



Robot-Assisted Training in Laparoscopy Using Deep Reinforcement Learning

Xiaoyu Tan , Chin-Boon Chng, Ye Su, Kah-Bin Lim, and Chee-Kong Chui 

Abstract—Minimally invasive surgery (MIS) is increasingly becoming a vital method of reducing surgical trauma and significantly improving postoperative recovery. However, skillful handling of surgical instruments used in MIS, especially for laparoscopy, requires a long period of training and depends highly on the experience of surgeons. This letter presents a new robot-assisted surgical training system which is designed to improve the practical skills of surgeons through intrapractice feedback and demonstration from both human experts and reinforcement learning (RL) agents. This system utilizes proximal policy optimization to learn the control policy in simulation. Subsequently, a generative adversarial imitation learning agent is trained based on both expert demonstrations and learned policies in simulation. This agent then generates demonstration policies on the robot-assisted device for trainees and produces feedback scores during practice. To further acquire surgical tools coordinates and encourage self-oriented practice, a mask region-based convolution neural network is trained to perform the semantic segmentation of surgical tools and targets. To the best of our knowledge, this system is the first robot-assisted laparoscopy training system which utilizes actual surgical tools and leverages deep reinforcement learning to provide demonstration training from both human expert perspectives and RL criterion.

Index Terms—Surgical robotics, laparoscopy, learning from demonstration, deep learning in robotics and automation, AI-based methods.

I. INTRODUCTION

MINIMALLY invasive surgery (MIS) is widely considered to be one of the most important surgical approaches to minimize intra-operative trauma and drastically improve post-operative recovery [1]. This approach has been implemented in multiple surgical disciplines and is also considered as the most reliable approach for preserving organ function and avoiding blood loss [2]. Typically, MIS is achieved through the usage of laparoscopic instruments which involves the creation of several small entrances on the patients skin to establish operation space under the skin for surgical tool manipulation. Visual feedback through laparoscopic systems is leveraged to assist the surgeon during the surgery. However, due to the lack of tactile sensation,

the loss of 3-dimensional direct observation and the disconnected viewpoint between surgical field and surgeons' hands, surgeons need to acquire a totally distinct skill set to handle laparoscopy [1]. As a result, residency training for laparoscopy is both challenging and time-consuming. Although, several surgical robotic systems (e.g., da Vinci and Zeus surgical systems [3], [4]) have been developed to overcome visualization and non-haptic feedback issues in MIS, high costs, operation complexity, and low adoption rates hampers attempts to fully replace traditional laparoscopic approaches and consequently leads to even more time-consuming training program [5], [6].

This letter introduced a new robot-assisted laparoscopy training system to improve surgical tool manipulation skill through exercise and demonstration from both human experts and Reinforcement Learning (RL) criterion. Human experts provide the demonstration with latent patterns in manipulation, while RL agents provide objective-constrained behaviors for demonstration. These two perspectives are equally important and complement each other, allowing trainees to achieve high-accuracy constantly in prolonged and complex operations. First, a deep RL agent is trained in simulation by using Proximal Policy Optimization (PPO) [7]. This agent is utilized to generate demonstration trajectory based on predefined reward signals from dynamics, providing alternate perspectives in training rather than solely replaying the trajectory captured from human experts. Subsequently, a Generative Adversarial Imitation Learning (GAIL) [8] agent is trained based on both PPO generated and expert trajectory. This deep inverse RL agent is trained to involve PPO trajectory, imitate latent patterns in expert demonstration, and overcome the distribution mismatch issue caused by multimodal behaviors of demonstrations [9]. These patterns are difficult to be predefined as reward signals and, consequently, hard to obtain optimal solution under RL criterion. Finally, the well-trained GAIL agent is used to manipulate the robot-assisted device to provide demonstration and deliver feedback to trainees during exercise.

In order to validate the error, provide the distinctive visualization, and improve the diversity of practice procedures, a Mask Region-based Convolution Neural Network (Mask R-CNN) is used to segment and track laparoscopic tools. To provide the direct experience of handling actual surgical tools, a robotic device with clinical laparoscopic tool is designed to record and replay tool trajectory. Based on the experimental results from simulation and practices on the robotic device, our system can successfully learn the manipulation from both experts and simulation, replay the demonstration, and provide the evaluation feedback.

Manuscript received August 15, 2018; accepted December 21, 2018. Date of publication January 7, 2019; date of current version January 17, 2019. This letter was recommended for publication by Associate Editor S. Oh and Editor D. Lee upon evaluation of the reviewers' comments. This work was supported in part by the Ministry of Education Academic Research Fund Tier 1 under Grant R265-000-614-114. (Corresponding author: Xiaoyu Tan.)

The authors are with the Department of Mechanical Engineering, National University of Singapore, Singapore 119077 (e-mail: xiaoyu_tan@u.nus.edu; e0015042@u.nus.edu; e0178254@u.nus.edu; mpelimkb@nus.edu.sg; mpeccck@nus.edu.sg).

Digital Object Identifier 10.1109/LRA.2019.2891311

II. RELATED WORKS

A. Residency Training in Laparoscopy

Typically, residency programs with laparoscopy training include ex-vivo and in-vivo full day laboratory courses [10]. A number of prior works have been reported to perform evaluation and training directly through surgical gestures using Hidden Markov Model (HMM) [11] and Descriptive Curve Coding (DCC) [12]. Although, these procedures can decompose the trajectory structures of MIS, they are context-based methods which are inadequate for discovering underlying features in demonstration [13]. These features may contain unique personal techniques from experts in handling surgical tools, such as choosing specific postures in long operation periods or changing the speed of tools depending on the distance to targets. These features, in particular, cannot be diametrically measured by regular performance metrics such as accuracy and time of accomplishing tasks [14]. Although, recently, Discovery of Deep Option (DDO) [15] and its extension: Discovery of Deep Continuous Option (DDCO) [16] achieved superior results in residency training from demonstration by utilizing deep learning and policy gradient in HMM, these algorithms are unsuitable for use with our training system. This is because the trajectory from dynamic perspectives and determined feedback have to be considered in our proposed system.

