

Autonomous Blood Suction for Robot-Assisted Surgery: A Sim-to-Real Reinforcement Learning Approach

Yafei Ou¹, *Student Member, IEEE*, Abed Soleymani¹, Xingyu Li¹, *Member, IEEE*,
and Mahdi Tavakoli^{1,2}, *Senior Member, IEEE*

Abstract—Recent applications of deep reinforcement learning (DRL) in surgical autonomy have shown promising results in automating various surgical sub-tasks. While most of these studies consider the rigid and soft body dynamics in the surgery such as tissue deformation, only a few have investigated the situation where fluid is present. However, the presence of fluids, particularly blood, is common in surgeries, and interacting with them adds additional challenges to task automation. In this work, we investigate the use of DRL in automating blood suction, a common surgical sub-task where blood is removed from the surgical field. We build a blood suction simulation environment based on position-based fluids (PBF), train an agent with domain-randomized environment parameters through curriculum learning, and obtain a generalizable policy that can be applied to various shapes of tissue and types of liquid. Real-world experiments show that the agent can perform autonomous suction in different tissue models with different amounts and types of liquid, and only one of the 50 trials resulted in more than 3 ml of blood remaining.

Index Terms—Medical robots and systems, reinforcement learning, surgical robotics, laparoscopy.

I. INTRODUCTION

A large number of recent research in surgical robotics focuses on the automation of surgical sub-tasks, such as needle manipulation [1], [2], [3], suturing [4], [5], cutting [6], [7], [8], vessel manipulation [9], tissue retraction and deformation [10], [11], [12], [13]. However, most of these studies focus on the manipulation of rigid and soft objects, although fluid-related tasks are also common in surgeries, due to the presence of body fluids, especially blood.

As bleeding is inevitable during surgery, blood is often present in surgical scenes. In practice, surgeons usually clear

Manuscript received: March 5, 2024; Revised: March 5, 2024; Accepted: June 24, 2024.

This paper was recommended for publication by Editor Jessica Burgner-Kahrs upon evaluation of the Associate Editor and Reviewers' comments. 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), Alberta Innovates, the Alberta Jobs, Economy, and Trade Ministry's Major Initiatives Fund A-Medico, the China Scholarship Council (CSC), and Alberta Advanced Education. (Corresponding author: Yafei Ou.)

¹Yafei Ou, Abed Soleymani, Xingyu Li and Mahdi Tavakoli are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta, Canada (e-mail: {yafei.ou, zsoleymani, xingyu, mahdi.tavakoli}@ualberta.ca).

²Mahdi Tavakoli is with the Department of Biomedical Engineering, University of Alberta, Edmonton, Alberta, Canada.

Digital Object Identifier (DOI): see top of this page.

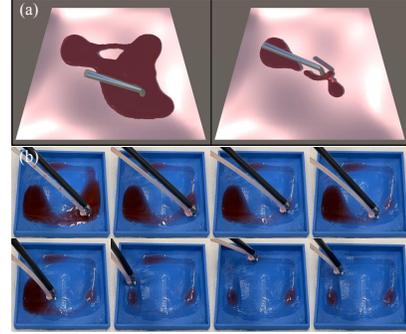


Fig. 1. Sim-to-real transfer of a blood suction agent. (a) Snapshots from the simulation environment and (b) Snapshot sequence taken from a real-world trial.

up the field at times by suctioning blood out of the field using a suction tool. Thus, it is an essential and common auxiliary task that involves a considerable amount of time and effort and leads to intermittent interruption of the primary surgical operation. Therefore, automating this process can significantly reduce the workload of surgeons.

Only a few studies have been conducted to automate blood detection and removal in surgeries [14], [15], [16]. In [14], a 3D-printed soft robot manipulator is used for blood suction. However, this work is focused on the design and control of the robot manipulator, instead of the optimality of the generated path for suction. In [15], optical flow is used to track the blood flow from the image, and a trajectory for suction is generated to maximize the ability to remove blood while moving upstream toward the bleeding location. However, this work uses a computational trajectory planner based on the initial detected blood region and the flow, and the trajectory is executed in an open-loop manner. The time spent on blood detection and trajectory generation results in a response time of a few seconds in trajectory planning, which is not ideal when blood needs to be cleared instantly.

Authors of [16] implemented a differentiable position-based fluids (PBF) model of the blood and used model predictive control (MPC) to generate a suction trajectory that minimizes a cost function determined by the distance between blood particles and a target height. However, the requirement for an accurate model of the bleeding surgical scene can limit the use of this approach, as the quality of the planned trajectory is affected by the accuracy of the model, making

identifying an appropriate set of fluid parameters such as cohesion necessary, which is typically challenging when using PBF. Furthermore, the need for a differentiable simulator, as well as the optimization process for solving MPC by forward-stepping the simulation at each step, potentially limits the scale of the problem that this method can address. For instance, if tissue deformation and tool-tissue contact are further considered, the forward-stepping of the simulation will be more computationally expensive at each step.

Recent advances have shown success in applying deep reinforcement learning (deep RL, DRL) to automating surgical sub-tasks. RL is known for better adaptability and generalizability, as it can continuously learn and adapt to new scenarios without requiring a predefined model, potentially handling unexpected variations in surgical environments more robustly. Furthermore, it enables learning a model-free policy that directly takes the raw sensory data as input, such as the images, thereby reducing or eliminating the need for manual feature extraction and is generally faster than model-based approaches such as MPC during execution because of the fast inference of neural networks.

