An Exploration of Multi-Agent Reinforcement Learning Algorithms for Human-Robot Interaction Using Physics-Based Simulation

Naveen Balaji
Institute for Robotics and Intelligent Machines
Georgia Institute of Technology
Atlanta, GA, USA
nnagarathinam6@gatech.edu

Jeremy Collins
Institute for Robotics and Intelligent Machines
Georgia Institute of Technology
Atlanta, GA, USA
jcollins90@gatech.edu

***Abstract*—Many robotic applications such as manufacturing and healthcare require humans to interact and work in parallel with robots within a shared workspace. However, planning in such environments remains a challenge due to high degrees of freedom and an intractably large search space. In this work, we model the behavior of robots and humans using multiple reinforcement learning algorithms to tackle this issue with the goal of creating a more efficient motion planning method. In this paper, we design a multi-agent, long-horizon manipulative task aiming to solve assistive tasks in human-robot interactive scenarios. Further, we compare several reinforcement learning algorithms such as Proximal Policy Optimization (PPO), Soft Actor-Critic (SAC), Advantage Actor-Critic (A2C), Behavior Cloning(BC), and Monotonic Advantage Re-Weighted Imitation Learning (MARWIL) to enable robots and humans to collaborate in an incentivized manner to accomplish a pick and place task: the retrieval of a beverage. It was found that the Soft Actor-Critic algorithm produced the best policy for this task, successfully picking and placing a cup near a moving human while avoiding collisions with the human and objects in the environment.**

***Index Terms*—multi-agent, reinforcement learning, simulation, imitation learning**

I. Introduction

Collaboration between robots and humans is a key part of humanity's transition to an automated society. The integration of cobots in manufacturing and healthcare environments has the potential to create a mutualistic relationship that incentivizes the automation of dull and dangerous tasks while protecting the livelihoods of everyday people. In recent years, reinforcement learning has become a popular method of generating control policies in autonomous systems. The ability of reward functions to capture a simple representation of complex environments has allowed reinforcement learning to rival hand-engineered classical control methods. In this work, we will compare several popular reinforcement learning algorithms (Proximal Policy Optimization, Soft Actor-Critic, Advantage Actor-Critic, Behavior Cloning, MARWIL) to enable robots and humans to collaborate in an incentivized manner to retrieve a cup of water for a human agent.

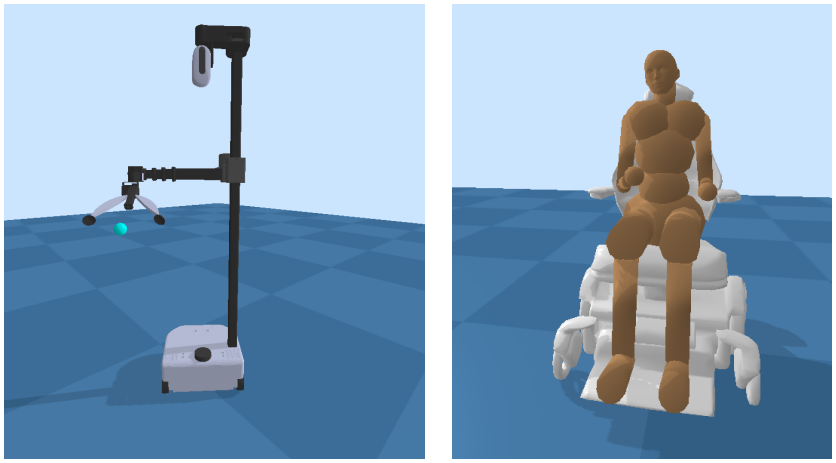
Reinforcement learning algorithms often require large amounts of data to train policies. Physics-based simulation is one of the keys to feeding data-hungry algorithms. Recently several simulations were able to provide realistic synthetic data that has enabled transferring of learned policies to real world [14]. Physics-based simulation also provides a common ground that can be used to create benchmark environments and evaluate algorithms over different robotic tasks, as will be done in this paper.