B. Motion Planning and Deep Reinforcement Learning

Various motion planning and trajectory optimization algorithms have been developed to perform the manipulation tasks, such as Linear Quadratic Regulator (LQR) [17], Rapidly-exploring Random Trees (RRT) [18] and its variant: RRT* [19] which could guarantee the convergence of optimal solution. However, these model-based methods are designed to find the shortest collision-free path in a transition model and not capable of finding the optimal solution in model-free tasks constrained by dedicated reward function on diverse attributes. Although, some trajectory optimization algorithms developed under RL criterion are model-free, including Guided Policy Search [20], these methods are usually guided or combined with other model-based methods and consequently unsuitable for the proposed laparoscopy training system.

Research on deep model-free RL algorithms has achieved success in learning control policies effectively among complicated interactive environment. David *et al.* proposed different deep RL systems to master the game of Go based on human knowledge and through self-competition without any expert demonstration [21]. Policy optimization algorithms have also been improved to enhance the stability and speed in training the policy agent by constraining the step size of each update [7], [20], [22]. Recently, deep inverse reinforcement learning algorithms have been reported to perform imitation learning purely based on demonstration features [8]. Some deep RL algorithms achieved success in robotic locomotion tasks by learning manipulation of real robots from simulation [23]–[25]. Hence, implementing deep RL algorithms in robot-assisted residency training for laparoscopy can potentially integrate learning from both dynamic objectives represented by reward signals and latent

features which cannot be predefined by rewards. Furthermore, well-trained deep inverse RL agents can provide a baseline for validating trajectory and feedback to trainees.

III. METHODS

In this chapter, the main approaches used in our robot-assisted laparoscopy training system are presented, including simulation environment, robotic device setup, PPO agent training, trajectory correction, expert data collection and learning through GAIL agent. Section III-A presents a high-level overview of our system.

A. System Architecture

The robot-assisted laparoscopy training system is developed to enhance the manipulation skills through practicing and demonstration from both human experts and RL criterion. These two perspectives complement each other because the RL agents will focus on achieving high accuracy in short-term objective-constrained tasks while the experts trajectories may contain long-term overall techniques. For the simplicity of illustrating the validation of our idea on directly utilizing simulated policy with the real robot, we focus our discussion on right-hand motion tasks without pick-and-place movement. Since the kinematics of both robotic tools are designed and developed to be symmetrical, what we have done with the right-hand robotic tool is applicable to the left-hand tool. The trainees will gain their first tactile, and hand-eye coordination experience of operating laparoscopic tools through this exercise. At the initial stage of training, it is critical to establish basic but effective understanding rapidly through demonstration. After that, trainees are encouraged to explore complex operation tasks to enhance the proficiency and form their own techniques without involving demonstration. The flow chart shown in Fig. 1 indicates the major procedures of system construction and implementation from three perspectives: simulation, robotic device, and residents.

B. Robotic Device for Laparoscopy Training

To record and replay demonstration, a robotic device is designed and constructed allowing real clinical laparoscopic tools to be mounted for usage in a physical workspace, recording and mimicking all motions during a laparoscopic surgery within a 60° spherical cone workspace. The device has 4 primary degrees of freedom (DOF), with an optional DOF for actuation of a surgical tool handle. Each DOF is actuated via a brushed DC motor, with an encoder for joint position feedback. The control system of the robotic device consists of a NI 9118 Xilinx Virtex-5 LX110 reconfigurable I/O FPGA, with NI 9505 Full H-Bridge Brushed DC Servo Drive Modules for each motor. This hardware configuration enables high speed control loop execution and high determinism for real time applications. The FPGA-based implementation of the control system provides highly parallel, fast and robust coordination of the robot axes allowing for synchronizing between multiple subsystems of the robotic device. Hence, while manipulating the surgical tools, the tool tip trajectories can be recorded and subsequently used for evaluation and demonstration. By replaying the recorded trajectory

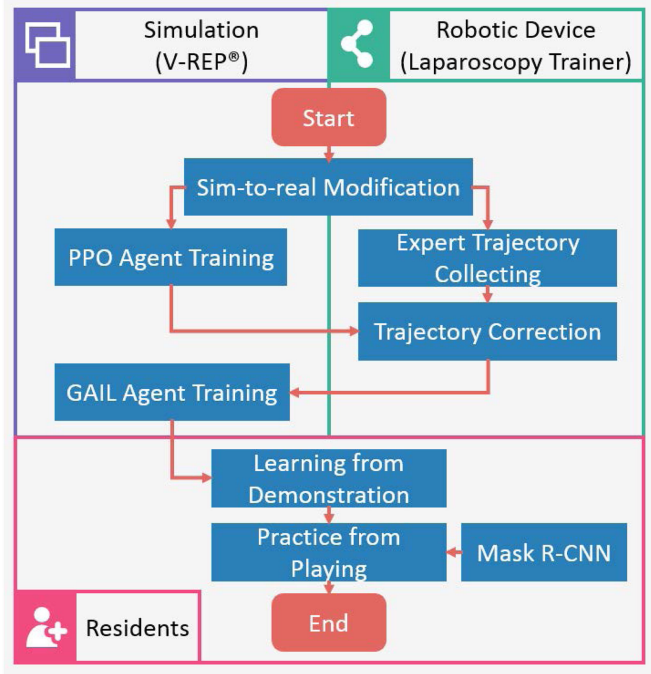


Fig. 1. The flow diagram of robot-assisted laparoscopy training system.

or generated policies, trainees can simply hold the handle and learn the patterns of manipulation from demonstration.

