exploration than training from scratch, a relatively large number of samples from the real world are still needed in general. Therefore, sample efficiency and safety considerations are two of the major challenges in surgical robot learning.

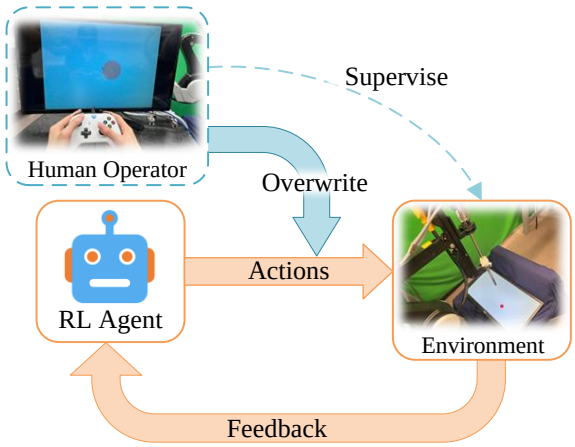

Fig. 1. A human supervisor can intervene during the training process of an RL agent by overwriting the actions.

One approach to both expediting learning and preventing unsafe actions is leveraging real-time human interventions [9]. In this case, a human supervisor monitors the training process and overwrites the agent's action when necessary, as shown in Fig. 1. This can be viewed as intermittent human demonstrations that happen throughout the training. However, unlike LfD, real-time human interventions in RL allow the human to provide important demonstrations only when needed without having to demonstrate the entire task.

While directly overwriting agent actions is impractical in some cases such as when a legged robot is learning to walk, this approach is more intuitive for the human supervisor if the task is traditionally completed by a human, such as teleoperation or driving, compared to providing evaluative feedback or demonstrating unsafe actions. Furthermore, although interventions are more demanding of skills for the human, the actions taken by the human are usually informative and can make the learning process more effective compared to evaluative feedback. In addition, safety is much more assured since preventing dangerous actions is not dependent on an algorithm or a trained model. As a result, it is particularly suitable for use in surgical robot learning. While prior work has shown success in applying DRL with real-time human interventions in problems such as unmanned aerial vehicles [10] and autonomous driving [11,12], its application in surgical robot learning has not been extensively studied. Recent work has attempted to incorporate human interactions in surgical robot learning environments [13]. However, it only considers human interaction as full task demonstrations, and is limited to simulation environments.

In this work, we consider surgical endoscope navigation as a typical surgical support task that could be learnt. A training framework based on our earlier work, [14], that combines DRL with real-time human interventions through the incorporation of generative adversarial imitation learning (GAIL) is applied to learn to automate endoscopic camera control for tracking a moving object in the real world. A DRL agent is first trained in the simulator with different robot and camera configurations than in the real world and fine-tuned in the real world in the presence of possible real-time human interventions. The main contributions of this work are: (a) we propose a DRL framework that enables learning endoscopic camera control first in a simulation environment and then transferred to the real world with fine-tuning for the endoscopic camera manipulator (ECM) of da Vinci Research Kit (dVRK) [15]; (b) we propose a training methodology that utilizes real-time human interventions by combining regular RL with GAIL to accelerate training and prevent catastrophic failures when the agent is fine-tuned in the real world. We validate the proposed framework and show that the endoscopic camera navigation task can be learned in the real world with few failures by utilizing human interventions. To the best of our knowledge, this is the first time a surgical support task is learned through human intervention in the real world using the dVRK.

This paper is organized as follows. Section 2 briefly reviews the related studies. Section 3 introduces the proposed method. In Section 4, we present the results obtained from the simulation environment, and in Section 5 the real-world experimental setup is discussed. Section 6 outlines and discusses the results. Section 7 concludes the paper with discussions on the limitations and potential future work.

2. Related Work

2.1. *Incorporating Human Knowledge in RL*

Leveraging human knowledge is an intuitive method to accelerate learning or prevent unsafe actions. For instance, using behavior cloning (BC), a naive learning from demonstration (LfD) approach, to pre-train the policy in a supervised manner is a common strategy used to accelerate learning. Additionally, humans can observe the training process and provide feedback on how good a given robot action is, which can be used directly to update the policy [16] or as an auxiliary reward to speed up the training process [17]. Humans can also provide initial demonstrations of *unsafe* actions for training a *safety critic* that prevents the agent from taking dangerous actions in the exploration [18].

Leveraging human interventions is one other effective strategy for both preventing catastrophic incidents during exploration and enhancing sample efficiency. Saunders et al. introduced a training mechanism that involves human

monitoring of the training process [9]. When the agent is in a dangerous situation, the human intervenes and overrides the agent's actions, resulting in a penalty in the reward function. Furthermore, an action blocker is trained using human interventions, enabling it to automatically block risky actions of the agent and eventually this action blocker replaces the human supervisor. Similarly, Xu et al. proposed a safe model-based RL framework by incorporating human interventions, where an action blocker is trained to mimic the human's blocking decisions [19]. Not only can interventions happen when the agent is acting dangerously, but they can also be used for assisting the agent to learn more efficiently. In [20], the human provides interventions to help overcome the bottleneck of inserting a pod into a machine slot. Wang et al. developed an algorithm for safe RL with human interventions by adding a BC loss to the original policy loss of proximal policy optimization (PPO), which accelerates the training process in the meantime [10].

2.2. *DRL and LfD for surgical robot autonomy*

The use of DRL and LfD in surgical autonomy has gained increasing attention in recent years. A number of recent studies have examined the automation of common subtasks that frequently occur during surgeries, such as knot-tying [1], needle hand-over [3, 4] and tissue manipulation [2, 5], with the goal of relieving surgeons from repetitive and monotonous tasks. For example, Tagliabue et al. trained a policy in a simulated environment using PPO for the robot to grasp and lift the tissue and reveal a region of interest underneath it, and validated the performance in a real-world setup [2]. Osa et al. used LfD for planning the motion trajectories to achieve autonomous knot-tying [1]. In addition, DRL and LfD approaches have also been investigated for achieving shared autonomy and control in robotic surgery. For instance, Zhu et al. proposed a DRL-based semi-autonomous control framework for peg transfer, where the coarse control is automated by the agent while the user only needs to focus on fine control and make decisions at critical points [21]. Zhang et al. applied dynamic movement primitives (DMP), an LfD approach to achieve shared control in a peg transfer task [22].

LfD is frequently integrated into DRL to enhance performance as it capitalizes on demonstrations provided by human experts. For instance, in [3], BC is used to help the exploration of a deep deterministic policy gradients (DDPG) agent in learning bimanual needle regrasping. In a follow-up study of [2], Pore et al. combined PPO with GAIL using human demonstrations to achieve a faster learning speed [23]. While these approaches involve integrating LfD into DRL by utilizing human demonstrations gathered before training, our work allows the human to start and stop intervention at any point during the training process.

