Computer Integrated Surgery

Group 20: Reinforcement Learning Environment for Robotic Suturing

Team Members: Walee Attia (wattia1), Jocelyn Hsu (jhsu37), Jihoon Kim (jkim620)

Mentors: Dr. Anqi Liu, Dr. Adnan Munawar, Dr. Manish Sahu, Dr. Peter Kazanzides

May 11, 2023

Introduction

Reinforcement learning (RL) is a machine learning framework involved in creating artificial agents that fulfill various complex problems. Surgical robots have opened the door to surgical task automation, which has piqued the interest of RL research. However, no robust framework exists for RL tasks in surgical robotics environments. We propose an OpenAI Gym environment based on an autonomous robotic suturing simulator with benchmark algorithms to pave the way for future surgical automation.

Background

Reinforcement Learning has made significant progress in the robotics domain, enabled by open-source frameworks such as OpenAI Gym [1], which provides effortless implementation of complex algorithms in both simulation and real robots. RL has specifically seen success in robotic manipulation and grasping, with evidence that learned policies are transferable from simulation to real robots [2]. However, RL's success in robotics hinges on having lightweight and efficient simulation environments as it requires thousands to millions of simulated attempts to evaluate and explore policy options, which is crucial for real-world use due to the impracticality of running millions of attempts on a physical system [3].

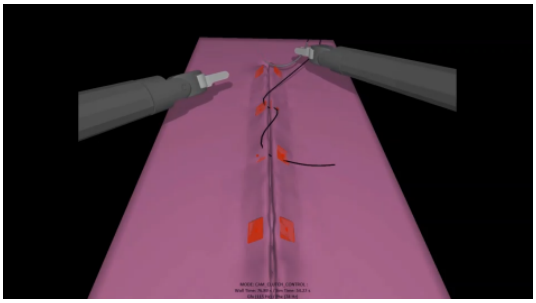


Figure 1: Surgical Robotics Challenge Environment [4]

Thankfully, paradigms in the field of surgical robotics exist in the form of simulators. The 2021-2022 AccelNet Surgical Robotics Challenge [4] is a simulation platform featuring two seven degrees-of-freedom (DOF) large needle drivers (Patient Side Manipulator - PSM) based on the da Vinci Surgical System, a controllable camera based on the da Vinci Endoscopic Camera Manipulator (ECM), a suturing phantom, and a needle with suture. The challenge is based on the Asynchronous Multi-Body Framework (AMBF) simulator [5], which is a real-time dynamics simulator for robots. The challenge is separated into three tasks:

1. Finding the pose of the metallic suture needle with respect to the endoscope pose. The data provided consists of the 3D model of the needle and the camera calibration. This task is not the

focus of our project as the nature of the problem, finding the needle, is more suited for a computer vision approach.

2. Grasping the needle using the large needle driver and driving the needle through the suturing phantom. The ground truth position of the needle is known as well as ground truth positions and orientations of the target entries on the phantom. This task is what we experiment with the reinforcment learning approach.
3. Driving the needle repeatedly through the entry and exit points on the suturing phantom. A single suture consists of the driving the needle through the phantom with the right needle driver, pulling it through with the left instrument, and handing the needle back to the right instrument. Demonstrating challenge two's completion with a reinforcement learning approach should serve as a proof of concept for an RL approach for this third task.

OpenAI Gym is an open source python tool that provides an API that enables reinforcement learning algorithms to communicate with simulation frameworks such as the AMBF real time dynamics simulator, making it uniquely suited for our task. A standardized, open-source environment, which we develop using this tool, enables rapid iteration of reinforcement learning approaches and reward functions, as well as benchmarking of reinforcement learning algorithm's ability to train autonomous robotic simulations. By developing an OpenAI Gym environment built on top of he 2022 AccelNet Surgical Robotics Challenge and the AMBF dynamics simulator, we can bridge the gap between the reinforcement learning and surgical robotics domains. Our aim is to address the lack of reinforcement learning platforms and environments conducive to medical robotics, and we anticipate that an open-source environment will produce wide applicability and drive more innovation in autonomous surgical robotics systems.

Prior Work

The use of surgical robotics has become increasingly popular in recent years. Robotic surgery offers several benefits over traditional open surgery, including improved precision, reduced blood loss, smaller incisions, and faster recovery times. However, robotic systems still require a human operator to control the instruments and make decisions during the procedure. Reinforcement learning (RL) is a type of machine learning that has the potential to improve robotic surgery by allowing robots to learn from their environment and make decisions autonomously.