The task is framed as follows: a human is sitting at a table, actively eating from a bowl. A cup of water is sitting on the other side of the table, out of the human's reach. Our robot, the Hello Robot Stretch RE1 mobile manipulator, is tasked with picking up the cup of water and placing it within the workspace of the human, all while avoiding collisions with the human, collisions with the environment, and spillage of liquid from the cup. All training and testing is performed in physics-based simulation using a specialized version of OpenAI gym called Assistive Gym [10], which allows the ability to implement the co-optimization of the robot and human reward functions, as well as 3D models and kinematics of the Stretch RE1, who's simple dynamics lends itself well to learning control policies to accomplish complex tasks. Each algorithm implementation is created using Ray RLlib [11].

II. Related Works

Many previous works have explored the application of reinforcement learning to human-robot interaction in simulation. Oliff et al. explore deep Q-learning as a method to allow robots to adapt to variable human task performance in a manufacturing setting [1]. Modares et al. employ integral reinforcement learning along with a linear quadratic regulator to control a robot's dynamics in an inner loop while applying a human-adaptive outer control loop [2]. Lowe et al. presented an adaptation of actor-critic methods that considers action policies of multiple agents and is able to successfully learn policies that require complex coordination [4].

Erickson et al [3] developed Assistive-gym, an open source physics-based simulation framework that includes six simulated environments in which a robotic manipulator can attempt to do activities like itch scratching, body manipulation,



(a) Stretch Robot in Simulation (b) Human model in Simulation

Fig. 1. Models of the human and robot agent.

dressing and bathing. All these tasks contains constraint of objects with the robots or with the environment, so the final task is simplified point to point movement task of the robot. Reinforcement learning shows good performance in these tasks due to their simplicity.

III. Methodology

The agents and environment were created using OpenAI Gym [10]. The agents were tasked with the problem of picking up a cup and placing it near a seated human while the human is eating from a bowl. A reward function was engineered to accomplish this task, and various reinforcement learning algorithms were then used to explore the environment and maximize the reward. All algorithms were implemented and trained using Ray RLlib [11].

A. Task Formulation

The goal of this project is to allow a robotic system to learn a pick and place task in a multi-agent setting. The steps are outlined below.

1) Robot: Pick the cup: The robot is initialized in the starting position, near the human and table. The cup is placed at random location on the table. The robot arm should go and grasp the cup with the help of its gripper.

2) Robot: Place the cup in target: Once the cup is grasped from the location, it should be lifted up and moved to its goal. In the process of transporting the cup, the robot should not collide or hit human.

3) Robot: Align to the initial pose: Once the robot puts the cup in the goal position, it needs to retract its arm and move away to its initial position.

4) Human: Pick the food from bowl: The role of the human agent role to perform an eating task, so that the robot can simultaneously provide them enough water to assist them. The human hand should reach the bowl to get food at the initial phase of the task.

5) Human: Place the food in mouth: In second phase, the human agent should move the hand towards their mouth as the goal. Once the task is done for the human agent, the cycle repeats.

All training was performed using the same reward function, robot/human action space, and robot/human observation space. The robot action space is composed of the base rotation, arm extension, and a binary gripping state. The human action space is composed of a parameterization of the right arm joints. The observation space of each agent contained each agents respective state and applied forces, as well as the current time step.

B. Reward Function

1) Robot Reward Function: The reward function of the robot is dependent on several factors, including the distances between the gripper, cup, and goal positions, the gripping force, the cup angle, and whether a collision occurs between the robot and the human. All logical expressions in the reward function should be evaluated as a numerical 0 or 1.

$$R_{robot} = -10C_{human} - 5(D_{cup-grip} + D_{cup-goal}) + 50(F_{grip} \cong F_{grip\_des} \;\&\&\; D_{cup-grip} < \epsilon \;\&\& \; D_{cup-goal} > \epsilon) + 75(D_{cup-goal} < \epsilon) + (50D_{cup\_grip})(D_{cup-goal} < \epsilon)$$

Where:

- C_{human} is a Boolean value indicating whether the robot is imparting force on the human
- $D_{cup-grip}$ is the Euclidean distance between the cup and the gripper
- $D_{cup-goal}$ is the Euclidean distance between the cup and the cup's goal location
- F_{grip} is the gripping force
- F_{grip_des} is the desired gripping force
- ϵ is the error tolerance

2) Human Reward Function: The reward function of the human depends only on the distances between the hand,

mouth, and bowl. The human's only objective is to alternate the hand position between the bowl and the human's mouth.

$$R_{human} = -5(D_{hand\_head})(P == 1)) + 500(D_{hand\_head} < \epsilon) - 5(D_{hand\_bowl})(P == 2) + 500(D_{hand\_bowl} < \epsilon)$$

Where:

- $D_{cup-grip}$ is the Euclidean distance between the cup and the gripper
- $D_{cup-goal}$ is the Euclidean distance between the cup and the cup's goal location
- ϵ is the error tolerance
- P indicates whether the goal position of the hand is at the bowl or the human's mouth - it will toggle between 1 and 2 once the hand is within ϵ of the goal.

C. Algorithms

1) Proximal Policy Optimization (PPO): Proximal Policy Optimization is an adaptation of policy gradient methods which prevents destructively large policy updates while remaining in the trust region of the policy with a clipped surrogate objective [7]. This allows for easier training using multi-threading due to the reduced risk of large policy updates.

2) Soft Actor-Critic (SAC): Soft Actor-Critic is an off-policy algorithm that seeks to maximize both the lifetime reward and the entropy of the policy. Maximizing the policy entropy encourages exploration and avoids repetitive behavior.

3) Advantage Actor-Critic (A2C): Advantage Actor-Critic is an on-policy algorithm which improves the behavior of policy gradients by using the value function as a baseline function to estimate the *advantage* of an action, i.e. how much better taking an action in a specific state is than the average action taken at that state.

4) Behavior Cloning: Behavior Cloning is a form of imitation learning that uses expert demonstrations to learn a policy for a Markov decision process by extracting state-action pairs and minimizing a loss function.

5) Monotonic Advantage Re-Weighted Imitation Learning (MARWIL): Monotonic Advantage Re-Weighted Imitation Learning (MARWIL) is an adaptation of behavior cloning which uses knowledge of the reward in expert demonstrations to estimate the advantage of state-action pairs. A higher sample weight is then attributed to actions with a higher estimated advantage [8].

IV. Physics-Based Simulation

Physics-based simulations are used in creating a virtual environments with similar physics (kinetics, dynamics) to real-world scenarios. We have used the PyBullet engine to simulate a robot and human. PyBullet is a Python module for physics-based simulation used in robotics, games, and visual effects with a focus on training learning-based policies. There are several open source projects that developed simulated environments for robotics tasks such as navigation, or manipulation [15]. Among the reinforcement learning community, OpenAI Gym is one of the most widely accepted frameworks for creating various environments. OpenAI Gym serves as a common interface for a collection of Atari games, board games, 2D, 3D simulations and has benchmark policies for each the environment.

A. Simulation details

1) Stretch: The Hello Robot Stretch RE1 is a mobile manipulator platform, developed in the Healthcare Robotics Lab at Georgia Tech [9]. The Stretch Robot is significantly smaller, lighter, and less expensive than prior mobile manipulators with comparable capabilities. Stretch has a total of 5 degrees of freedom, which includes 2 for the mobile base, 1 for the arm height, 1 for the arm extension, and 1 for the gripper. We have used the stretch URDF model available online to simulate the robotic agent in simulation. The PyBullet physics engine controls all the robot's joints and motors through the position control system.

2) Human: Simulating realistic human in the physics simulation is one of the potential areas of exploration in the research world. Modelling human joints and actions improves the collaboration and feasibility of human-robot interaction. We have used the capsulized human body model provided by the Assistive-Gym environment [3] to train the RL agent. There are several features in the simulated model that can be used to personalize a policy for humans, who have limitations including tremor, joint weakness, and limited range of motion. In the environment which we created we have limited action space only with the right hand, elbow, and shoulders of the human body.