2.3. *Autonomous endoscopic camera control*

As a surgical support task, adjusting the motion of the endoscopic camera is traditionally carried out by the surgeons to realign the field of view in accordance with the surgical procedure during surgery. Therefore, automating the task could potentially reduce the burden on the surgeons.

Autonomous endoscopic camera control has been thoroughly explored in a number of studies. Traditional approaches focus on hand-crafting or developing knowledge-based rules for controlling the camera motion based on surgical tool position [24–28] and eye gaze [29, 30]. Separate feature extraction pipelines, such as detecting surgical tools, as well as human knowledge about the task are usually required for these approaches to be successful.

Recent advances show the potential of utilizing data-driven approaches, especially LfD methods to automate endoscopic camera control [31–34]. In [32], the authors utilized inverse reinforcement learning (IRL), an LfD approach, to learn the task from expert trajectories. Li et al. used GAIL for learning an end-to-end policy that takes the image feedback from the endoscopic camera as the input to generate camera motions directly from the endoscopic videos recorded during surgery [33]. A supervised learning approach has also been explored in [34], where a sequence-to-sequence recurrent network was trained to generate future camera movements based on previous motions. This type of work learns an endoscopic camera control policy from prior expert demonstrations or video recordings. In contrast to these studies that utilize LfD approaches, we consider endoscopic camera control as an RL problem, where an agent is trained first in a simulator from scratch, and then fine-tuned in the real world.

3. Methods

3.1. *Soft actor-critic*

RL typically addresses the problem of a Markov decision process (MDP) defined as $\langle \mathcal{S}, \mathcal{A}, P, r, \gamma \rangle$. $\mathcal{S}$ and $\mathcal{A}$ are the state and action space, respectively. The corresponding state and action variables are $\mathbf{s}$ and $\mathbf{a}$. $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the state transition function which maps a state-action pair $(\mathbf{s}_t, \mathbf{a}_t)$ at time step t to the next state $\mathbf{s}_{t+1}$. $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function that relates a state-action pair $(\mathbf{s}_t, \mathbf{a}_t)$ to a reward value. $\gamma \in [0, 1]$ is the discount factor.

Since off-policy algorithms allow the behavior policy used for collecting experience to be different from the target policy being learned, experience replay can be utilized to reuse previous experiences during training, which improves sample efficiency compared with on-policy algorithms. Additionally, this naturally allows for human interventions during exploration, since the human policy is essentially different than the target policy. It is therefore more appropriate to use off-policy algorithms in this case.

We use soft actor-critic (SAC) [35], an off-policy DRL algorithm that considers the maximum entropy reinforce-

ment learning problem. The learning objective is to find an optimal policy that maximizes the expectation of the discounted return and the policy entropy at the same time:

$$\pi^* = \arg\max_{\pi} \sum_{t=0}^{T} \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\rho_\pi} \left[\gamma^t r(\mathbf{s}_t,\mathbf{a}_t) + \alpha\mathcal{H}(\pi(\cdot|\mathbf{s}_t))\right] \tag{1}$$

Here, π is the policy, ρ_π is the trajectory distribution produced by the policy π, T is the horizon, $\mathcal{H}(\pi(\cdot|\mathbf{a}_t))$ is the entropy of the action distribution, and α is a weighting factor. Taking into account maximum entropy encourages exploration and enables learning a more robust policy [35].

SAC exploits an actor-critic structure. The critic is a neural network $Q_\theta(\mathbf{s}_t,\mathbf{a}_t)$ with trainable parameters θ for estimating the soft Q-value, and the actor is the policy π_ϕ parameterized by ϕ that generates actions from given states. $Q_\theta(\mathbf{s}_t,\mathbf{a}_t)$ is trained by minimizing the Bellman residual

$$J_Q(\theta) = \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\mathcal{R}} \left[\frac{1}{2}\left(Q_\theta(\mathbf{s}_t,\mathbf{a}_t) - \hat{y}_t\right)^2\right] \tag{2}$$

where $\mathcal{R}$ is the trajectories stored in the experience replay buffer, and

$$\hat{y}_t = r(\mathbf{s}_t,\mathbf{a}_t) + \gamma\mathbb{E}_{\mathbf{s}_{t+1}\sim p}[V_\theta(\mathbf{s}_{t+1})] \tag{3}$$

is the temporal difference (TD) target, with $V_\theta(\mathbf{s}_t)$ being the soft state-value function implicitly parameterized by θ.

The policy π_ϕ is trained by maximizing the sum of the soft Q-value predicted by the critic and the α-weighted policy entropy, i.e. minimizing the loss

$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_t\sim\mathcal{R}}\left[\mathbb{E}_{\mathbf{a}_t\sim\pi_\phi}\left[-Q_\theta(\mathbf{s}_t,\mathbf{a}_t) + \alpha\log(\pi(\mathbf{a}_t|\mathbf{s}_t))\right]\right] \tag{4}$$

In practice, two Q networks (Q_{θ_1} and Q_{θ_2}) and two target networks ($Q_{\bar{\theta}_1}$ and $Q_{\bar{\theta}_2}$) are used to mitigate the overestimation problem and stabilize training.

3.2. *Generative adversarial imitation learning*

Generative adversarial imitation learning (GAIL) is an LfD algorithm that is based on RL and generative adversarial networks (GANs). In GAIL, the reward function of the task is unknown. As an alternative, successful trajectories from human experts are collected as demonstrations that an RL agent should imitate. During training, a discriminator $D_\varphi(\mathbf{s}_t,\mathbf{a}_t)$ is jointly trained with an RL agent to discriminate between expert human actions and the actions taken by the learning agent. While the true reward function of the task is unknown, the RL agent uses a value predicted by the discriminator as the reward instead ("surrogate reward"), which indicates how similar an action is to the human expert.

GAIL is originally implemented for on-policy algorithms [36], but can be extended to off-policy algorithms as well [37–39]. In off-policy GAIL, the expert demonstrations are stored in the dataset $\mathcal{D}$, and the trajectories generated by the agent during training are stored in the replay buffer $\mathcal{R}$. The discriminator and the RL agent are trained in an adversarial manner and both of them improve eventually during the training process. The discriminator is trained by minimizing the loss

$$\begin{aligned} J_D^{GAIL}(\varphi) = &\mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\mathcal{D}}\left[\log D_\varphi(\mathbf{s}_t,\mathbf{a}_t)\right] + \\ &\mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\mathcal{R}}\left[\log(1 - D_\varphi(\mathbf{s}_t,\mathbf{a}_t))\right] \end{aligned} \tag{5}$$

In practice, gradient penalization is used to encourage the Lipshizness of D_φ, which is essential for successful off-policy GAIL [39].

Three common forms of rewards are often used as the surrogate reward function for the RL agent: $r(\mathbf{s}_t,\mathbf{a}_t) = -\log(1-D_\varphi(\mathbf{s}_t,\mathbf{a}_t))$, $r(\mathbf{s}_t,\mathbf{a}_t) = \log(D_\varphi(\mathbf{s}_t,\mathbf{a}_t))$, or $r(\mathbf{s}_t,\mathbf{a}_t) = \log(D_\varphi(\mathbf{s}_t,\mathbf{a}_t)) - \log(1-D_\varphi(\mathbf{s}_t,\mathbf{a}_t))$. As in regular RL, the surrogate reward function is used for training the agent.