In this work, we build a fast blood suction simulation environment for RL using GPU-based PBF by leveraging Nvidia PhysX¹. We train an agent that utilizes a binary image mask of the blood region and tissue shape as input to automate blood suction. The agent is first trained in the simulation environment, and then transferred to the real-world setup, as shown in Fig. 1.

The main contributions of this work are as follows:

- 1) We build a blood suction simulation environment for RL training using PBF based on Nvidia Physx 5, Unity, and ML-Agents that enables randomized tissue shapes and liquid parameters for curriculum learning;
- 2) While the task difficulty increases as the learning curriculum advances, a blood suction agent that observes the tissue shape and the binary image mask of the blood is trained to remove blood from various shapes of tissue with different liquid properties;
- 3) We show through real-world experiments that the trained agent can perform autonomous blood suction in different tissue models under a direct sim-to-real transfer, and investigate the unique behavior patterns of the agent.

Although the problem considered in this work is simplified, the results have shown promise in scaling up the problem to more complex situations, such as training RGB image-based end-to-end policies and considering more realistic surgical simulations including tissue deformation.

II. RELATED WORK

A. Surgical Autonomy using DRL

DRL has been applied to a number of surgical sub-tasks, including needle manipulation [2], [3], suturing [5], tissue manipulation [11], [12], [13] and cutting [6], [7], [8]. While DRL performs well in many rigid manipulation tasks and some soft object manipulation tasks, tasks involving more

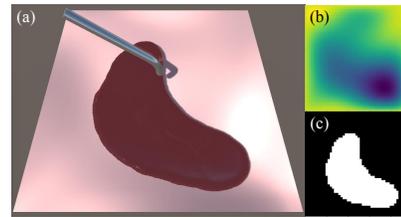


Fig. 2. An example of the simulation environment. (a) rendered scene; (b) the depth of the tissue at each location; (c) synthetic binary image mask representing the blood region.

complex dynamics, such as soft tissue cutting and blood suction, continue to pose significant challenges due to the difficulty in simulating the manipulation and training agents in these environments.

B. Surgical Robotics Simulation

High-speed and realistic simulation environments are essential for training DRL agents and for achieving sim-to-real transfer. A number of surgical simulation environments for this purpose have been proposed in recent years, including dVRL [17], AMBF-RL [18], UnityFlexML [19], SurRoL [20], LapGym [21], and Surgical Gym [22]. Most of these simulators do not provide straightforward integration of fluid manipulation. While UnityFlexML is based on Nvidia Flex, which supports PBF, the discontinuation of Flex makes it inadvisable to develop new environments based on it.

C. Fluid Simulation and Manipulation

This work is also related to the recent research on robotic manipulation involving fluids and the development of fluid simulation environments for robot learning. A few examples of previous work are [23], [24], [25], [26]. To simulate fluids, these studies have utilized Navier–Stokes equations, material point method (MPM), and PBF. PBF is generally more efficient and is often used when learning is involved. Most of these studies focus on daily tasks such as pouring and fluid mixing, and none of them considers liquid suction.

III. METHODOLOGY

A. Problem Formulation

Without loss of generality, we consider blood suction within a square-shaped footprint of tissue and assume that the blood is detected and segmented separately from regular RGB images to obtain an image mask. At each action step, the current observations including the blood region mask are passed to the policy to obtain the next step movement of the suction tool. This differs from [15], where optimal suction tool trajectories are generated beforehand based on initial knowledge about the blood flow, and then executed in an open-loop manner.

B. Simulation Environment Overview

To build a blood-suction learning environment, it is important to simulate fluid flow efficiently, as fast simulation can result in faster training in terms of the wall-clock time. PBF is

¹<https://developer.nvidia.com/physx-sdk>

one of the most commonly used approaches for efficient fluid simulation [27]. Compared with the other approaches such as smoothed particle hydrodynamics (SPH) and material points method (MPM), PBF offers a balance between computational costs, simulation stability, and visual plausibility, and has become increasingly popular in recent years for both game development and real-time simulation.

We build our simulation environment based on Nvidia PhysX 5 SDK. PhysX is a real-time physics engine that allows various physical simulations with GPU optimization. Specifically, we utilize its capability of high-performance PBF simulation for simulating blood suction in this work. With PhysX 5 as the low-level physics engine, we build our simulation environment in Unity, a 3D game development software that has shown potential in both medical simulations and RL environment development. It is worth noting that the current built-in 3D physics engine of Unity is PhysX 4.1, which does not support PBF. Particularly, the Machine Learning Agents Toolkit (ML-Agents) for Unity [28] is utilized for building the RL environment.

The simulation environment consists of a square-shaped footprint of tissue with random depth at different locations, a fixed amount of blood, and a suction tool, as shown in Fig. 2. The binary image mask is from the top view and is synthesized through the orthographic projection of each blood particle.

C. Suction Model

In PBF [27], the fluid is represented by a large number of particles with positions \mathbf{x}_i and velocities \mathbf{v}_i . During each simulation step, the velocities of the particles are first predicted based on the external forces $\mathbf{f}_{ext}(\mathbf{x}_i)$, such as gravity:

$$\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta\tau \mathbf{f}_{ext}(\mathbf{x}_i). \quad (1)$$

Here, $\Delta\tau$ is the simulation time step. The next-step positions can thus be predicted by $\mathbf{p}_i = \mathbf{x}_i + \Delta\tau \mathbf{v}_i$. \mathbf{p}_i is then iteratively corrected by solving positional constraints, such as collision [27]. Finally, the velocities are updated based on the predicted position change of the particles, and the positions are set to the predicted ones using $\mathbf{v}_i \leftarrow (\mathbf{p}_i - \mathbf{x}_i)/\Delta\tau$ and $\mathbf{x}_i \leftarrow \mathbf{p}_i$.