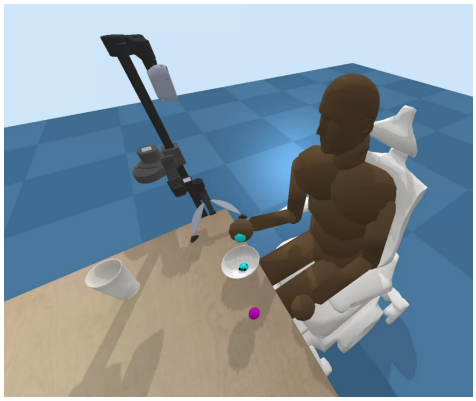
B. Environment

The goal of our project is to develop a collaborative environment where the heterogeneous agents (human and robot) can learn their control policies. We have created a dining table set, where the human's objective is to eat the food, and the robot's objective is to provide a beverage to the selected point to the goal. The agents have a common goal of avoiding collisions.

The robot agent can observe its own orientation, its manipulator joint angles, the position and the orientation of the cup, the force on human, and the goal position to place the cup.

The human agent can observe their hand position, joint angles, the robot's end-effector position, and force applied by the robot.

1) Collecting dataset using teleoperation : In this project, we have explored learning from demonstration methods to perform our task. We collected 10 sets of experiments from simulation to collect these data. The experiments were conducted with the help of keyboard teleoperation, where each of the target joints and motor are mapped to specific keys. The experiments resulted were stored and parsed back to the imitation learning methods such as behavior cloning and MARWIL to learn.



(a) Simulation starting state.

(b) The robot should first grasp the cup.

(c) The robot should then place the cup in the goal position.

(d) Finally, the robot should move to avoid collision with the human.

Fig. 2. Task sequence (teleoperated).

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

Fig. 3. PPO clipped surrogate objective.

$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t\big[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)\big]$$

Fig. 4. PPO loss function.

$$J_V(\psi) = \mathbb{E}_{\mathbf{s}_t\sim\mathcal{D}}\left[\tfrac{1}{2}\left(V_\psi(\mathbf{s}_t) - \mathbb{E}_{\mathbf{a}_t\sim\pi_\phi}\left[Q_\theta(\mathbf{s}_t,\mathbf{a}_t) - \log\pi_\phi(\mathbf{a}_t|\mathbf{s}_t)\right]\right)^2\right]$$

Fig. 5. SAC value network loss function.

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

Fig. 6. SAC Q network loss function.

V. Results

Method	Avg Reward	Std Deviation
SAC	-1262	1567
PPO	-2873	609.8
A2C	-2783	572
MARWIL	-2853	495
Behavior Cloning	-2760	563
Expert Demonstration	33595	-

It was observed that the Soft Actor-Critic algorithm performed the best out of the tested methods, achieving a maximum sliding average reward of 256. It should be noted that the rewards of each episode in the validation batch were not recorded, but had a significant variance. SAC produced the largest maximum episode reward, intermittently producing rewards of over 20000, rivaling the expert demonstration reward of 33000. No other method produced an episode reward above 0.

SAC was also the only method producing meaningful renderings of episodes, with the other methods appearing to get stuck in local maxima of the reward function or effectively following random policies. Models trained on SAC were observed to successfully pick and place the cup in the goal position while avoiding human contact, although the cup was almost always knocked over and dragged to its destination rather than being picked up.

$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}} \left[\mathrm{D}_{\mathrm{KL}} \left(\pi_\phi(\cdot|\mathbf{s}_t) \,\middle\|\, \frac{\exp\left(Q_\theta(\mathbf{s}_t, \cdot)\right)}{Z_\theta(\mathbf{s}_t)} \right) \right]$$

Fig. 7. SAC policy network loss function.

$$\begin{aligned} \nabla_\theta J(\theta) &\sim \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t)(r_{t+1} + \gamma V_v(s_{t+1}) - V_v(s_t)) \\ &= \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) A(s_t, a_t) \end{aligned}$$

Fig. 8. A2C update equation.

VI. Discussion

Through the execution of multiple algorithms in our physics-based simulation environment, we were able to gain intuition for the behavior of each algorithm and which ones do and don't work for our application.