C. Simulation Setup and Sim-to-Real

The simulation of laparoscopic tool usage and further learning are accomplished on the Virtual Robot Experimentation Platform (V-REP). The right-hand assembly model is modeled and controlled by Python remote API in real-time synchronous mode in the simulation. Under this mode, the simulation waits for remote commands in each time step (50 ms) and executes one time step after receiving the signal. Compared with torques controllers, musculotendon units (MTU) controllers, and proportional-derivatives (PD) controllers, velocity controller generated by deep RL agents have been shown to always achieve compatible scores with PD controllers and outperform other methods [24], [26]. Hence, velocity controllers are applied to the four target joints due to its simplicity and similarity to its actual implementation.

Generally, it is difficult to transfer and implement deep RL agents which have been well-trained on simulated environments directly on real locomotion tasks due to the reality gap and disparate complexity of tasks. To overcome the reality gap, system identification is first performed by implementing accurate physical parameters retrieved from the actual robot in the simulation. The parameters used in simulated actuator models are fine-tuned to achieve identical performance with the robotic device from the sampled trajectories. To test the controller performance, a depth sensor is deployed, and Mask R-CNN is used to determine the coordinates of tools' tip via segmentation. The error could be calculated by replaying sampled trajectories and the simulation could be subsequently fine-tuned to minimize this error. One of

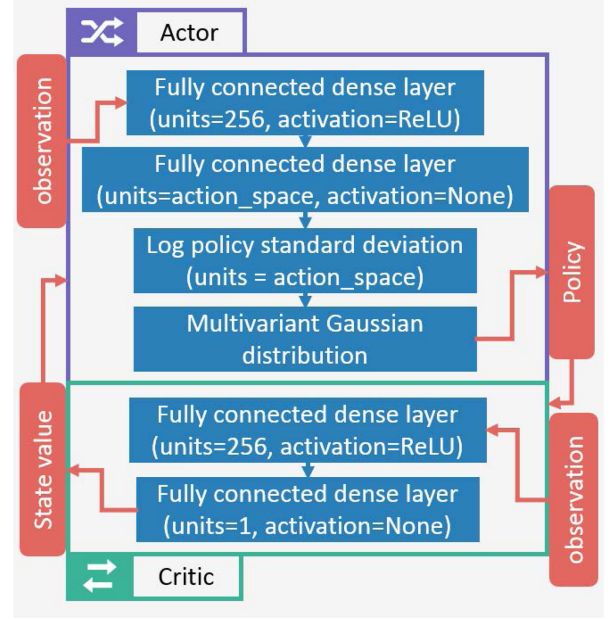


Fig. 2. The PPO agents architecture with actor-critic style. Value net and policy net are both constructed by two-layer perceptrons which have 256 units in the first layer with Rectifier Linear Unit (ReLU) activation function. The value net will output one state-value from one observation. The policy net will generate a multivariate Gaussian distribution over the action dimension.

the sample trajectories is demonstrated in Fig. 3 (b). The error is ± 2.3 mm. Next, perturbation (Gaussian noise) is introduced in the PPO training procedure to improve the robustness of the controller. Finally, actual physical trajectories from manipulating the robotic device are leveraged in training the GAIL agent, further enhancing the robustness of controller. The experiment design will be introduced in Section IV-A.

D. PPO Agents Training

After constructing the simulation environment, PPO agents [7] are trained to generate objective-oriented trajectory from the designed reward signals and leverage them in demonstration. For training in simulation, PPO agents follow standard RL setup constituted with Markov Decision Processes (MDPs) and interact with simulation environment E . At time step t , agents observe the state s_t and take action a_t through policy $\pi_\theta(a_t|s_t)$ which maps from the state to a probability distribution over actions by model parameters θ . After taking an action a_t , the agent will reach the next state s_{t+1} following transition dynamics $p(s_{t+1}|s_t, a_t)$ and receive reward r from reward function $r(s_t, a_t)$. The return values from the state are defined in a learned state-value function $V_{\theta'}(s)$ with model parameters θ' . The PPO agent is trained in an actor-critic style with separate value model and policy model. The whole structure is shown in Fig. 2.

The PPO agent is trained to maximize the loss function in each iteration:

$$L_t^{CLIP+VF}(\theta, \theta') = \mathbb{E}_t[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta') + c_2 E(\pi_\theta|s_t)] \quad (1)$$

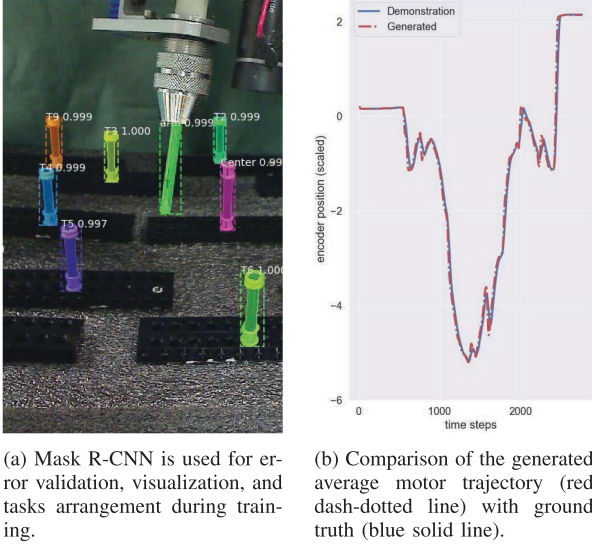


Fig. 3. Sample results of Mask R-CNN and trajectory comparison.