To simulate suction, we apply an additional external force to modify the velocities of each particle at the beginning of each simulation step in (1). Given the suction force scale F_s , and the mass of the i -th particle m_i , suction is simulated by a spherical cone-shaped force field applied at the end of the suction tool,

$$\mathbf{f}_{suction} = \frac{F_s}{m_i} \cdot \frac{\mathbf{x}_{o,i}}{\|\mathbf{x}_{o,i}\|} \cdot \left(\frac{1}{\|\mathbf{x}_{o,i}\|} - \frac{1}{r_s} \right) \quad (2)$$

if

$$\arccos\left(\frac{\mathbf{x}_{o,i} \cdot \mathbf{F}_c}{\|\mathbf{x}_{o,i}\|}\right) \leq \theta_s, \text{ and } \|\mathbf{x}_{o,i}\| < r_s.$$

where $\mathbf{x}_{o,i}$ is the position vector pointing from the force center to the i -th particle, \mathbf{F}_c is a unit vector representing the central axis of the spherical cone, θ_s is the cone angle, and r_s is the radius of the force field. Increasing distance from the force center results in a gradual decrease in force to zero. Since all particles will be looped over at each simulation

step, a strict cut-off value of r_s is used to allow early-stopping of the computation if a particle is outside the force's effective range to increase the computation efficiency, unlike in [16] where a 2D Gaussian function is applied across all particles as suction displacement. Time-consuming operations such as exponentiation are also avoided. A faster simulation allows a higher time scale to speed up the simulation during training, thus reducing the clock time needed, as discussed in Section III-G. During simulation, the particles are removed after reaching a certain height threshold above the tissue.

The orientation of the suction tool is fixed such that \mathbf{F}_c is always pointing downwards. As we assume that the shape of the tissue is known to the agent, the height of the suction tool is manually kept at a fixed distance from the tissue at each location. Therefore, the action of the agent is simply the 2D movement of the suction tool within the tissue area.

D. Tissue Shape Generation

Introducing randomness in the shape of tissues plays a pivotal role in enhancing the robustness and generalization capabilities of the agent, enabling it to handle a broad spectrum of scenarios. Such randomness simulates a wide range of near-to-real challenges for the RL agent and mirrors the unpredictability and diversity present in actual surgeries. While other methods, such as Perlin noise or fractal-based approaches, can introduce randomness for a given surface, Bezier surfaces [29] offer distinct advantages due to their simplicity and well-defined mathematical formulation. This enables us to adjust the level of smoothness and control the surface's shape while maintaining a balance between the randomness and realism of the tissue's structure. A Bezier surface \mathcal{S} can be defined as

$$\mathcal{S}(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{i,j} \cdot B_{n,i}(u) \cdot B_{m,j}(v), \quad 0 \leq u, v \leq 1 \quad (3)$$

where $\mathcal{S}(u, v)$ represents the Bezier surface, $P_{i,j}$ are the control points, and $B_{n,i}(u)$ and $B_{m,j}(v)$ are the Bernstein basis (or blending) functions for the parameters u and v defined as

$$B_{n,i}(u) = \frac{n!}{i! \cdot (n-i)!} \cdot u^i \cdot (1-u)^{n-i}.$$

Adjusting the positions of control points generates a variety of tissue shapes with controlled randomness.

E. Reward and Observation Design

As a simplification, similar to [16], we consider the main objective as removing as much blood as possible. The reward function consists of a reward for the amount of blood removed during each step, an extra terminal reward for removing all blood, and an action penalty for tool movements. The number of particles being removed during each step is used to determine the amount of blood being suctioned out:

$$r(s_t, a_t, s_{t+1}) = N_p^t - N_p^{t+1} + C_1 \delta(N_p^{t+1}) - C_2 \|a_t\|, \quad (4)$$

where N_p^t is the number of active particles at time step t , $\delta(N_p^{t+1})$ indicates whether there is active particles remaining,

TABLE I
VALUES/RANGES OF ENVIRONMENTAL PARAMETERS IN EACH LESSON

Parameter	Lesson 0	Lesson 1	Lesson 2
Control points grid size	[6, 8]	[4, 10]	-
Penalty scale C_2	0	0.01	0.02
Friction coefficient	[0.05, 0.1]	[0.02, 0.2]	-
Viscosity parameter	[4, 8]	[4, 15]	[1, 15]
Surface tension	0.05	[0.05, 0.08]	[0.05, 0.1]
Cohesion	25	[10, 30]	[5, 50]

and $\|a_t\|$ is the norm of the actions. C_1 and C_2 are two weighting factors.

Due to the complex dynamics of the environment introduced by different shapes of the tissue and the fluid, designing an appropriate observation space is essential to ensure efficient learning of the task. In order to provide more information regarding the dynamics of the environment, the observation includes the tissue depth map, the binary image mask of the blood, and the suction tool location. The binary image mask is stacked with the ones from 3 previous steps to capture the temporal feature, and the tool location is also stacked with the ones from 4 previous steps.