3.3. *Incorporating human interventions in SAC with GAIL*

During the training of an RL agent, a human can supervise the process and choose to intervene by directly overwriting the actions taken by the agent, as shown in Fig. 1. The occurrence of human interventions during the training can be formulated as a switching function that is only known by the human supervisor. Thus, the actual action taken at time step t can be expressed by

$$\mathbf{a}_t = \mathcal{I}(\mathbf{s}_t)\mathbf{a}_t^H + (1 - \mathcal{I}(\mathbf{s}_t))\mathbf{a}_t^A \tag{6}$$

where $\mathcal{I}(\mathbf{s}_t) \in \{0,1\}$ is the switching function representing whether the human intervenes or not, $\mathbf{a}_t^H$ is the action proposed by the human, and $\mathbf{a}_t^A$ is the action proposed by the RL agent.

By considering the human interventions as intermittent demonstrations, GAIL can be incorporated into regular RL to encourage the agent to imitate human behavior and accelerate learning. During training, the transitions caused by the agent are stored in the agent replay buffer $\mathcal{R}_A$ the transitions caused by human interventions are stored in a separate replay buffer $\mathcal{R}_H$. Same as GAIL, a discriminator $D_\varphi(\mathbf{s}_t,\mathbf{a}_t)$ is trained jointly with the RL agent to predict whether an action is taken by the human or the agent. Similar to (5), the loss of the discriminator is

$$\begin{aligned} J_D(\varphi) = &\mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\mathcal{R}_H}\left[\log D_\varphi(\mathbf{s}_t,\mathbf{a}_t)\right] + \\ &\mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\mathcal{R}_A}\left[\log(1 - D_\varphi(\mathbf{s}_t,\mathbf{a}_t))\right] \end{aligned} \tag{7}$$

Unlike GAIL, $\mathcal{R}_H$ which stores the human transitions is also changing since more human interventions are added throughout the training process. The reward used for training the agent is then augmented by the GAIL reward:

$$r'(\mathbf{s}_t,\mathbf{a}_t) = (1-\beta)r(\mathbf{s}_t,\mathbf{a}_t) + \beta\, r^{GAIL}(\mathbf{s}_t,\mathbf{a}_t) \tag{8}$$

where β is a weighting factor, r is the actual reward function of the environment, and

$$r^{GAIL}(\mathbf{s}_t,\mathbf{a}_t) = D_\varphi(\mathbf{s}_t,\mathbf{a}_t) \tag{9}$$

is the auxiliary reward predicted by the discriminator. Without applying logarithm, the output of the discriminator $D_\varphi(\mathbf{s}_t, \mathbf{a}_t) \in [0, 1]$ is directly used as the auxiliary reward to avoid large values. In practice, β can be small and decay gradually throughout training. The critic network Q_θ is trained with the augmented reward using the same equation as (2) without any modification.

Similar to [38], an imitation loss term is added to the original policy loss to encourage the policy to generate actions similar to the human expert by making the policy update similar to the training of a GAN generator:

$$J_\pi^{GAIL}(\phi) = J_\pi(\phi) + \mathbb{E}_{\mathbf{s}_t \sim \mathcal{R}_A}\left[\mathbb{E}_{\mathbf{a}_t \sim \pi_\phi}\left[-\omega \log D_\varphi(\mathbf{s}_t, \mathbf{a}_t)\right]\right] \tag{10}$$

where ω is a weighting factor. Therefore, the policy loss of SAC is changed to

$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{R}}\left[\mathbb{E}_{\mathbf{a}_t \sim \pi_\phi}\left[-Q_\theta(\mathbf{s}_t, \mathbf{a}_t) + \alpha \log(\pi(\mathbf{a}_t|\mathbf{s}_t)) - \omega \log D_\varphi(\mathbf{s}_t, \mathbf{a}_t)\right]\right] \tag{11}$$

We name the proposed RL framework with human interventions "RLHI-SAC". Fig. 2 shows an overview of the framework and the detailed procedure is summarized in Algorithm 3.1.

Algorithm 3.1. RLHI-SAC

Initialize actor network π_ϕ, critic networks Q_{θ_1}, Q_{θ_2}, discriminator network D_φ
Initialize target networks $Q_{\bar{\theta}_1} = Q_{\theta_1}$, $Q_{\bar{\theta}_2} = Q_{\theta_2}$
Initialize empty human replay buffer $\mathcal{R}_E$ and empty agent replay buffer $\mathcal{R}_A$, $\mathcal{R} \equiv \mathcal{R}_E \cup \mathcal{R}_A$
for each iteration **do**
  for each environment step **do**
    $\mathbf{a}_t^A \sim \pi_\phi(\mathbf{s}_t)$
    if human intervenes **then**
      $\mathbf{a}_t \leftarrow \mathbf{a}_t^H$
      $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$
      $\mathcal{R}_H \leftarrow \mathcal{R}_H \cup \{\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1}\}$
    else
      $\mathbf{a}_t \leftarrow \mathbf{a}_t^A$
      $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$
      $\mathcal{R}_A \leftarrow \mathcal{R}_A \cup \{\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1}\}$
    end if
  end for
  if train discriminator now **then**
    for each discriminator gradient step **do**
      Update $\mathcal{D}_\varphi$ using (7)
    end for
  for each policy gradient step **do**
    Sample $\{(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, r_t)\} \sim \mathcal{R}$
    Augment the reward using (8)
    Update Q_{θ_1}, Q_{θ_2} using (2)
    Update π_ϕ using Equation (11)
    $\bar{\theta}_i \leftarrow \tau\theta_i + (1-\tau)\bar{\theta}_i$ for $i \in \{1, 2\}$
  end for
end for

4. Simulation Results

To validate the proposed method (RLHI-SAC), we train agents in the simulator using different approaches and compare their performances. The task to learn is endoscopic camera control where the endoscopic camera manipulator (ECM) from the dVRK should move and keep tracking a moving object in a plane, as introduced in SurRol [40]. Fig. 3 shows the original ECM ActiveTrack tasks developed in [40].

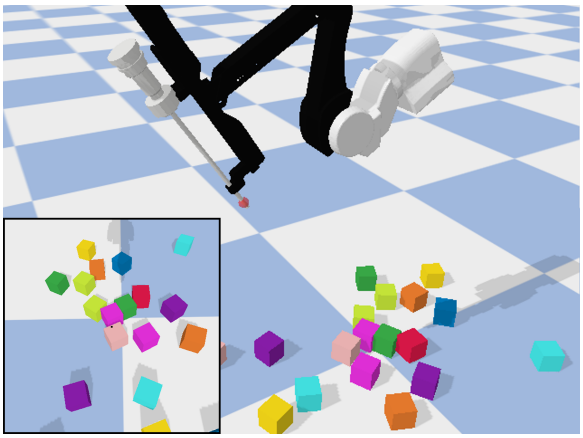

Fig. 3. ECM ActiveTrack from SurRol. Bottom-left: image captured from the simulated endoscopic camera.