where $E(\pi_\theta|s_t)$ is an entropy bonus to ensure that the agent can fully explore the simulation environment, and c_1, c_2 are both coefficients to adjust weights of different loss. L_t^{VF} is a squared-error loss to update the value net with discount factor γ and target value function:

$$L_t^{VF} = (V_{\theta'}(s_t) - (r(s_t, a_t) + \gamma V_{\theta_{target}}(s_{t+1})))^2. \quad (2)$$

L_t^{CLIP} is a clipped loss which is used to simplify and replace the surrogate loss used in Trust Region Policy Optimization (TRPO) [22]:

$$L_t^{CLIP} = \hat{\mathbb{E}}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t)] \quad (3)$$

where $r_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_{old}}(a_t|s_t)$ is the probability ratio between current policy and old policy. ϵ indicates the clipped ratio which is used to limit the volume of update and stabilize learning procedure. This modification is a trade-off between the precision and calculation efficiency. High stability is achieved with the introduction of inductive bias. The Guided Policy Search [20] also utilizes similar KL-divergence loss with TRPO [22] which could be potentially simplified by the clipped loss.

\hat{A}_t is an advantage estimator for T time step trajectory:

$$\hat{A}_t = \delta_t + \gamma\delta_{t+1} + \dots + \gamma^{T-t+1}\delta_{T-1} \quad (4)$$

with $\delta_t = r_t + V(S_{t+1}) - V(S_t)$. The PPO agents can achieve various optimal policies under different criterions predefined by divergent reward functions. Compared with model-free off-policy RL algorithms which generate deterministic continuous policy, such as Deep Deterministic Policy Gradient (DDPG) [27], PPO can achieve high-convergent stable learning with a limited amount of data. To demonstrate the difference in learning, DDPG is tested in Section IV-B under the identical setting with PPO agents. The RRT is also tested in Section IV-B to compare RL algorithms with traditional motion planning method using predefined reward function.

In our experiments, the reward function is defined as follow: $r_t = \lambda - c_3\Delta_t + c_4e^{-\Delta_t}$ where Δ_t indicates the distance between tip and targets at time step t , λ is a constant, c_3 and

c_4 are coefficients used to weight the two reward terms respectively. Compared with the commonly used sparse reward signals [25], our reward signal is a continuous function with exponential bonus to encourage a rapid reaching and stable learning. This reward function also has the advantage in implementing model-free RL methods during simulation because the distance Δ_t could be easily acquired without kinematic models. The PPO agents are programmed on Tensorflow [28] platform with Python.

E. Trajectory Collection and Correction

The expert trajectory consists of various joint positions and is stored locally using the FPGA resources. The information is transferred to the remote workstation at the end of every session or as required for storage. Expert trajectories to be replayed can be pushed to the FPGA on demand.

Due to the limitation of reward function design, the trajectory generated by PPO agents cannot satisfy the demonstration requirement. In our experiment (Section IV-A), the designed reward function cannot accurately represent some of the expert patterns, such as the constrain behavior of human wrist. However, it is important for residents to learn the relaxed posture of wrist to ensure the accuracy and reduce fatigue during the operation. Therefore, a trajectory correction approach is implemented to replace the handle motion of PPO generated trajectory. In this approach, the robotic device will replay the PPO generated trajectory without handle motion. At the same time, the experts will only constrain the handle motion to correct the trajectory. The data collection on handle motion is similar to that of trajectory collection which is meant for all tool motions from human experts.

F. GAIL Agent Training, Residency Learning and Practicing

After the expert data collection and trajectory correction, GAIL agents are trained to extract policies directly based on the features and patterns from demonstrations. Compared with traditional inverse RL algorithms [9], [29], which recover the reward function from the features of data without calculating the optimal policy, GAIL agents are easier to train and output both reward signal and policy simultaneously. Furthermore, GAIL overcomes the distribution mismatch issue caused by multimodal behaviors of demonstrations. Such multimodal behaviors are more likely to occur in learning trajectory from medical experts because they may differ in performing the same task. Normal behavior cloning methods may introduce significant bias in this situation by constructing a direct mapping from observation to action. GAIL agents follow the same RL setup with PPO agents shown in Section III-D. These agents are constructed in a Generative Adversarial Networks (GAN) framework with discriminator D_w and policy generator $G_{w'}$. D_w is trained to perform classification and separate the policy generated by $G_{w'}$ from demonstrations. $G_{w'}$ is trained to generate policies based on the classification results from D_w . The agent architecture is shown in Fig. 4.

$G_{w'}$ is trained by PPO which leverages identical update rules mentioned in Section III-D. The discriminator D_w is trained by