F. Randomization and Curriculum Settings

As discussed in Section III-D, the shape of the tissue is randomized during each episode to enhance the robustness of the trained agent to various tissue shapes during the surgery. To further facilitate a more generalizable policy that can deal with the various physical properties of the blood, such as cohesion and viscosity, the fluid-related parameters are also randomized throughout the training. Furthermore, the initial position of the suction tool and the position of the blood from where it drops and flows in the tissue are randomized at each episode as well. This procedure is known as domain randomization (DR) [30].

While randomization of the environment parameters increases the generalizability of the trained policy, large variations in the environment can make the learning slow or limit the performance of the trained model. Furthermore, the action penalty regarding the motion of the tool also impedes the training of the agent as it limits the exploration by favoring smaller movements. To overcome such issues, we utilize curriculum learning [31] by starting from simpler environments with fewer fluid variations and without action penalty, and eventually approaching the more complex situations. The randomized parameters and their values or ranges during each lesson are listed in Table I. The parameters that have a sampling range are sampled uniformly during training. We show in Section IV-A that this approach outperforms training without a curriculum.

G. Training Settings

The ML-Agents Toolkit is utilized for training. The physics simulation (environment) step size is 0.02 seconds. However, the decision (action) frequency is lower than the simulation steps and is set to be once per 0.2 seconds. At the beginning of each episode, 3,000 fluid particles are dropped from a small distance above the tissue and flow in accordance with its shape. During each episode, the maximum allowed environment steps

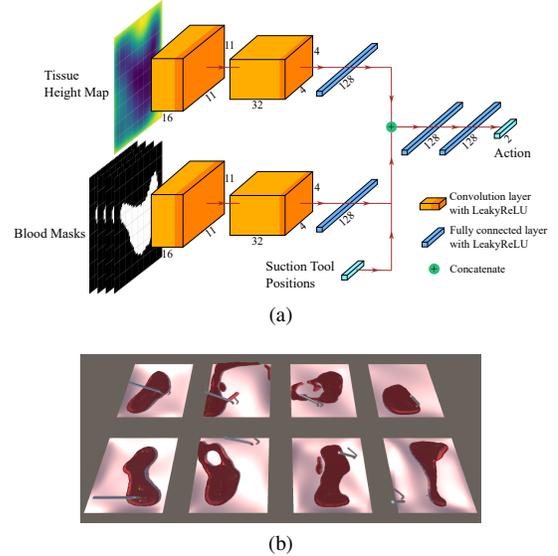


Fig. 3. Training settings. (a) Policy network architecture; (b) Training with parallel simulation environments.

is 1,000, while the first 100 steps are used to drop and stabilize the fluid without agent actions being taken. An episode terminates if the maximum number of steps is reached, or if all the fluid has been removed.

The size of the tissue shape depth map and the blood region mask are both 50x50 pixels as input to the policy and the value networks. Soft actor-critic (SAC) [32] is used and a total of 3 million environment (physics) steps are trained. The policy network architecture is shown in Fig 3a. The learning rate is 0.003, the batch size is 128, the replay buffer size is 50,000, the initial entropy coefficient is 1.0, and the discount factor γ is 0.95. To accelerate training, experiences are collected from 8 parallel simulation environments, as shown in Fig 3b. The simulation runs at a higher speed than the actual clock time at a time scale of 2. A higher time scale can cause unreliable simulation performances, as each PhysX step takes just under 0.01 seconds. Training and evaluations are run on PCs with Nvidia RTX 4070 and 3070 GPUs. Training takes 1.3 to 1.6 clock-time days.

IV. EXPERIMENTS AND RESULTS

A. Training Results

To evaluate the efficacy of using a learning curriculum, an agent is also trained without a curriculum directly using the final values or ranges of the parameters in the last lessons as a comparison. The learning curves of the agents trained with and without curriculum are shown in Fig. 4a. Furthermore, both of the trained models are evaluated for 1,000 episodes, and the frequency distribution regarding the cumulative reward of all evaluated episodes is shown in Fig. 4b. While the agent trained with a curriculum has only a slight advantage in terms of accumulative reward over the one without a curriculum, the frequency distribution clearly indicates that training with a curriculum results in higher generalizability to different environment variables, as more episodes tend to have a higher cumulative reward. Since the same seed is used when

evaluating both agents, the Wilcoxon signed-rank test can be used to examine the statistical difference between the two agents' performances. A one-tailed test is conducted and The p -value of the test is 0.0196, rejecting the null hypothesis at the 5% significance level, indicating that the two agents generate statistically different outcomes in terms of the cumulative reward in various episodes, and the performance of the agent using the curriculum is generally better. Indeed, when the curriculum is used, the agent can achieve a cumulative reward of more than 87.5 in 69.2% of the episodes, 5.7% higher than that achieved by the agent without using the curriculum. The standard deviation of the cumulative reward using the agent with a curriculum is 7.49, while for the agent without a curriculum, the value is 8.06. The lower standard deviation further suggests that the former one has a more consistent performance across all variations of the environment.

The performance of the agent is also compared with a human operator performing the task in the simulator through a joystick controller while watching the top view of the scene. Both the agent and the human operator perform blood suction for 50 trials in the same environments, and the blood remaining at each step is recorded, as shown in Fig. 4c. Among the 50 trials performed by the agent, there are three cases in which more than 5% of blood remains and one case in which more than 25% remains, leading to a high standard deviation at the end. Such edge cases may be difficult to avoid due to the sparser reward when less blood remains. Despite cases where the agent fails to remove all blood, the performance of the agent is generally close to that of a human.