As part of method validation, in this section, we use the original environment setup without modification. In the original environment of ActiveTrack, the action is the camera velocity in its own frame coordinate ${}^cV_c$. The observation is the robot and the object poses in the Cartesian space. The reward function is

$$r(\mathbf{s}_t, \mathbf{a}_t) = C - (\|p_t^{ij} - p_c\|_2 + \lambda \cdot |\theta^*|) \tag{12}$$

where $C = 1$ is a constant and $\lambda = 0.1$ is a weighting factor, p_t^{ij} is the normalized position of the tracked object in the image, and p_c is the image center.

We compare RLHI-SAC with the following baseline approaches:

IA-SAC: Intervention-aided reinforcement learning (IARL) is derived from [10, 11], where a behavior cloning loss (BC loss) is added to the policy loss for the human-intervened state-action pairs to encourage the agent to generate actions close to the human actions. It was originally implemented based on PPO and was reimplemented for DDPG in [11]. In this work, we reimplement this method based on SAC and name it IA-SAC.

HI-SAC: Human intervention reinforcement learning (HIRL) is derived from [9]. This method also allows a human to directly overwrite agent actions. However, no modification is made to the learning algorithm and human actions are treated in the same manner as agent actions. We

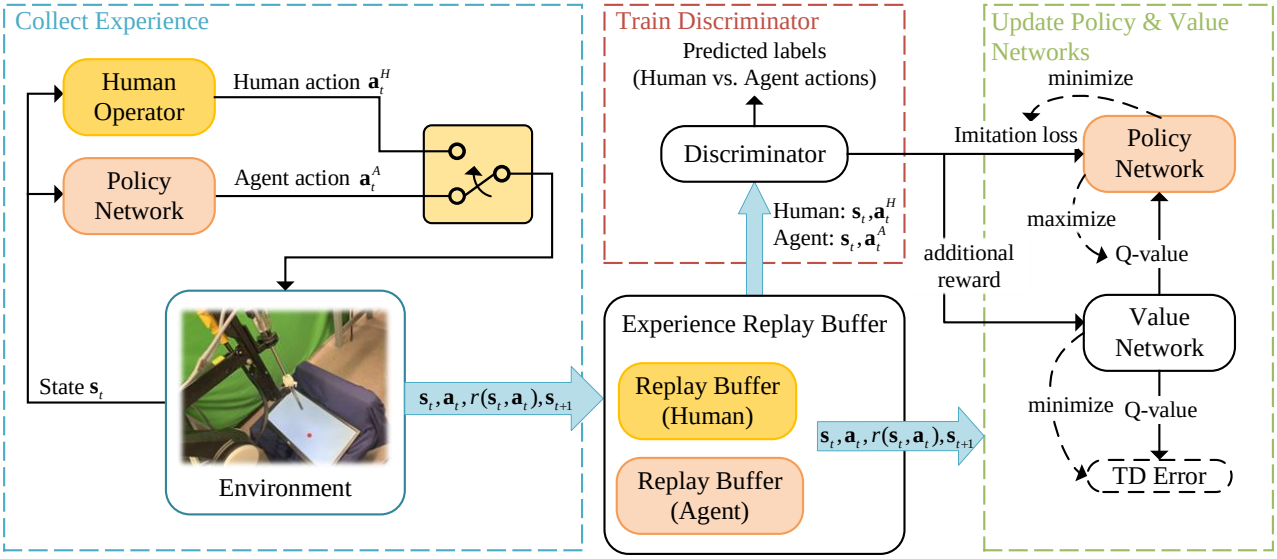

Fig. 2. Framework of the proposed method with human interventions.

re-implement this method based on SAC and name it HI-SAC.

Standard SAC without human interventions ("**Unguided**") is also implemented as a comparison to assess the effectiveness of adding human interventions. It is worth noting that in some of the related methods, training techniques such as penalizing human interventions, auto-tuning weighting parameters, and non-uniform data sampling strategies are employed. However, since these techniques are generally applicable to the proposed and the compared methods, they are not implemented and compared for simplicity and fair comparison.

A total of 5,000 steps are trained for each method and repeated for 3 trials, and each trial consists of at most 500 steps of human interventions. Multilayer perceptrons (MLPs) are used as the policy and value networks, and the learning rate is set to be 3×10^{-4}. The GAIL reward weight is $\beta = 0.2$ and the imitation weight is $\omega = 4$, both of which decay exponentially over time. Same as in [40], the maximum allowed number of steps is 500 during training. However, during evaluation, it is set to 200 to discard the repetitive trajectories of the moving object for faster evaluation. The learning curves are shown in Fig. 4. RLHI-SAC achieves a faster learning speed in general compared to the other approaches and outperforms both HI-SAC and IA-SAC. IA-SAC reaches a higher average return compared to other approaches during the initial training stage, possibly due to the effect of BC as it is known to be able to encourage the policy to imitate human actions rapidly. However, in the long term, there is no significant difference between IA-SAC and HI-SAC in this specific task and IA-SAC reaches even a lower average return compared to HI-SAC at 5,000 steps. Despite this, both IA-SAC and HI-SAC with human interventions achieve better results than "Unguided", which does not include any human interventions. The differences in the learning curves between RLHI-SAC and HI-SAC also show the effectiveness of incorporating GAIL for imitating human behavior, as HI-SAC can be viewed as an ablation of RLHI-SAC and IA-SAC that does not include any imitation components.

RLHI-SAC learns faster and outperforms standard SAC without human interventions and HI-SAC where a BC loss is added for imitating human behaviors.

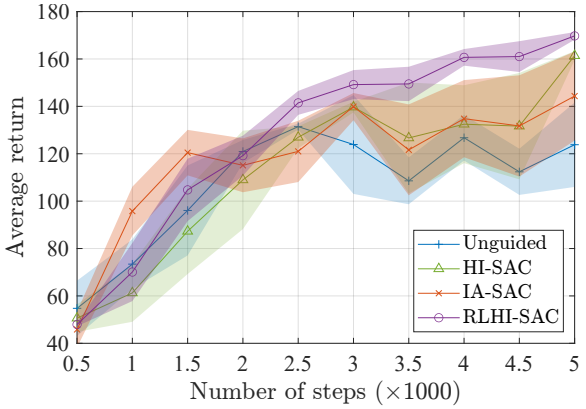

Fig. 4. Evaluation performance of the trained models in simulation experiments. The solid line is the mean value and the shaded area represents half of a standard deviation across the three trials.