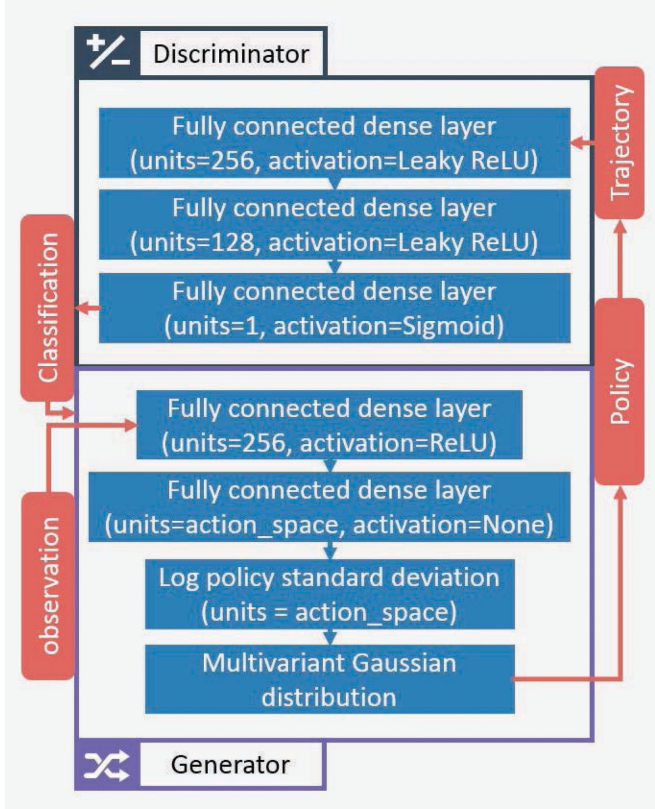


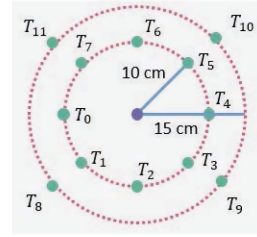
Fig. 4. Architecture of the GAIL agents. The discriminator net is constructed with fully connected three-layer perceptrons which have 256 units in the first layer with leaky ReLU activation function, 128 units in the second layer with leaky ReLU activation function and output classification probability with sigmoid activation function. Generator uses the same architecture as the PPO actor shown in Fig. 2.

minimizing the loss function:

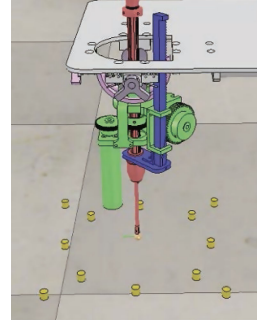
$$L^D = \hat{\mathbb{E}}_{\tau_i} [\log(D_w(s, a))] + \hat{\mathbb{E}}_{\tau_E} [\log(1 - D_w(s, a))] \quad (5)$$

where τ_i indicates the trajectory generated by G_w and τ_E represents the demonstration trajectory acquired from Section III-E. After the training, the trajectory generated by G_w could be utilized as a demonstration sample for surgical residents to learn from. D_w can provide distinct feedback on the trajectory captured during the surgical resident's practice session by directly validating the trajectory data.

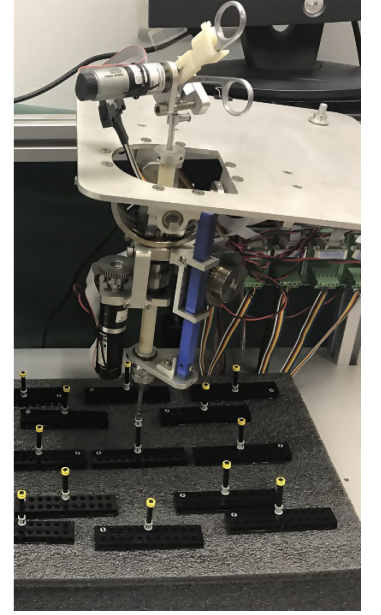
A Mask R-CNN is trained to track the surgical tools and calculate the total scores of the practice session. The designed experiment shown in Section IV-A contains multiple objectives and requires disparate models for individualized demonstration and evaluation. Therefore, during the practicing procedures, due to the noisy input from trainees, typical systems may not accurately recognize the objectives in which trainees intend to achieve purely from recorded data. Although strictly limiting the order in practicing different objectives can solve this issue, it is contrary to the goal of our system which is designed to provide sufficient freedom of variation in practicing and enhance skill sets in self-oriented practicing. Hence, the system could be programmed to consider both recorded data and distance between region masks to determine the model to be utilized.



(a) Tasks design.



(b) Simulation setup with test targets.



(c) Setup of right-hand robotic device with test targets.

Fig. 5. Tasks plan, simulation environment and device setup with test targets.

IV. EXPERIMENTS

A. Experiment Design and Tasks Setup

To fully test our RL agents performance and ensure sufficient objectives used in residency training, the designed experiment contains 12 individual right-hand motion training tasks. Eight of them were evenly distributed in a circle of radius 10 cm and centered 1cm vertically below the tip of laparoscopic tool. These targets ($A_{learn} = \{T_0, T_1, \dots, T_7\}$) in the training set were used for demonstration and learning. Other targets located in an outer circle of radius 15cm with the same center and height were mainly used for practicing and evaluation. These targets were represented in training set $A_{play} = \{T_8, T_9, T_{10}, T_{11}\}$. Based on this design (Fig. 5(a)), we setup identical testing environments in both simulation (Fig. 5(b)) and robotic device (Fig. 5(c)). In these tasks, users are required to manipulate surgical tools from one center point to the designated target in each task. The robotic devices are programmed to record the trajectory from both experts' and students' manipulation and manipulate surgical tools through GAIL agents for demonstration.

B. PPO Agents Training and Testing

After the setup of simulation environment, PPO agents were trained to accomplish each tasks 10 times with 1500 episodes with predefined reward signal mentioned in Section III-D. For each episode, agents ran in real-time simulation with $d_t = 50$ ms for each timestep and terminated at $t_{max} = 2$ s. Similarly, DDPG agents were also tested under identical settings in networks architectures and hyper-parameters. These agents were tested every 100 episodes to generate policies only with the mean of the output. The experimental results are shown in Fig. 6.