Our project encompasses development of an RL platform specifically for automating robotic suturing with the da Vinci Research Kit (dVRK). In order to better understand the field of RL applied to the robotic surgery, we explore applications of current RL algorithms in medicine as well as different RL paradigms.

We explored the use of RL in surgical robotics, with a particular focus on two paradigms: Deep Deterministic Policy Gradient (DDPG) and Hindsight Experience Replay (HER). We then reviewed recent research on the application of these paradigms in surgical robotics and their potential benefits and challenges as we synthesize the similarities between their research and our project.

As RL tasks are still emerging, there are limited papers in the intersection of surgical robotics and RL. However, there are two research endeavors that have attempted to bridge this gap: AMBF-RL [5] and dVRL [3].

AMBF-RL is a sandbox toolkit that can be used to design control algorithms for medical robotics in the AMBF simulator. Vignesh Manoj Varier, Dhruv Kool Rajamani, Farid Tavakkolmoghaddam, Adnan Munawar, and Gregory S Fischer present AMBF-RL (ARL), an RL toolkit that enables the design of control algorithms and the collection and processing of expert data from demonstration on real systems. Moreover, they validated the toolkit by simulating a debris removal using a reach task on the da Vinci Research Kit (dVRK) Patient Side Manipulator (PSM) arm. This was done using by finding the optimal RL policy using both DDPG and DDPG + HER (Deep Deterministic Policy Gradient and Hindsight Experience Replay) models before successfully transferring it to the physical system. However, AMBF-RL environment only supports kinematic tasks such has reaching and grasping, which are preliminary in nature compared to a dynamical suturing task.

Similarly, the dVRL project is the first open-source RL environment designed specifically for surgical robotics. One feature of dVRL is that their environment framework is functionally equivalent to OpenAI

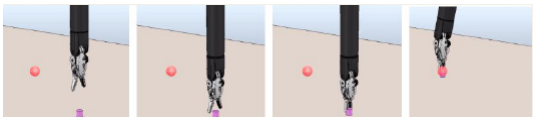


Figure 2: dVRL Reach and Grasp Task [3]

Gym's API, which makes the dVRL environment more accessible to the public, given its open-sourced nature. Again, the dVRL environment can only support simple reach and grasp policies. Richter, Orosco, and Yip are the researchers who introduced this reinforcement learning simulation envi- ronment compatible with dVRK. The goals of the environment are two-fold: to provide a reinforcement learning environment for training surgical robotics, and to transfer the learned policies from simulation to an actual robot.

Goals and Relevance

Our goal is to develop an OpenAI Gym compatible interface for the Surgical Robotics Challenge (SRC) environment with efficient, accurate RL algorithms. An OpenAI Gym environment for the SRC will provide wide-spread use to the masses and will drive further innovation in the interdisciplinary fields of surgical robotics and RL. We also provide baseline RL algorithms for the suturing tasks in the Surgical Robotics Challenge for comparison and evaluation with previous winners of the SRC, with a potential NeurIPS 2023 paper submission on the Datasets and Benchmarks track.

We incorporate results of these studies into our project. To develop our RL algorithm, we incorporate DDPG and HER when selecting the next step to perform in the suturing process that maximizes the reward. Not only is HER compatible with DDPG and built to augment RL algorithms specifically, but it has also been shown to increase learnability of models in both the AMBF and dVRL studies. The OpenAI Gym environment that we design is inspired by the setup of AMBF and dVRL, both of which have publicly available code to reference. Our goal is to construct an efficient and accurate autonomous suturing environment, and build the environment transferable such that it can transferable for other da Vinci Research Kit RL projects.

Technical Summary

Reinforcement Learning

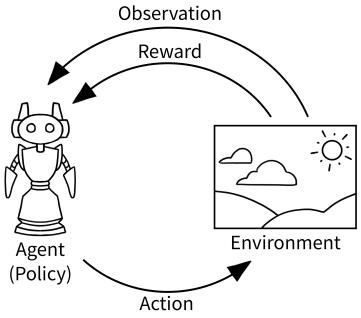


Figure 3: Agent Environment Loop [1]

Reinforcement learning involves an agent interacting with an environment to learn how to take actions that maximize a cumulative reward signal. The agent learns through trial and error, gradually improving its policy for taking actions based on the feedback it receives from the environment in the form of rewards or penalties. As an agent reaches closer to its end goal, it is rewarded positively, and if it missteps and deviates from its goal, it is punished with a negative reward. Due to the agent's ability to manipulate its environment, it will need to observe following each action to assess the subsequent action that will be taken. Our goal is to complete automated suturing with a reinforcement learning algorithm.