5. Experimental Setup

5.1. *Learning endoscopic camera control for object tracking in the real world*

Different from Section 4, we now focus on learning the task of tracking the moving object directly in the joint space of the robot, which is more practical since the image Jacobian is camera-specific and the mapping from image space motion to the joint space actions is also dependent on the camera mounting configuration. Learning a direct mapping from the observation to the joint space motion of the robot eliminates the need for extra calibration and the calculation of the Jacobians at each step. In this case, the observation is the same as that in the ActiveTrack task, but the actions are now changed to the joint movements of the robot. The reward function is similar to that in the ActiveTrack task:

$$r(\mathbf{s}_t, \mathbf{a}_t) = 1 - \left(\frac{\|p_t^{ij} - p_c\|_2}{\sqrt{2}} + 0.1 \cdot |\theta^*| \right) \tag{13}$$

A simulation environment for learning in the joint space is built, as shown in Fig. 5. A regular RL agent is first trained in the simulator without human interventions and then transferred to the real world to continue training with human interventions. To widen the sim-to-real gap, making transferring from simulation to the real world more challenging, and testing the effectiveness of the proposed approach, the camera orientation in the simulator is set to be 90 degrees different than that in the real world along its Z-axis.. Furthermore, the distance between the robot and the plane where the object moves, and the range and speed of the object's movement are different in the simulator and in the real world. This simulates a practical surgical robot learning situation where the simulation in which the agent is initially trained is different from the real world due to various factors such as different camera configurations and reconstruction errors.

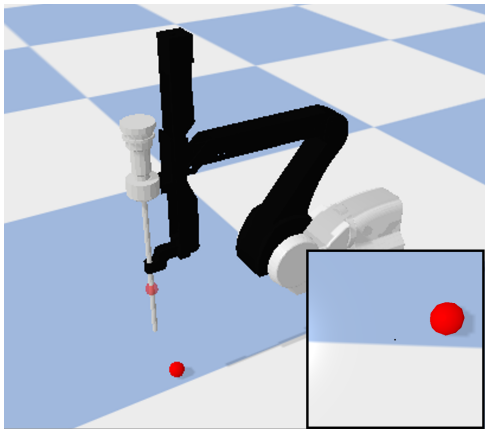

Fig. 5. Simulation environment for learning endoscopic camera control in the joint space. Bottom-right: image captured from the simulated endoscopic camera.

During training in the real world, human intervention is utilized to prevent failures, which happens when the object is completely out of the view of the camera image since tracking has to be terminated in this case. While it is difficult for the human to provide correct actions in the joint space, we use a naive visual servoing approach to allow the human to act in the camera frame and map the actions to the joint space. This is essentially the same as the case when a human holds an impedance-controlled endoscope camera holder robot and moves it directly in the task environment. It is worth noting that the image and robot Jacobians for this visual servoing purpose are only used for mapping the human actions to the joint space, and the learning agent does not have access to them.

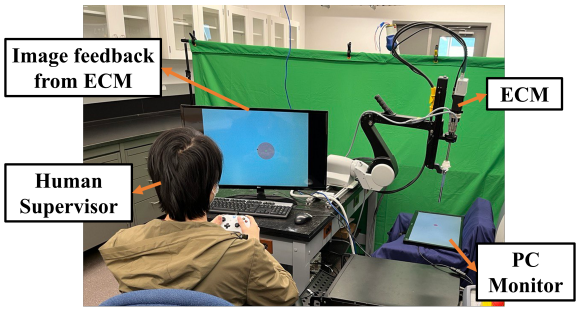

Fig. 6. Real-world training setup for learning endoscopic camera control.

The real-world training setup is shown in Fig. 6, where the tracked object (a red circle) is displayed on a 22-inch computer monitor (LG Corporation, Seoul, South Korea) placed underneath the ECM robot. The random movement trajectories of the object are generated using the same method as in [40]. A naive image processing approach using adaptive thresholding and contouring is applied to detect the red circle and its centroid. During training, a human supervisor watches the monitor and proposes expert actions occasionally through an Xbox Wireless Controller (Microsoft Corporation, Redmond, WA, USA) by pressing a button and moving the two joysticks. Three inputs are recorded from the joysticks, representing the movement of the camera along the three axes of its own 3D frame. As discussed previously, these inputs are mapped to the joint space movements of the robot by utilizing the image and robot Jacobians. These mapped joint space actions are then viewed as the expert interventions and stored in the replay buffer as discussed in Section 3.

We build the RL environment of endoscopic camera control using the real dVRK ECM robot with OpenAI Gym-like interfaces [41]. This enables a smooth transfer when fine-tuning a trained model from the simulator. On top of the existing dVRK software [15], changes have been made to manually handle some of the safety violations imposed by the low-level controller to make the behavior

consistent with the simulation environments, including the joint space limits and Cartesian workspace limits. When the robot reaches the joint or Cartesian limits in some DoFs, instead of raising errors and stopping moving the robot, the modification allows the robot to still move in the other DoFs that have not reached the limits.

5.2. *Training settings*

An RL agent is first pre-trained in the simulator for 10,000 steps and continues learning in the real world with human interventions for 2,000 steps. It is recorded when the human intervenes and when the object is out of sight during training in the real world. The results are compared to those obtained when the same pre-trained model learns in the real world without human interventions ("Unguided"). The hyper-parameters are the same as in Section 4. The maximum number of steps is 200 in both training and evaluation. Three trials are trained for both with and without human interventions.

6. Results and Discussion

Fig. 7 shows the number of times the view of the moving object is lost during training in the real world. With human interventions, the cases of the object being out of view are significantly fewer than without human interventions. In all three trials with human interventions, this occurred only twice when the human intervened too late.

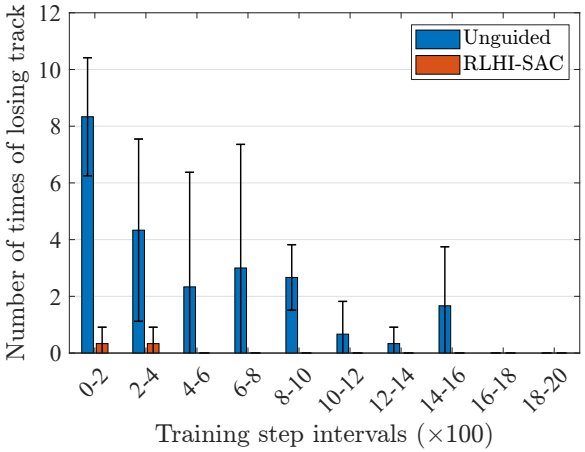

Fig. 7. Number of times of losing track of the moving object during training with and without human interventions within different training step intervals. The error bars represent the standard deviations out of three trials.

The human intervention rate is shown in Fig. 8. It is calculated using the total number of interventions divided by the number of steps within a time interval. As expected, the intervention rate decreases gradually throughout training as the performance of the agent improves.