B. Real-World Experiment Setup

In real-world experiments, the patient side manipulator (PSM) from the dVRK is utilized to grasp and hold a silicone tube acting as a suction tool. The tube is connected to a small pump for suctioning. Liquid with a red color is used to simulate the blood. 3D-printed box containers with a curved bottom surface are used as the tissue models where the fluid is poured into. The experiment setup is shown in Fig. 5.

Three tissue models (120 mm \times 120 mm) are generated using the same approach discussed in Section III-D and 3D-printed. They are labeled as Tissue 1, 2, and 3 with the control point grid sizes being 5, 8, and 10, respectively, with increasing shape complexity. The tissue models are registered in the robot Cartesian space.

Two different types of liquid are used during the experiments. One is a commercially available beverage with a red hue, the density of which is approximately 1.08 g/ml ("*regular*"). Another one is the same liquid but with a large amount of sugar added, resulting in a more viscous and heavy liquid whose density is 1.32 g/ml ("*viscous*"). For each tissue model, 4 sets of experiments are conducted with different amounts and types of liquid injected into the tissue model: (a) 10 ml of regular liquid; (b) 10 ml of viscous liquid; (c) 15 ml of regular liquid; (d) 15 ml of viscous liquid. One exception is that due to the simplicity of the tissue shape, the 15 ml configurations (c) and (d) are not tested on the tissue model with control point grid size 5 since there is no

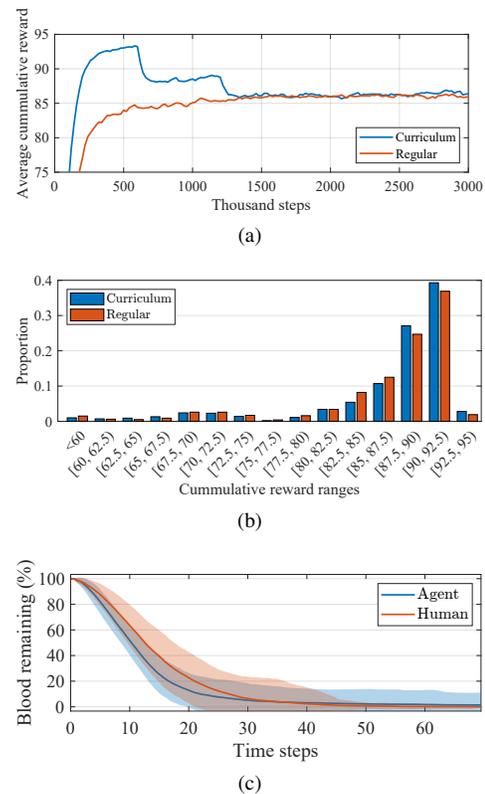


Fig. 4. Training results of agents with and without a learning curriculum. (a) Learning curves of agents trained with and without a curriculum. Exponentially smoothed with a window size of 4. (b) Distribution regarding the cumulative reward of 1,000 evaluation episodes. (c) Suction curves averaged across 50 trials performed by the agent and human operation. The shaded areas represent the standard deviations.

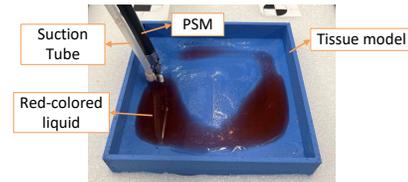


Fig. 5. Experiment setup.

significant difference in terms of the resultant blood regions and the suction tool trajectories, as liquid tends to flow to the only small valley of the tissue model. For each set of experiments, 5 trials are conducted. The liquid is poured from different positions in each trial, and the initial position of the suction tool is also randomized.

During the experiments, two side-by-side webcams (Logitech International S.A., Lausanne, Switzerland) are positioned above the setup and the tissue model position in the image is specified manually. The binary mask of the blood region is detected by manually setting thresholds in the hue, saturation, and value (HSV) color space. Detection results from both cameras are merged through naive feature-matching and projective transformation to resolve the problem of the PSM tool obscuring the field of view. The shape of the tissue is assumed to be known, and the suction tool is always kept at a certain distance above the tissue surface, the same as in

the simulation. The action frequency of robot movement varies from 1 to 5 Hz, depending on the travel distance of the suction tool during each step, which is generally lower than the one in the simulation. A precise scale under the whole setup is used to measure the weight change of the liquid inside the tissue model during the suction process. However, as the suction tool occasionally touches the tissue model due to inaccurate registration, the measurement of the weight can be noisy, and the valid data samples are selected manually.

C. Performance in the Real World

For the sake of concision, the sets of experiments are labeled using the format “T{1, 2, 3}_{R, V}{10, 15}”, where the number following “T” stands for the ID of the tissue model, “R” or “V” specifies whether the liquid is regular (R) or viscous (V), and the following number indicates the volume of the liquid used for each trial (10 ml or 15 ml). To denote individual trials within a set of experiments, an additional number is appended at the end. For instance, T2_V10 refers to experiments with the second tissue model using 10 ml of viscous liquid, and T2_V10_2 indicates the second trial in this set of experiments.

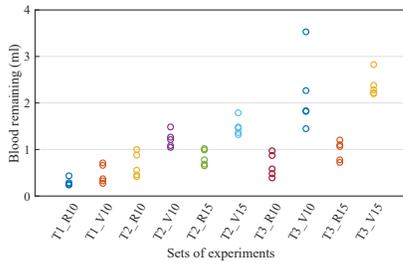


Fig. 6. Blood remaining after each trial in the 10 sets of experiments.