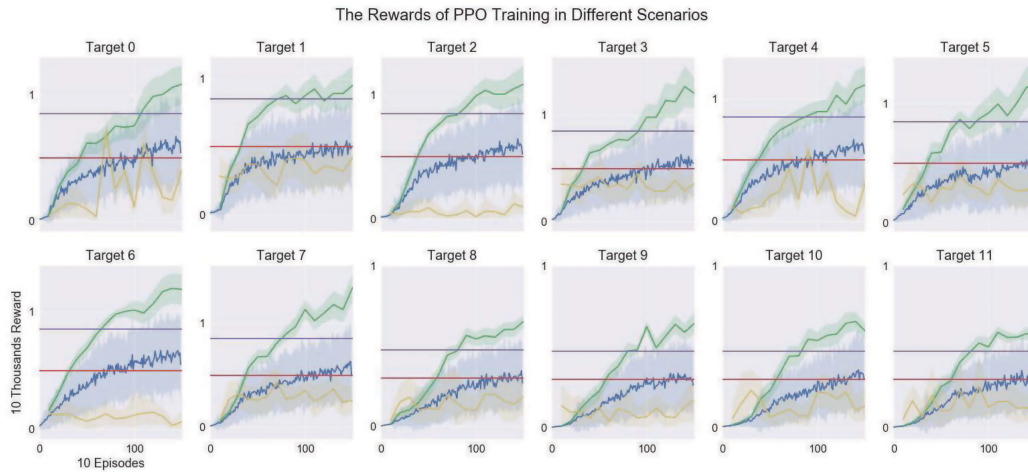


Fig. 6. Experimental results for training PPO agents at different scenarios. The green lines indicate the test results with standard deviation. The blue lines represent the training results on each episode with standard deviation. The red lines indicate the averaged baseline achieved by experts. The averaged RRT results are plotted as purple lines. The testing results of DDPG are plotted as yellow lines. The parameters used in training PPO agents: $\epsilon = 0.2$, $c_1 = 0.5$, $C_2 = 10^{-3}$, $\gamma = 0.99$, $\lambda = 10$, $c_3 = 0.5$, and $c_4 = 10^3$.

During the training of PPO agents, expert trajectories were collected from two medical collaborators. Each expert demonstrates each task five times. By replaying the expert trajectories, the average predefined rewards that the experts could achieve for A_{learn} and A_{play} were found to be 5197 and 3116 respectively. Hence, these results were considered as the baseline to validate the performance of agents. The RRT was also performed on tasks A_{learn} and A_{play} to compare the learning results with RL based methods. The RRT was implemented 10 times on each task under step distance $d_s = 2$ mm and maximum number of vertices $n_{maxv} = 2000$. The average predefined rewards that RRT could achieve for A_{learn} and A_{play} were 8505 and 4735 respectively. These results are all plotted in Fig. 6.

Based on the experimental results, PPO agents outperform both experts and RRT methods under the evaluation of specific reward function on each target. Hence, the trajectory generated by PPO agents could represent the objective-constrained behaviors for demonstration. Due to the limited quantity of data and relatively low number of update iteration, DDPG agents cannot achieve a stable learning.

C. GAIL Agents Training and Testing

After PPO agents training, the well-trained agents were used to generate five sets of trajectories for each target. Subsequently, as mentioned in Section III-E, the generated trajectories will be corrected by our collaborating clinicians and combined with original trajectories from experts as training data. This procedure is to uncover the reward function based on the features from trajectories and merge the objective functions. Partially utilizing demonstration will lead to recovering an incomplete reward function and contradict the purpose of learning the demonstration from both dynamics and experts.

The generators of GAIL agents utilized the identical simulation setting to that of the PPO agents training. The architecture of discriminator has been shown in Fig. 4. As the true objective function of demonstration is not known, it is not possible to

evaluate the performance of GAIL generator directly under this metric. However, a similar evaluation could be performed by the PPO reward function and the discriminator of well-trained GAIL agent. Since the true objective function is merged through mixed demonstration, PPO reward function can partially indicate the performance (an effective GAIL agent should achieve high scores). The results of well-trained discriminator in GAIL agents could be considered as a special reward function because it can successfully classify the trajectory which is similar to demonstration (high score means high similarity). The discriminator itself is also hard to evaluate because the input data distribution is consistently changing during the adversarial training. Hence, the inference results on generator trajectories were also recorded in Fig. 7 to validate the performance of both generator and discriminator.

We first evaluated the entire training procedure by predefined reward function and discriminator shown in Fig. 7. Next, the trajectories from human experts, PPO agents with correction, PPO agents without correction, and GAIL agents were evaluated by well-trained discriminators shown in Fig. 8. Behavior cloning agents using identical policy model architecture and hyper-parameters with GAIL agents were also trained to perform a direct mapping from observation and action by minimizing the mean squared error. Similarly, the results are evaluated by the predefined reward signal.

Based on the simulation results shown in Fig. 8, GAIL agents achieved compatible performance compared with expert trajectory and were able to successfully separate the trajectory without correction. It indicates the robustness of discriminator in classifying two similar trajectories with many same attribute values. It also shows that GAIL agents successfully imitate the demonstrations and can learn the features from expert behavior which are not easily represented in terms of dynamics. The scores of PPO correction and human experts are slightly higher than the scores of GAIL generators because the aforementioned trajectories are similar to the positive samples used in discriminator training. This difference also shows the capability and reliability