Since testing in the real world takes a large amount of time, we evaluate the models for only 2 episodes after training for 300, 1,000, and 2,000 steps. As shown in Fig. 9, while there is no significant difference between the cases with and without human interventions during the initial stage of training, the model achieves a much higher average return of around 159.8 within 1,000 steps with human interventions, compared to only 121.8 without human interventions. As the misalignment error $|\theta^*|$ has a relatively small contribution to the reward function (13) compared with the tracking error due to its small weighting factor, a larger accumulative reward suggests that the tracking error is generally lower and the object is closer to the center of the camera frame. Considering that a smaller tracking error is generally associated with a lower chance of losing view of the object in the short term, the results also indicate that by imitating the preventative behavior from human interventions, the trained policy learns to behave in a safe manner at a faster rate than merely relying on reward signals.

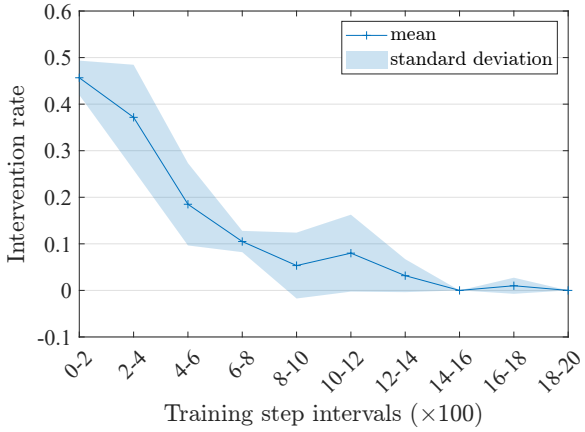

Fig. 8. Intervention rate throughout the training process with human interventions.

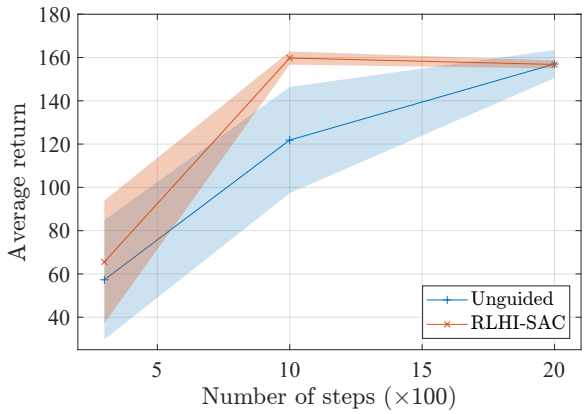

Fig. 9. Evaluation performance of the trained models after

training for 300, 1,000, and 2,000 steps. The solid line is the mean value and the shaded area represents half of a standard deviation.

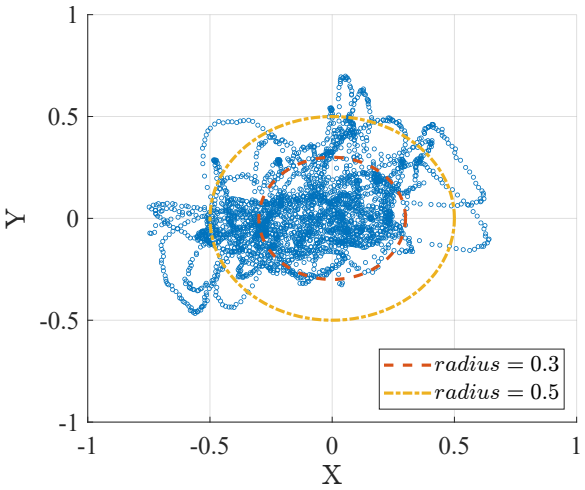

Fig. 10. Normalized object positions in the image frame during tracking across 15 trials (5 trials for each model trained with different seeds). Each point represents the position of the object at one step.

After training for 2,000 steps, we further evaluate each of the 3 models trained with human interventions for 5 episodes and plot the positions of the object centroids in the image frame (normalized) at each step during tracking in all 15 trials, as shown in Fig. 10. The tracking performance in the horizontal direction is not as good as that in the vertical direction, which is related to the higher movement speed of the object horizontally, since it moves in a rectangular region. Around 85% of the points are inside a circle with a radius of 0.5, and 53% are within a radius of 0.3. The root mean square error measured in normalized units is 0.35 (the maximum possible value is $\sqrt{2}$). Fig. 11 shows a sequence of screenshots during one trial.

7. Conclusion

In this work, we presented a DRL method for surgical robot learning to automate endoscopic camera control for moving object tracking. The proposed method leverages human interventions during training to improve the training speed and avoid significant failures. By viewing the human interventions as intermittent demonstrations, regular RL is combined with GAIL, an LfD approach to improve the training process. Experimental results using simulation first show the effectiveness of this approach in accelerating training, and real-world experiments using the real ECM robot are further carried out to show that it can achieve faster learning speed with few significant failures. The human intervention rate decreases throughout the training process. The trained policy can achieve good tracking performance by directly controlling the joint space movements of the robot.

One major limitation of this work is that the task is relatively simple to learn since the position of the object in the image frame is assumed to be known through traditional image processing. End-to-end policies that directly use images as input for generating motion commands are usually more desired. Additionally, the safety consideration in this task is straightforward, while more complicated safety restrictions can exist in complex surgical tasks. However, as an initial attempt to learn directly in the real world using the dVRK, this work has shown the potential of incorporating human interventions in surgical robot learning. Potential challenges and limitations exist if the proposed method is to be applied to more complex scenarios, as the effort of human experts can be demanding and the diverse behaviors of humans for completing a complex task may have adversarial effects on the training. While increasing the number of experts included could potentially mitigate the issues by distributing the workload and providing sufficiently diverse behaviors to overcome the adversarial effects, further studies are needed to investigate the effectiveness of the proposed approach in more complex situations, such as considering image-based policies or complex surgical tasks.

Acknowledgments

This research was supported by the Canada Foundation for Innovation (CFI), the Natural Sciences and Engineering Research Council (NSERC) of Canada, the Canadian Institutes of Health Research (CIHR), the Alberta Jobs, Economy and Innovation Ministry's Major Initiatives Fund to the Center for Autonomous Systems in Strengthening Future Communities, and the China Scholarship Council (CSC).

References

[1] T. Osa, K. Harada, N. Sugita and M. Mitsuishi, Trajectory planning under different initial conditions for surgical task automation by learning from demonstration, *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE (2014), pp. 6507–6513.

[2] 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, *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE (2020), pp. 3261–3266.