We group the 10 sets of experiments under different tissue and liquid configurations and present the final remaining blood volume measured using the precise scale in Fig. 6. In general, most trials using the regular liquid ended up with less than 1 ml of liquid remaining. In the trials using the viscous liquid, the amount of blood remaining varied between 0.3 and 2.8 ml, with an outlier of 3.5 ml. For the same tissue model and the same amount of liquid injected, there tends to be much more viscous liquid remaining compared with the regular one. This is due to the higher difficulty of suctioning viscous fluid, as more precise positioning of the tools is needed to create a larger contact area between the tube and the liquid and generate enough force to suction the viscous liquid. In general, more than 90% of the regular liquid and more than 80% of the viscous liquid can be removed, except for two trials. Given the small amount of liquid initially injected, the performance can be considered satisfactory.

As the complexity of the tissue model increases, the amount of blood that remains tends to increase. In experiments involving viscous liquid, this trend is more prominent. The increase in the volume of remaining liquid is largely related to the increased number of valley regions of the tissue, as the small amount of remaining liquid in the valleys can be undetectable by the blood detection algorithm, or inaccessible due to the

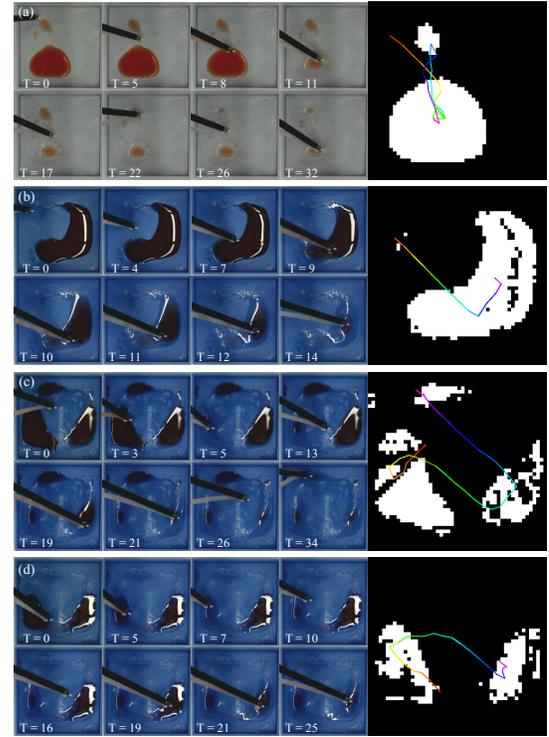


Fig. 7. Snapshots of example suction trials taken from the camera and the initial blood region masks augmented with the suction tool trajectories. Trajectories are marked using gradient colors starting from red and ending with magenta. (a) is taken from trial T1_R10_1, (b) from T2_R15_3, (c) from T3_R15_5, and (d) from T3_V10_1.

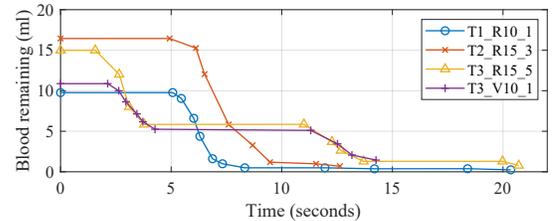


Fig. 8. Suction curves of example trials shown in Fig. 7. Due to inaccuracies in measurement and operation, the initial amount of liquid injected can slightly deviate from 10 or 15 ml.

small distance between the suction tube and the tissue. To record the amount of blood at each step and obtain suction curves as shown in Fig. 8, this small distance is necessary since it is not possible to measure the weight accurately if the suction tube is in contact with the tissue model on the scale. The more prominent trend regarding viscous liquid may be attributed to the difficulties associated with suctioning viscous fluids discussed previously, as we do not find a significant difference between using regular liquid and viscous liquid in terms of the quality of the suction tool motion based on the recorded trajectories of the tool.

Out of the 50 trials conducted, only one trial (“T3_V10_4”) ended up with a significant amount of liquid remaining (around 3.5 ml) due to the suction tool staying in one area. Our analysis of the binary mask detected at each step revealed an incorrectly detected area of blood in the region where the suction tool stays. As the incorrect detection persists, the suction tool is

stuck in the area and does not move to another location.

Fig. 7 shows snapshots taken from 4 trials, and the initial binary mask of the blood region augmented with the suction tool trajectory. Fig. 8 displays the amount of blood remaining in the tissues over time in the same 4 trials.

Two prominent behavior patterns of the suction policy in the experiments are further discussed in Section V-A.

V. DISCUSSION

A. Behavior Patterns in the Experiments

Tendency to visit local valley regions of the tissue. One pattern of the suction tool motion is that it tends to directly visit the local valleys of the tissue model. For instance, in Fig. 7d, the initial trajectory of the tool is towards the valley of the tissue model on the left, rather than the nearest blood pixel. This behavior is learned through training as going directly to the valley area where blood accumulates should generally result in a higher immediate reward and less penalty for actions.

Favoring large blood areas. Another typical pattern of the agent’s behavior is that it tends to go to larger areas of blood first, even if there is a smaller area closer to the current tool position, one example of which is shown in Fig. 7a. This is related to the reward function and the discount factor as the agent will be more likely to receive a higher immediate reward if it goes to a large area first and removes a large amount of blood. Whether this behavior is desired or not depends on the judging criteria, as a surgeon might choose to either remove a large amount of blood first or start with a smaller and closer one depending on the situation. This behavior pattern makes the agent more robust to noisy detection results of the blood region. However, in some cases, it can result in the tool moving between blood regions back and forth.