The RL framework consists of an agent, an environment, a set of states S, a set of actions A, a reward function R, transition probabilities P, and a discount factor $\gamma \in [0, 1]$. The agent is responsible for taking actions in the environment based on its current policy, which maps states to actions. The environment provides feedback to the agent in the form of rewards or penalties, which the agent uses to update its policy. The state of the environment is typically represented as a vector of features that captures relevant information about the current situation.

The agent's goal in RL is to learn a policy that maximizes the cumulative reward over time. The cumulative reward for a step at time t can be computed with

$$R_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{1}$$

This is typically achieved through the use of a value function or a Q-function, which estimates the expected cumulative reward of following a given policy from a given state or state-action pair, respectively. The optimal value function $V^{\pi *}$ can be updated using the Bellman equation, where

$$V^{\pi *} = \max_a \{R(s, a) + \gamma \{P(s'|s, a\} V^{\pi *}(s')\} \tag{2}$$

which expresses the expected reward in terms of the current maximized reward and the expected future reward given the future state s'.

OpenAI Gym Environment

To accomplish Surgical Robotics Challenge #2, we implemented an overarching OpenAI Gym environment compatible with accomplishing several suturing tasks, including `Grasp`, `Insert`, and `Target`. The environment contains the methods `step()`, `reset()`, `render()`, and `close()` methods implemented, and each task will have customized `reward()` functions for guiding the PSM to completion of the task's ultimate goal.