[3] Z.-Y. Chiu, F. Richter, E. K. Funk, R. K. Orosco and M. C. Yip, Bimanual regrasping for suture needles us-

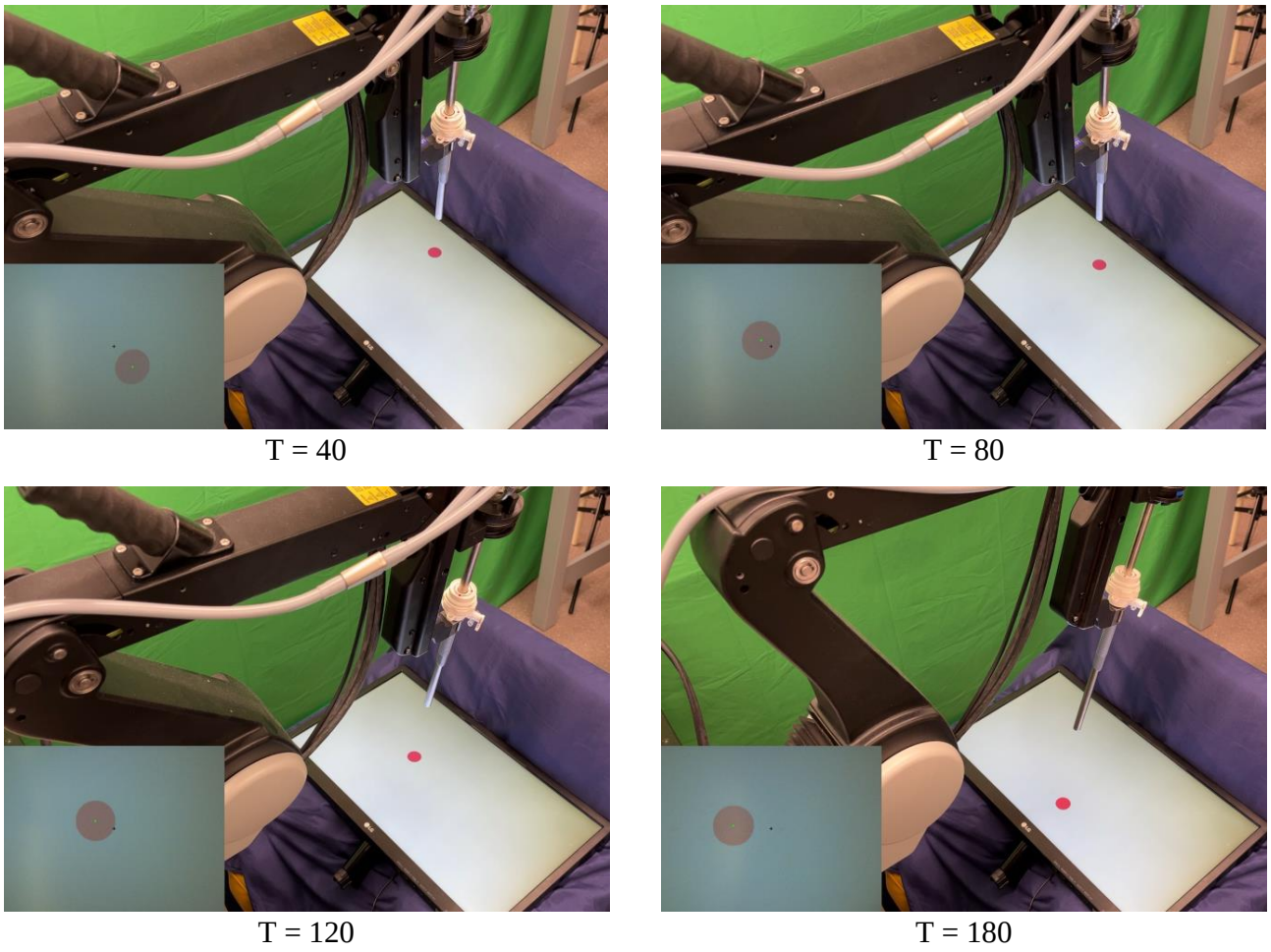

Fig. 11. Screenshots of one trial at different time steps after training for 2,000 steps.

ing reinforcement learning for rapid motion planning, *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE (2021), pp. 7737–7743.

[4] V. M. Varier, D. K. Rajamani, N. Goldfarb, F. Tavakkolmoghaddam, A. Munawar and G. S. Fischer, Collaborative suturing: A reinforcement learning approach to automate hand-off task in suturing for surgical robots, *2020 29th IEEE international conference on robot and human interactive communication (RO-MAN)*, IEEE (2020), pp. 1380–1386.

[5] Y. Ou and M. Tavakoli, Sim-to-real surgical robot learning and autonomous planning for internal tissue points manipulation using reinforcement learning, *IEEE Robotics and Automation Letters* **8**(5) (2023) 2502–2509.

[6] E. Yurtsever, J. Lambert, A. Carballo and K. Takeda, A survey of autonomous driving: Common practices and emerging technologies, *IEEE access* **8** (2020) 58443–58469.

[7] F. Muratore, C. Eilers, M. Gienger and J. Peters, Data-efficient domain randomization with bayesian optimization, *IEEE Robotics and Automation Letters* **6**(2) (2021) 911–918.

[8] R. Julian, B. Swanson, G. S. Sukhatme, S. Levine, C. Finn and K. Hausman, Never stop learning: The effectiveness of fine-tuning in robotic reinforcement learning, *arXiv preprint arXiv:2004.10190* (2020).

[9] W. Saunders, G. Sastry, A. Stuhlmueller and O. Evans, Trial without error: Towards safe reinforcement learning via human intervention (2017).

[10] F. Wang, B. Zhou, K. Chen, T. Fan, X. Zhang, J. Li, H. Tian and J. Pan, Intervention aided reinforcement learning for safe and practical policy optimization in navigation, *Conference on Robot Learning*, PMLR (2018), pp. 410–421.

[11] J. Wu, Z. Huang, Z. Hu and C. Lv, Toward human-in-the-loop ai: Enhancing deep reinforcement learning via real-time human guidance for autonomous driving, *Engineering* (2022).

[12] J. Wu, Z. Huang, W. Huang and C. Lv, Prioritized experience-based reinforcement learning with human guidance for autonomous driving, *IEEE Transactions*

on Neural Networks and Learning Systems (2022).

[13] Y. Long, W. Wei, T. Huang, Y. Wang and Q. Dou, Human-in-the-loop embodied intelligence with interactive simulation environment for surgical robot learning, *IEEE Robotics and Automation Letters* (2023).

[14] Y. Ou and M. Tavakoli, Towards safe and efficient reinforcement learning for surgical robots using real-time human supervision and demonstration, *2023 International Symposium on Medical Robotics (ISMR)*, IEEE (2023), pp. 1–7.

[15] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor and S. P. DiMaio, An open-source research kit for the da vinci® surgical system, *2014 IEEE international conference on robotics and automation (ICRA)*, IEEE (2014), pp. 6434–6439.

[16] S. Griffith, K. Subramanian, J. Scholz, C. L. Isbell and A. L. Thomaz, Policy shaping: Integrating human feedback with reinforcement learning, *Advances in neural information processing systems* **26** (2013).

[17] G. Warnell, N. Waytowich, V. Lawhern and P. Stone, Deep tamer: Interactive agent shaping in high-dimensional state spaces, *Proceedings of the AAAI conference on artificial intelligence*, **32**(1) (2018).

[18] B. Thananjeyan, A. Balakrishna, S. Nair, M. Luo, K. Srinivasan, M. Hwang, J. E. Gonzalez, J. Ibarz, C. Finn and K. Goldberg, Recovery rl: Safe reinforcement learning with learned recovery zones, *IEEE Robotics and Automation Letters* **6**(3) (2021) 4915–4922.