B. Choice of Observation

The learning curves of agents with different observation spaces trained for the first lesson are presented in Fig. 9. It is shown that in the absence of the tissue shape or stacking of the recent binary masks and the positions of the suction tool, training performance deteriorates. This is generally caused by the fact that less information is available to the agent, such as when the tissue shape is not provided, the valley regions of the tissue are unknown to the agent, and therefore the amount of blood at each location and the possible blood flow in the next few steps cannot be inferred. The number for the stacked observations of the blood masks and suction tool positions was determined experimentally through trial and error, while also considering the increase in memory usage.

C. Limitations and Future Work

Several simplifications are made in this work, one of which is the lack of continuous addition of blood during suction, although in reality there can be a bleeding point that continues to add blood to the field. However, simulating this situation significantly adds to the complexity of the simulation environment, as the bleeding speed, initial position and direction of

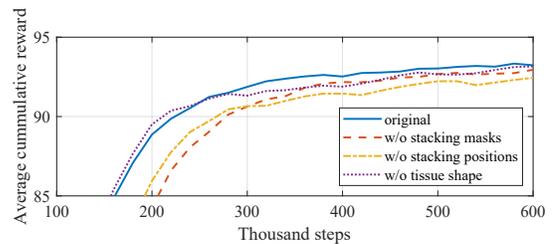


Fig. 9. Learning curves of agents with other configurations of observation in the first lesson (exponentially smoothed).

the blood can have large variations. We will simulate active bleeding and develop an agent that can handle this situation to increase the applicability of our agent in real surgeries in future work. Additionally, we have assumed that the deformation of the tissue is insignificant during suction, and a rigid tissue is used. In practice, the tissue shape topology can be obtained through a depth camera, or 3D reconstruction using stereo computer vision algorithms by utilizing the stereo endoscopic camera before bleeding happens. However, in reality, the tissue shape can change due to deformation caused by tool-tissue contact. Furthermore, there can be cases where the tissue shape cannot be represented by a Bezier curve, such as when a cut happens. Further investigation is needed to assess whether the agent trained on tissue represented by a Bezier curve can be applied to more realistic and complex shapes. Future work will include the simulation of more realistic tissue shapes and tissue deformation, and the agent will be adjusted accordingly to observe only the shape of the visible area of the tissue and output the 3D motion of the suction tool.

Although visually plausible by tuning the force parameters, the suction is approximated by a cone-shaped force field for simplicity and computational efficiency, while in reality suction is largely caused by the pressure difference and the mechanism is much more complex. While computational efficiency is an essential aspect of our proposed method, we have not quantitatively compared the performance of the proposed simulator with existing fluid simulation solutions, such as the ones proposed in [16], [26]. Preliminary trials indicate that optimizing the simulation speed increases the maximum possible framerate and allows a higher time scale without causing simulation instability, which improves the training speed in terms of wall-clock time. Further experiments are needed to provide an in-depth analysis and comparison. We plan to enhance the simulation by adding more realistic features and providing quantitative profiling results of the simulation in the continuation of this work.

In this work, the binary mask of the blood region is manually extracted using traditional computer vision algorithms. The detection of blood is more challenging in real surgeries given that the background color can also be close to red. As a next step to address this limitation, we will train an RGB and depth image-based policy without the need for manual feature extraction by building more realistic simulation scenes and using DR and domain adaptation (DA) in the image domain. To achieve this, more realistic fluid rendering and simulation for soft tissue and tool-tissue contact will be achieved in future

work.

Further studies are needed to examine and improve the choice of the reward function, to account for actual surgical requirements. The current reward that considers the removal of blood and the travel of the suction tool might not reflect the actual optimality of the task. Expert demonstrations and feedback can also be utilized in future work.

VI. CONCLUSION

This work explores an RL approach for autonomous blood suction in robotic surgeries. A blood suction simulation environment was developed for RL training using PhysX 5 and Unity, and an agent that utilizes the synthetic blood region mask, the tissue shape, and the suction tool position was trained for autonomous blood suction. We showed the importance of using domain randomization and curriculum learning to achieve a more generalizable policy and showed through experiments that the trained policy can be directly transferred to the real world. In the experiments, the agent was able to perform autonomous suction in various tissue shapes with different amounts and types of liquid, and less than 3 ml of blood remained in the 50 trials, except for one failure. Future work will mainly focus on improving the realism of the simulation in terms of the blood property, tissue shape and deformation, and fluid rendering to allow an RGB image-based policy for real surgical scenarios.