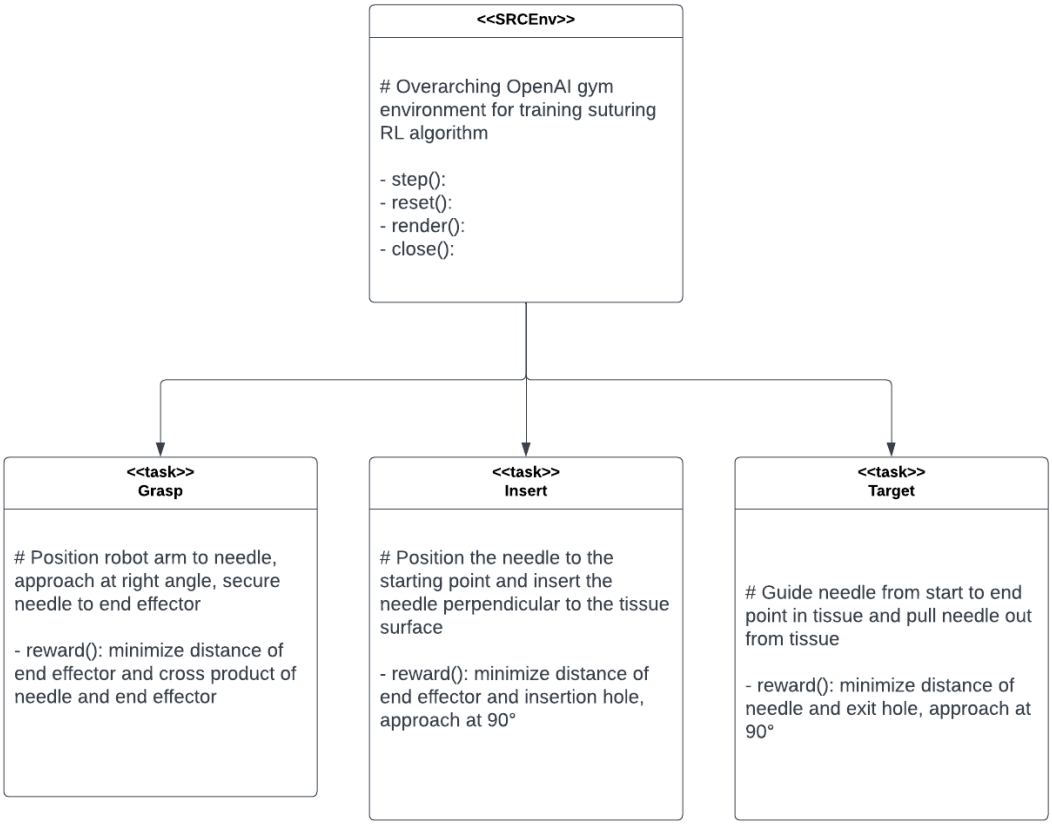


Figure 4: Environment Architecture

`Grasp`

The `Grasp` task will be responsible for handling the initial grabbing of the needle from the surface of the surgical table. Given the positions of the needle and the robotic arm, this task will involve moving the robotic arm to the coordinates of where the needle is located, rotating the joint of the arm for an ideal grasping angle, securing the needle to the end effector of the robotic arm, and picking the needle up.

`Insert`

The `Insert` task will move the needle already grasped in the robotic arm to the starting position provided. Once at the right position and angle, the robot will puncture the needle through the surface of the tissue.

`Target`

The `Target` task guides the needle embedded in the tissue to the target end point. The needle will then be pulled through the end point to complete the single suture.

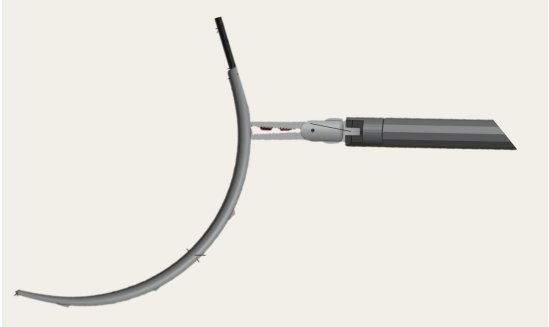
Method development

For each of the sub-environments, the following methods have been developed:

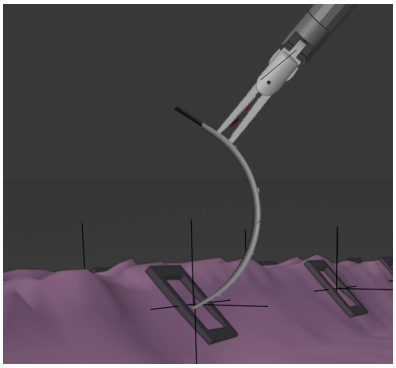
- `step()`
  - Adjusts the robotic arm position based on the calculated action and returns the observation and reward resulting from the action. The robotic arm position can be backtracked to a previous state if a failure mode is encountered, such as dropping the needle after grasping.
  - Parameters:
    - `self`: robotic arm position, needle position
    - `action`: change in robotic arm position
  - Returns
    - `observation`: robotic arm position, needle position
    - `reward`: calculated reward score
    - `terminated`: `True` if task is accomplished, `False` otherwise
    - `truncated`: `True` if failing to accomplish task within certain number of iterations, `False` otherwise
    - `info`: additional diagnostic information
- `reset()`
  - Sets all variables of the environment to its initial state.
  - Parameters:
    - `self`: robotic arm position, needle position
    - `seed` (optional): randomization seed for replication, if needed
  - Returns
    - `observation`: initial robotic arm position, initial needle position
    - `info`: additional diagnostic information
- `render()`
  - Visualize the state of the environment from the agent's perspective.
  - Parameters:
    - `self`: robotic arm position, needle position
  - Returns: None
- `close()`
  - Completes the simulation.
  - Parameters:
    - `self`: robotic arm position, needle position
  - Returns:
    - `Env.unwrapped`: raw state of the environment

Reward algorithm development

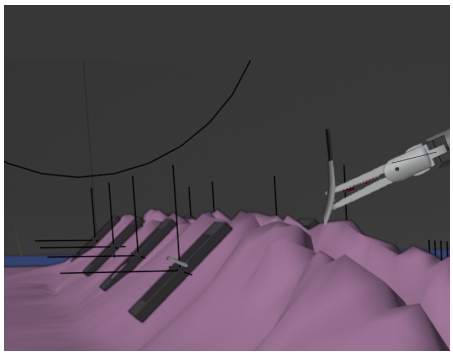
To position the PSM correctly in order to accomplish Challenge 2, we have determined the following positions to be optimal positions for Grasp, Insert, and Target.



(a) Grasp



(b) Insert



(c) Target

Figure 5: Optimal PSM Positions

More details about the reward algorithm can be found in our software design document.