[19] Y. Xu, Z. Liu, G. Duan, J. Zhu, X. Bai and J. Tan, Look before you leap: Safe model-based reinforcement learning with human intervention (2021).

[20] A. Mandlekar, D. Xu, R. Martín-Martín, Y. Zhu, L. Fei-Fei and S. Savarese, Human-in-the-loop imitation learning using remote teleoperation, *arXiv preprint arXiv:2012.06733* (2020).

[21] R. Zhu, D. Zhang and B. Lo, Deep reinforcement learning based semi-autonomous control for robotic surgery (2022).

[22] D. Zhang, Z. Wu, J. Chen, R. Zhu, A. Munawar, B. Xiao, Y. Guan, H. Su, W. Hong, Y. Guo *et al.*, Human-robot shared control for surgical robot based on context-aware sim-to-real adaptation, *2022 International Conference on Robotics and Automation (ICRA)*, IEEE (2022), pp. 7694–7700.

[23] A. Pore, E. Tagliabue, M. Piccinelli, D. Dall'Alba, A. Casals and P. Fiorini, Learning from demonstrations for autonomous soft-tissue retraction, *2021 International Symposium on Medical Robotics (ISMR)*, IEEE (2021), pp. 1–7.

[24] T. Osa, C. Staub and A. Knoll, Framework of automatic robot surgery system using visual servoing, *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE (2010), pp. 1837–1842.

[25] A. Bihlmaier and H. Woern, Automated endoscopic camera guidance: A knowledge-based system towards robot assisted surgery, *ISR/Robotik 2014; 41st International Symposium on Robotics*, VDE (2014), pp. 1–6.

[26] B. Yang, W. Chen, Z. Wang, Y. Lu, J. Mao, H. Wang and Y.-H. Liu, Adaptive fov control of laparoscopes with programmable composed constraints, *IEEE Transactions on Medical Robotics and Bionics* **1**(4) (2019) 206–217.

[27] A. Mariani, G. Colaci, T. Da Col, N. Sanna, E. Vendrame, A. Menciassi and E. De Momi, An experimental comparison towards autonomous camera navigation to optimize training in robot assisted surgery, *IEEE Robotics and Automation Letters* **5**(2) (2020) 1461–1467.

[28] T. Da Col, G. Caccianiga, M. Catellani, A. Mariani, M. Ferro, G. Cordima, E. De Momi, G. Ferrigno and O. De Cobelli, Automating endoscope motion in robotic surgery: a usability study on da vinci-assisted ex vivo neobladder reconstruction, *Frontiers in Robotics and AI* **8** (2021) p. 707704.

[29] K. Fujii, G. Gras, A. Salerno and G.-Z. Yang, Gaze gesture based human robot interaction for laparoscopic surgery, *Medical image analysis* **44** (2018) 196–214.

[30] G. Gras, K. Leibrandt, P. Wisanuvej, P. Giataganas, C. A. Seneci, M. Ye, J. Shang and G.-Z. Yang, Implicit gaze-assisted adaptive motion scaling for highly articulated instrument manipulation, *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE (2017), pp. 4233–4239.

[31] I. Rivas-Blanco, C. J. Perez-del Pulgar, C. López-Casado, E. Bauzano and V. F. Muñoz, Transferring know-how for an autonomous camera robotic assistant, *Electronics* **8**(2) (2019) p. 224.

[32] A. S. Agrawal, Automating endoscopic camera motion for teleoperated minimally invasive surgery using inverse reinforcement learning, PhD thesis, Worcester Polytechnic Institute (2018).

[33] B. Li, R. Wei, J. Xu, B. Lu, C. H. Yee, C. F. Ng, P.-A. Heng, Q. Dou and Y.-H. Liu, 3d perception based imitation learning under limited demonstration for laparoscope control in robotic surgery, *2022 International Conference on Robotics and Automation (ICRA)*, IEEE (2022), pp. 7664–7670.

[34] B. Li, B. Lu, Z. Wang, F. Zhong, Q. Dou and Y.-H. Liu, Learning laparoscope actions via video features for proactive robotic field-of-view control, *IEEE Robotics and Automation Letters* **7**(3) (2022) 6653–6660.

[35] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel *et al.*, Soft actor-critic algorithms and applications, *arXiv preprint arXiv:1812.05905* (2018).

[36] J. Ho and S. Ermon, Generative adversarial imitation learning, *Advances in neural information processing systems* **29** (2016).

[37] I. Kostrikov, K. K. Agrawal, D. Dwibedi, S. Levine and J. Tompson, Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning, *arXiv preprint arXiv:1809.02925*

(2018).
[38] G. Zuo, K. Chen, J. Lu and X. Huang, Deterministic generative adversarial imitation learning, *Neurocomputing* **388** (2020) 60–69.
[39] L. Blondé, P. Strasser and A. Kalousis, Lipschitzness is all you need to tame off-policy generative adversarial imitation learning, *Machine Learning* **111**(4) (2022) 1431–1521.
[40] J. Xu, B. Li, B. Lu, Y.-H. Liu, Q. Dou and P.-A. Heng, Surrol: An open-source reinforcement learning centered and dvrk compatible platform for surgical robot learning, *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE (2021).
[41] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang and W. Zaremba, Openai gym (2016).

Yafei Ou received his B.Sc. degree in Mechanical Design, Manufacturing and Automation from the University of Electronic Science and Technology of China (UESTC) in 2021. He is currently pursuing a Ph.D. degree in Electrical and Computer Engineering at the University of Alberta. His research interests focus on surgical robotics and automation.

Sadra Zargarzadeh received his B.Sc. degree in Mechanical Engineering from Sharif University of Technology, Iran, in 2022. He is currently pursuing an M.Sc. degree in Electrical and Computer Engineering at the University of Alberta. His research interests include medical robotics and computer vision.

Mahdi Tavakoli is a Professor in the Department of Electrical and Computer Engineering, University of Alberta, Canada. He received his BSc and MSc degrees in Electrical Engineering from Ferdowsi University and K.N. Toosi University, Iran, in 1996 and 1999, respectively. He received his PhD degree in Electrical and Computer Engineering from the University of Western Ontario, Canada, in 2005. In 2006, he was a post-doctoral researcher at Canadian Surgical Technologies and Advanced Robotics (CSTAR), Canada. In 2007-2008, he was an NSERC Post-Doctoral Fellow at Harvard University, USA. Dr. Tavakoli's research interests broadly involve the areas of robotics and systems control. Specifically, his research focuses on haptics and teleoperation control, medical robotics, and image-guided surgery. Dr. Tavakoli is the lead author of Haptics for Teleoperated Surgical Robotic Systems (World Scientific, 2008). He is a Senior Member of IEEE and an Associate Editor for IEEE/ASME Transactions on Mechatronics, Journal of Medical Robotics Research, Control Engineering Practice, and Mechatronics