REFERENCES

- [1] C. D’Ettorre *et al.*, “Automated pick-up of suturing needles for robotic surgical assistance,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1370–1377.
- [2] Z.-Y. Chiu, F. Richter, E. K. Funk, R. K. Orosco, and M. C. Yip, “Bimanual regrasping for suture needles using reinforcement learning for rapid motion planning,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7737–7743.
- [3] R. Bendikas, V. Modugno, D. Kanoulas, F. Vasconcelos, and D. Stoyanov, “Learning needle pick-and-place without expert demonstrations,” *IEEE Robotics and Automation Letters*, 2023.
- [4] S. Sen, A. Garg, D. V. Gealy, S. McKinley, Y. Jen, and K. Goldberg, “Automating multi-throw multilateral surgical suturing with a mechanical needle guide and sequential convex optimization,” in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 4178–4185.
- [5] 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,” in *2020 29th IEEE international conference on robot and human interactive communication (RO-MAN)*. IEEE, 2020, pp. 1380–1386.
- [6] N. D. Nguyen, T. Nguyen, S. Nahavandi, A. Bhatti, and G. Guest, “Manipulating soft tissues by deep reinforcement learning for autonomous robotic surgery,” in *2019 IEEE International Systems Conference (SysCon)*. IEEE, 2019, pp. 1–7.
- [7] T. Nguyen, N. D. Nguyen, F. Bello, and S. Nahavandi, “A new tensioning method using deep reinforcement learning for surgical pattern cutting,” in *2019 IEEE international conference on industrial technology (ICIT)*. IEEE, 2019, pp. 1339–1344.
- [8] A. A. Shahkoo and A. A. Abin, “Deep reinforcement learning in continuous action space for autonomous robotic surgery,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 18, no. 3, pp. 423–431, 2023.
- [9] K. Dharmarajan *et al.*, “Automating vascular shunt insertion with the dvrk surgical robot,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 6781–6788.
- [10] A. Pore *et al.*, “Safe reinforcement learning using formal verification for tissue retraction in autonomous robotic-assisted surgery,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4025–4031.
- [11] P. M. Scheikl *et al.*, “Sim-to-real transfer for visual reinforcement learning of deformable object manipulation for robot-assisted surgery,” *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 560–567, 2022.
- [12] A. A. Shahkoo and A. A. Abin, “Autonomous tissue manipulation via surgical robot using deep reinforcement learning and evolutionary algorithm,” *IEEE Transactions on Medical Robotics and Bionics*, vol. 5, no. 1, pp. 30–41, 2023.
- [13] 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*, vol. 8, no. 5, pp. 2502–2509, 2023.
- [14] J. Lai, K. Huang, B. Lu, Q. Zhao, and H. K. Chu, “Verticalized-tip trajectory tracking of a 3d-printable soft continuum robot: Enabling surgical blood suction automation,” *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 3, pp. 1545–1556, 2021.
- [15] F. Richter *et al.*, “Autonomous robotic suction to clear the surgical field for hemostasis using image-based blood flow detection,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1383–1390, 2021.
- [16] J. Huang, F. Liu, F. Richter, and M. C. Yip, “Model-predictive control of blood suction for surgical hemostasis using differentiable fluid simulations,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 12380–12386.
- [17] F. Richter, R. K. Orosco, and M. C. Yip, “Open-sourced reinforcement learning environments for surgical robotics,” *arXiv preprint arXiv:1903.02090*, 2019.
- [18] V. M. Varier, D. K. Rajamani, F. Tavakkolmoghaddam, A. Munawar, and G. S. Fischer, “Ambf-rl: A real-time simulation based reinforcement learning toolkit for medical robotics,” in *2022 International Symposium on Medical Robotics (ISMR)*. IEEE, 2022, pp. 1–8.
- [19] E. Tagliabue, A. R. Pore, D. Dall’Alba, M. Piccinelli, P. Fiorini *et al.*, “Unityflexml: Training reinforcement learning agents in a simulated surgical environment,” in *I-RIM 2020 conference proceedings*, 2020, pp. 0–1.
- [20] J. Xu, B. Li, B. Lu, Y.-H. Liu, Q. Dou, and P.-A. Heng, “Surrlo: An open-source reinforcement learning centered and dvrk compatible platform for surgical robot learning,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1821–1828.
- [21] P. M. Scheikl *et al.*, “Lapgym—an open source framework for reinforcement learning in robot-assisted laparoscopic surgery,” *Journal of Machine Learning Research*, vol. 24, no. 368, pp. 1–42, 2023.
- [22] S. Schmidgall, A. Krieger, and J. Eshraghian, “Surgical gym: A high-performance gpu-based platform for reinforcement learning with surgical robots,” *arXiv preprint arXiv:2310.04676*, 2023.
- [23] P. Ma, Y. Tian, Z. Pan, B. Ren, and D. Manocha, “Fluid directed rigid body control using deep reinforcement learning,” *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–11, 2018.
- [24] M. Chu, N. Thuerey, H.-P. Seidel, C. Theobalt, and R. Zayer, “Learning meaningful controls for fluids,” *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–13, 2021.
- [25] E. Babaians, T. Sharma, M. Karimi, S. Sharifzadeh, and E. Steinbach, “Pournet: Robust robotic pouring through curriculum and curiosity-based reinforcement learning,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 9332–9339.
- [26] Z. Xian *et al.*, “Fluidlab: A differentiable environment for benchmarking complex fluid manipulation,” in *The Eleventh International Conference on Learning Representations*, 2022.
- [27] M. Macklin and M. Müller, “Position based fluids,” *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, pp. 1–12, 2013.
- [28] A. Juliani *et al.*, “Unity: A general platform for intelligent agents,” *arXiv preprint arXiv:1809.02627*, 2020. [Online]. Available: <https://arxiv.org/pdf/1809.02627.pdf>
- [29] B. H. Jafari and N. Gans, “Surface parameterization and trajectory generation on regular surfaces with application in robot-guided deposition printing,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6113–6120, 2020.
- [30] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain randomization for transferring deep neural networks from simulation to the real world,” in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.
- [31] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.
- [32] T. Haarnoja *et al.*, “Soft actor-critic algorithms and applications,” *arXiv preprint arXiv:1812.05905*, 2018.