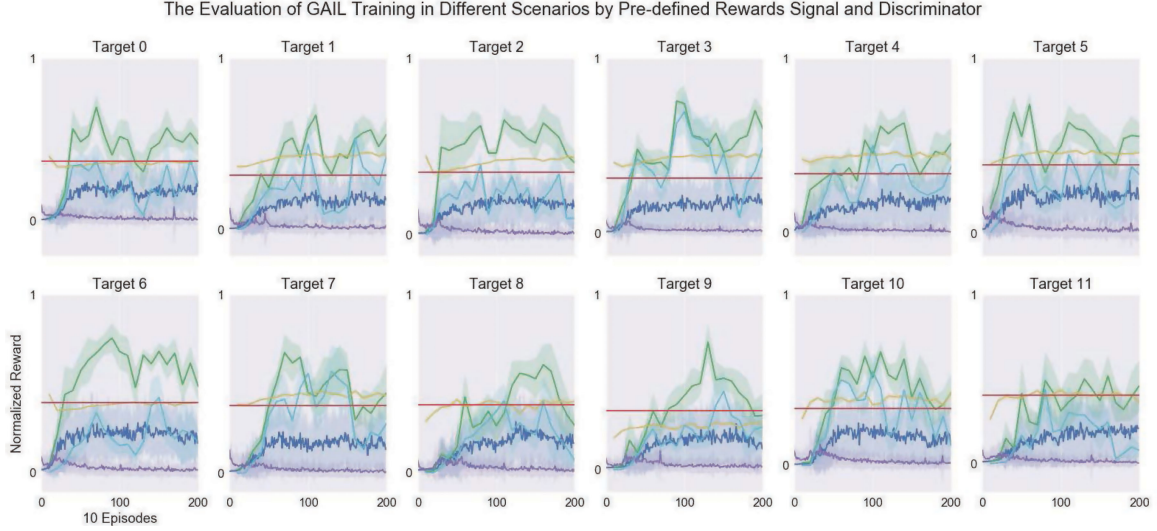


Fig. 7. Evaluation results for training GAIL agents at different scenarios validated by predefined rewards signal and discriminators. The green lines and dark blue lines represent the testing and training results with standard deviation evaluated by predefined reward signal respectively. The red lines indicate the average baseline. The test results and training results evaluated by discriminators are represented by light blue lines and purple lines respectively. The behavior cloning testing results evaluated by predefined reward function are plotted as yellow lines.

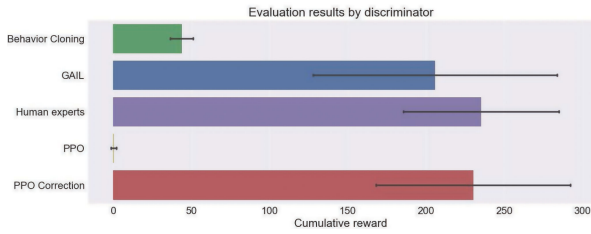


Fig. 8. Discriminator evaluation results for different types of trajectory with standard deviation. The data is acquired by averaging 5 scores on each scenario.

of discriminator in adversarial training. The behavior cloning results shown in Fig. 7 exhibit that it achieves similar results with demonstration, but cannot significantly boost the performance and may also lose precision in some cases (e.g., Target 9 in Fig. 7) because of the multimodal behavior of demonstration.

D. Learning from Demonstration and Practicing

The training procedures could be various due to specific course schedule of trainers. Our system is designed to best cooperate different schedules by pre-training a Mask R-CNN to segment surgical tools and targets. The samples generated by Mask R-CNN are depicted in Fig. 3(a). It could be used to enhance the visualization in learning without obstinate practicing. Based on mask regions and motor data, the system can also be programmed to calculate the final total score without restricting the sequence of different tasks.

Although our system can provide feedback on all tasks, we recommend trainees to learn the demonstration in tasks A_{learn} and practice in tasks A_{play} to generalize the learned skill on unseen targets. The scores for feedback could be calculated by the following equation:

$$Score = c_5 D_w(\tau_p) - c_6 Time \quad (6)$$

where c_5 and c_6 are weights, τ_p is testing trajectory, and $Time$ is the total time of completing one task. The coefficient could be adjusted by trainers to realize difference focus in training.

A preliminary experiment is conducted to compare our proposed system with traditional training method (i.e., box training). We recruited 50 students without prior experience in handling laparoscopic tools from *Department of Medicine, NUS* to participate in the training experiment. The subjects were equally and randomly separated into two groups with 25 students in each group. The control group performed the training on traditional box training solely based on their own practice for 15 minutes, while the study group learned the manipulation on our proposed system by utilizing demonstration for 10 minutes and practicing for 5 minutes. The performance of the two groups was validated before and after the training. Since the scoring method (6) cannot evaluate the training result of traditional methods, the following equation is used to evaluate the performance of trainees:

$$Score = c_7 - Time - c_8 * N_f \quad (7)$$

where c_7 is the standard flag of completing the tasks, $Time$ is the total time of completing tasks, N_f is the times of failing to complete one task amplified by coefficient c_8 . In our statistical analysis, we set $c_7 = 300$ and $c_8 = 100$ based on our experience. For each student, $Score_{pre}$ and $Score_{aft}$ will be calculated in the tests before and after training. Hence, the skill improvement for each student could be measured by $Score_{imp} = Score_{aft} - Score_{pre}$. We perform the t -test (alpha level $\alpha = 0.05$) by proposing a null hypothesis based on the mean of two populations: $H_0 : \mu_s - \mu_c = 0$ where μ_s denotes the mean of skill improvement for population using the proposed system and μ_c indicates the mean of skill improvement for population using the traditional method. The t -test result is shown in the Table I.

Based on the t -test result, we can reject the null hypothesis H_0 and consider that the proposed system statistically

TABLE I
STATISTICAL RESULT OF STUDENT'S *t*-TEST

Methods for <i>t</i> -test	Parameters of Student's <i>t</i> -test			
	Mean	Variance	<i>t</i> -value	<i>p</i> -value
Proposed System	178.48	86.97	3.313	$p < 0.005$
Traditional Method	101.96	75.97		

outperform the traditional method in laparoscopy training. However, more statistical analyses on different training procedures with complex tasks are recommended as future works. Each function could be validated individually and sequentially with specially designed tasks and evaluation metrics to fully investigate the effectiveness of our system for training and robot-assisted surgical training in general.