Reinforcement Learning Development

In designing our RL algorithms for the Surgical Robotics Challenge, we employ two advanced techniques: Deep Deterministic Policy Gradient (DDPG) and Hindsight Experience Replay (HER).

Deep Deterministic Policy Gradient [6]

DDPG is a model-free, off-policy actor-critic algorithm tailored for continuous control problems, making it ideal for robotic control tasks. DDPG adapts the widely-used Q-learning algorithm, operating in continuous action spaces by learning a deterministic policy function mapping states to actions. The algorithm utilizes deep neural networks to approximate the value function and policy function, enabling it to manage high-dimensional state and action spaces.

In DDPG, the actor network selects actions based on the current state, while the critic network assesses the chosen actions and the resulting state transition. The actor network is trained using a deterministic policy gradient, optimizing the parameters by minimizing the loss.

$$L(\theta^Q) = E_{s_t \sim \rho^\beta, a_t \sim \beta, r_t \sim E}[(Q(s_t, a_t|\theta^Q) - y_t)^2] \tag{3}$$

where

$$\begin{aligned} &y_t = r(s_t, a_t) + \gamma Q(s_{t+1}, \mu(s_{t+1})|\theta^Q) \\ &\beta \text{ is the stochastic policy} \\ &\rho^\beta \text{ is the discounted policy} \end{aligned} \tag{4}$$

The critic network is trained using temporal difference (TD) learning, updating the action-value function estimate based on the discrepancy between observed and predicted rewards.

Hindsight Experience Replay [7]

HER is a DDPG adaptation designed to address sparse rewards in RL, a common problem in robotics applications. HER adjusts the reward function during training by relabeling the initial goal with the achieved goal in hindsight and computing the reward using the altered goal.

In HER, a modified replay buffer stores both the original and modified transitions resulting from relabeling the goal. During training, the agent samples transitions from this buffer and updates the policy and value functions. HER converts unsuccessful episodes into successful ones, providing more learning opportunities and reducing reward signal sparsity.

HER can be applied to RL agents learning a set of goals G. For all $g \in G$, there exists a state s such that the agent's goal g is achieved, or $f_g(s) = 1$. For a given batch of iterations B, the states and their corresponding actions that achieved transitions between states can be stored in a replay buffer. After updating the replay buffer R, R is sampled, allowing the policy to be optimized based on the history of states and actions, irrespective of the rewards of those actions.

Implementing RL Algorithms with Tianshou RL

Implementing, training, and evaluating the DDPG and HER algorithms described above will be facilitated by the Tianshou library [8]. Tianshou is a reinforcement learning (RL) platform built on PyTorch that supports OpenAI Gym environments. The compatibility between the Python API and our custom-developed SRCEnv is seamless, as both are developed using the Gym environment. Tianshou has demonstrated state-of-the-art benchmark performance on existing Gym environments, particularly the MuJoCo Benchmark, which consists of a collection of 3D robotics RL environments. Tianshou remains the only RL platform supporting the latest version of Gym and is regularly updated and maintained.

The API's design effectively modularizes the reinforcement learning process, streamlining the development and integration of various algorithms into the project. This simplification allows for smoother implementation of the DDPG and HER algorithms, ultimately enhancing the overall effectiveness of the autonomous system.

Outcomes and Results

The main objective of our project was to address the challenges in medical robotics and surgical tasks by developing a specialized OpenAI Gym environment. Our work resulted in several key outcomes, which are detailed in this section.

Establishment of a Baseline Environment

We successfully established a baseline OpenAI Gym environment tailored to the Surgical Robotics Challenge tasks. This environment is the first RL environment for surgical robotics, incorporating realistic physics and robot dynamics, and serves as a foundation for future research in the field. By designing a unique environment that closely mimics real-world surgical procedures, we have created a platform that facilitates the development and testing of novel RL algorithms in a surgical robotics context.

Reward Functions for Grasping and Driving Needle

An essential aspect of our work was the development of customized and tuned reward functions for the specific tasks involved in the Surgical Robotics Challenge. These tasks include grasping the needle, inserting the needle through the tissue, and accurately guiding the needle to the target exit position. Our reward functions were carefully designed to reflect the inherent complexity and precision required in surgical robotics tasks, ensuring that the agent's performance is evaluated based on realistic criteria.