V. CONCLUSION

In this letter, we introduce a robot-assisted laparoscopy training system which utilizes deep RL algorithms (i.e., PPO and GAIL) to learn from both simulation and expert behaviors. By incorporating actual laparoscopic tools and operated by RL agents, trainees can learn from both demonstrations and practice with real tactile experience. These demonstrations combine the latent patterns from expert trajectories and objective-constrained trajectories generated by RL agents. The usage of Mask R-CNN in our system enhances the automation of training feedback, visualization, and error validation. Based on the results from simulation and practices on the robotic device, our system can successfully learn from simulation and expert data, generate optimal policies for demonstration, and evaluate the trajectory from trainees. The statistical analysis shows that the skill improvement by utilizing our training system is statistically significant.

For future works, we would like to include more training tasks (e.g., pick-and-place) in our system, invite trainees to fully investigate our system and conduct comprehensive statistical evaluation on each function of our system. We also like to investigate the application of other deep RL algorithms on our robotic system, for example, Hindsight Experience Replay [25] which has the capability for universal value function approximation.

ACKNOWLEDGMENT

C.-K. Chui would like to acknowledge the contribution of A/Prof. S. Chang of Mount Elizabeth Hospital, Singapore for his input on surgeries and medical education.

REFERENCES

- [1] G. G. Hamad and M. Curet, "Minimally invasive surgery," *Amer. J. Surgery*, vol. 199, no. 2, pp. 263–265, 2010.
- [2] K. Fuchs, "Minimally invasive surgery," *Endoscopy*, vol. 34, no. 02, pp. 154–159, 2002.
- [3] G. T. Sung and I. S. Gill, "Robotic laparoscopic surgery: A comparison of the da vinci and zeus systems," *Urology*, vol. 58, no. 6, pp. 893–898, 2001.
- [4] G. H. Ballantyne and F. Moll, "The da vinci telerobotic surgical system: The virtual operative field and telepresence surgery," *Surgical Clinics*, vol. 83, no. 6, pp. 1293–1304, 2003.
- [5] M. A. Lerner, M. Ayalew, W. J. Peine, and C. P. Sundaram, "Does training on a virtual reality robotic simulator improve performance on the da vinci surgical system?" *J. Endourology*, vol. 24, no. 3, pp. 467–472, 2010.
- [6] D. Kaushik, R. High, C. J. Clark, and C. A. LaGrange, "Malfunction of the da vinci robotic system during robot-assisted laparoscopic prostatectomy: An international survey," *J. Endourology*, vol. 24, no. 4, pp. 571–575, 2010.
- [7] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, arXiv:1707.06347.
- [8] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Proc. Adv. Neural Inf. Process. Syst. Conf.*, 2016, pp. 4565–4573.
- [9] P. Abbeel and A. Y. Ng, "Inverse reinforcement learning," in *Proc. Encyclopedia Mach. Learn.*, Springer, 2011, pp. 554–558.
- [10] E. N. Spruit, G. P. Band, and J. F. Hamming, "Increasing efficiency of surgical training: Effects of spacing practice on skill acquisition and retention in laparoscopy training," *Surgical Endoscopy*, vol. 29, no. 8, pp. 2235–2243, 2015.
- [11] B. Varadarajan, C. Reiley, H. Lin, S. Khudanpur, and G. Hager, "Data-derived models for segmentation with application to surgical assessment and training," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, Springer, 2009, pp. 426–434.
- [12] N. Ahmadi *et al.*, "Automated objective surgical skill assessment in the operating room from unstructured tool motion in septoplasty," *Int. J. Comput. Assisted Radiol. Surgery*, vol. 10, no. 6, pp. 981–991, 2015.
- [13] M. J. Fard, S. Ameri, R. B. Chinnam, A. K. Pandya, M. D. Klein, and R. D. Ellis, "Machine learning approach for skill evaluation in robotic-assisted surgery," in *Proc. World Congr. Eng. Comput. Sci.*, vol. 1, 2016.
- [14] D. W. Stovall, A. S. Fernandez, and S. A. Cohen, "Laparoscopy training in united states obstetric and gynecology residency programs," *J. Soc. Laparoendoscopic Surgeons*, vol. 10, no. 1, pp. 11–15, 2006.
- [15] R. Fox, S. Krishnan, I. Stoica, and K. Goldberg, "Multi-level discovery of deep options," 2017, arXiv:1703.08294.
- [16] S. Krishnan, R. Fox, I. Stoica, and K. Goldberg, "DDCO: Discovery of deep continuous options for robot learning from demonstrations," in *Proc. Conf. Robot Learn.*, 2017, pp. 418–437.
- [17] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.
- [18] S. M. LaValle and J. J. Kuffner Jr., "Rapidly-exploring random trees: progress and prospects," in *Algorithmic and Computational Robotics: New Directions*, 2000.
- [19] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.
- [20] S. Levine and V. Koltun, "Guided policy search," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1–9.
- [21] D. Silver *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.
- [22] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1889–1897.
- [23] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 49–58.
- [24] J. Tan *et al.*, "Sim-to-real: Learning agile locomotion for quadruped robots," 2018, arXiv:1804.10332.
- [25] M. Andrychowicz *et al.*, "Hindsight experience replay," in *Proc. Adv. Neural Inf. Process. Syst. Conf.*, 2017, pp. 5048–5058.
- [26] X. B. Peng and M. van de Panne, "Learning locomotion skills using deeprl: Does the choice of action space matter?" in *Proc. ACM SIGGRAPH/Eurographics Symp. Comput. Animation*, ACM, 2017, Art. no. 12.
- [27] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," 2015, arXiv:1509.02971.
- [28] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Conf. Operating Syst. Des. Implementation*, 2016, vol. 16, pp. 265–283.
- [29] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *23rd AAAI Conf. Artif. Intell.*, Chicago, IL, USA, 2008, vol. 8, pp. 1433–1438.