To fine-tune the reward functions, we utilized an ideal trajectory path for completing Challenge 2, allowing us to visualize the relationship between the reward and the distance between the needle and the PSM arm. This approach enabled us to optimize the reward functions for effective and efficient training of the RL agent, ensuring that the agent's performance converges to the desired outcome.

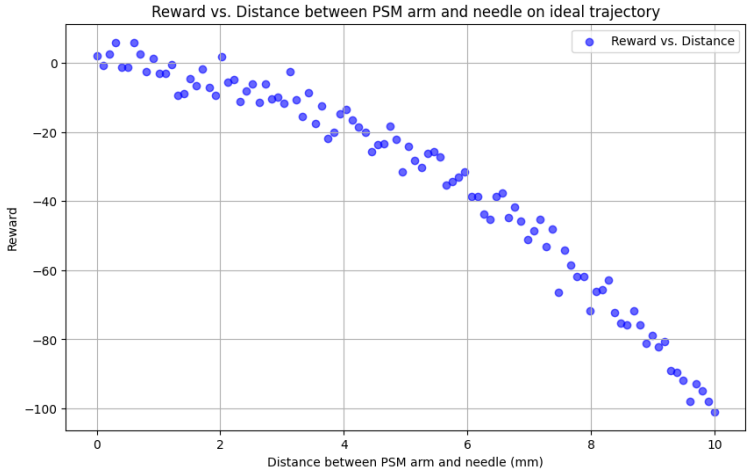


Figure 6: Verification of Reward Function on Ideal Trajectory

To verify the implementation of our reward function, we graph the reward vs the distance between the PSM arm and the needle as shown in 6. This graphs shows that our reward is maximized (reward = 0) when the PSM arm reaches the needle, and decreases as the PSM arm moves farther from the needle.

Implementation of Reinforcement Learning Algorithms

Our project involved the implementation of multiple state-of-the-art RL algorithms, specifically Deep Deterministic Policy Gradient (DDPG) and Hindsight Experience Replay (HER). These algorithms were

selected based on their effectiveness in continuous control problems, making them well-suited for training autonomous robotic simulations in our environment. By leveraging these advanced RL techniques, our project aimed to demonstrate the potential of reinforcement learning in addressing complex surgical robotics tasks.

Although we successfully implemented the DDPG and HER algorithms in our environment, we encountered computational and time limitations that prevented us from completing the training process. These limitations highlight the inherent challenges of training RL agents in high-dimensional, continuous control tasks that require significant computational resources and time.

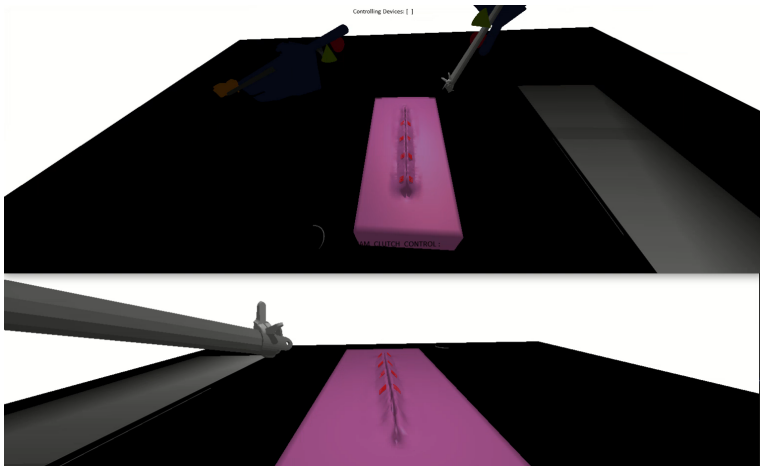


Figure 7: Visualization of RL Training on Grasp Task

We do report that when visualizing the training as seen 7, the models for both DDPG and HER move the PSM arm in a very random and sporatic behavior, and in each episode, we see that the models diverge from the reward, meaning that the reward decreases per time step, which is the opposite of what we would want. This is most likely because the model was trained for a limited amount of epochs, and the actor and critic networks have yet to adapt to the current task of grasping the needle.

Management Summary

Completion of Deliverables

The following contains an outline of our minimal, expected, and maximum deliverables.

- Jocelyn: Transfer of SRC to OpenAI Gym environment (sandbox env)
- Jocelyn: Core functionality for OpenAI Gym: `reset()`, `step()`, `render()`, `close()`
- Jocelyn: Environments built to accomplish SRC Challenge #2: grasp needle and drive through tissue
- Jocelyn, Jihoon, Walee: Documentation of our work through entire development pipeline
- Walee: Compatibility with dVRK in the LCSR lab
- Jocelyn, Jihoon, Walee: Literature review of state-of-the-art (SOTA) RL Algorithms
- Jihoon, Jocelyn: RL algorithm training for automated suturing task (SRC #2)
- Jihoon: Performance evaluation (accuracy) of SRCEnv reward with respect to distance froom goal position

Completion of Dependencies

All dependencies were completed in time in accordance with our project timeline.

Dependency	Need	Source	Date Needed	Status	Contingency Plan
Swipe access to Robotarium, LCSR	Environment Development	Dr. Adnan Munawar	3/1/2023	Completed	N/A
Access to dVRK systems at the Robotarium, LCSR	Test simulation environment	Dr. Adnan Munawar	3/15/2023	Completed	Linux Virtual Machine w/ AMBF + ROS
Rockfish GPU Access	Benchmarking	Dr. Anqi Liu	4/1/2023	Completed	Google cloud
SRC Winning Algorithms	Benchmarking	Dr. Adnan Munawar	4/1/2023	Completed	N/A

Conclusion

In this project, we have made significant strides in the application of reinforcement learning of surgical robotics by developing a specialized OpenAI Gym environment tailored to the Surgical Robotics Challenge tasks, implemented on state-of-the-art RL algorithms. We have also designed customized reward functions to train these autonomous robotic simulations in the environment, using reinforcement learning and imitation learning. Our work serves as a foundation for future research in this field, demonstration the potential of reinforcement learning to address complex tasks in mechanical robotics like autonomous suturing.

As our team members are graduating, we will not be continuing the project full time. However, we intend to publish our current work in conference such as IROS and ICRA 2024 workshops, where our research findings can contribute to the broader scientific robotics community. Our contributions will enable researchers to utilize our implemented environment to create autonomous robotic suturing models.

Next steps will entail continued modification of the reward algorithm. We envision that a piece-wise reward function coupled with segmented goals prior to the end goal of each task will improve performance of the autonomous suturing task. For instance, when grasping the needle, the PSM can first be directed to a point close to the base of the needle (distance reward function), then rotated for optimizing the angle of needle pick-up (angle reward function), and finally directed to the base of the needle (distance reward functions).

RL algorithms will have to continued to be modified, both the architecture and hyperparameters, for successful training and learning of the policy. Although the grasp, insert, and target tasks are relatively simple and could be implemented with an algorithmic approach, the entire suturing process is an inherently complex task. From accidental needle drops to coordination with the 2 PSM robotic arms, an autonomous suturing environment should be capable of reacting to any potential changes in the environment that algorithmic approaches cannot. Thus, our SRCEnv and rudimentary models provide a baseline approach for future sophisticated complete suturing models. Most importantly, however, is that we have implemented an novel environment suited for medical robotic suturing in a dynamics physics simulator, which will pave the road for future breakthroughs in the reinforcement learning industry. We hope this achievement can be expended as a benchmark environment for all future reinforcement learning suturing tasks that is soon to come with the AI boom.

References

[1] OpenAI, "Openai gym documentation: Basic usage." [Online]. Available: https://www.gymlibrary.dev/content/basic_usage/

[2] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2018. [Online]. Available: https://doi.org/10.1109/icra.2018.8460528

[3] F. Richter, R. K. Orosco, and M. C. Yip, "Open-sourced reinforcement learning environments for surgical robotics," *arXiv preprint arXiv:1903.02090*, 2019.

[4] C. R. T. (CRTK), "2021-2022 accelnet surgical robotics challenge." [Online]. Available: https://collaborative-robotics.github.io/surgical-robotics-challenge/challenge-2021.html

[5] 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.

[6] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015. [Online]. Available: https://arxiv.org/abs/1509.02971

[7] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba, "Hindsight experience replay," *Advances in neural information processing systems*, vol. 30, 2017.

[8] J. Weng, H. Chen, D. Yan, K. You, A. Duburcq, M. Zhang, Y. Su, H. Su, and J. Zhu, "Tianshou: A highly modularized deep reinforcement learning library," *Journal of Machine Learning Research*, vol. 23, no. 267, pp. 1–6, 2022. [Online]. Available: http://jmlr.org/papers/v23/21-1127.html