Behavior cloning was not able to successfully accomplish the given task. We believe this is due to the failure of behavior cloning to capture the complexity and time dependence of the task at hand due to a massive assumption - the Markov property. Because behavior cloning attempts to approximate a policy for the expert demonstrations as a Markov decision process, there is an inherent assumption that the next state is dependent only on the current state and independent of the past. This assumption is false - due to the complexity of the given task, there are often multiple "optimal" actions to take in a given state, depending on what stage you are at in the process.

MARWIL fails to accomplish the task for similar reasons - although additional system complexity is captured with the addition of an estimated advantage, the time dependence of the expert demonstrations coupled with the assumption of time independence inherent to the Markov property causes the system to fail. However, it is reasonable to assume that results would significantly improve with the collection of more training data.

For future work, curriculum-based reward generation may be a viable approach for multi-task problems. We spent most of the time fine tuning the reward function for this particular problem. We tried approaching the reward function as discrete multi-phase reward or single continuous reward. In both the cases, the task was not able to complete the indented problem. The paper [13] describes the reverse curriculum based technique where the agent strives to learn starting from the goal, and then progressively searching from the goal to the starting state. This may be a viable method of training or pretraining a policy to accomplish a complex task such as ours.

Within the online learning methods, neither A2C nor PPO were successful at accomplishing the given task. Upon observing renderings of the A2C and PPO policies in action, it was apparent that the policies would easily fall into local maxima and would achieve a very uniform policy, choosing exploitation over exploration. Although more effort could be

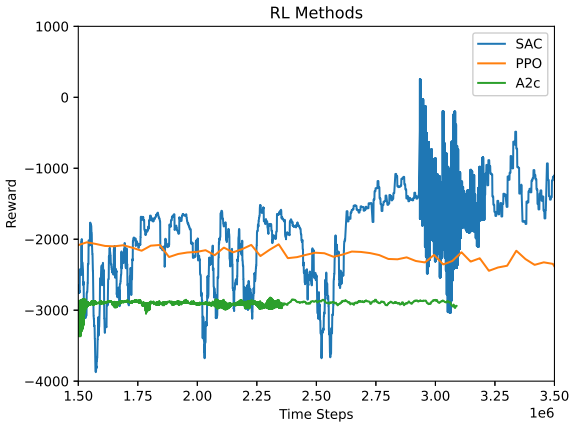


Fig. 9. Sliding average rewards of reinforcement learning algorithms.

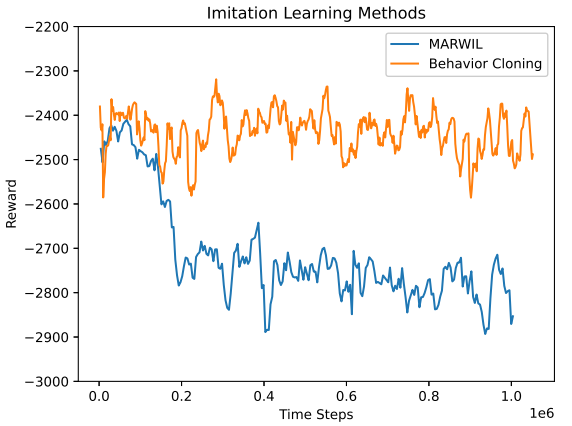


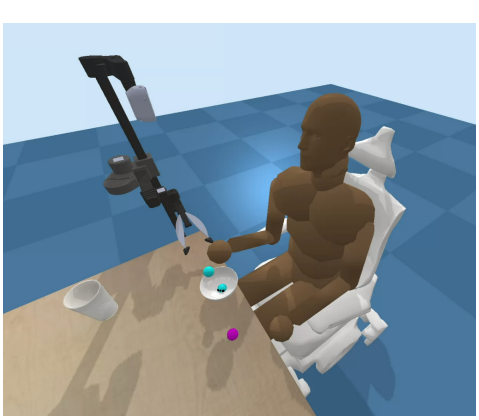
Fig. 10. Sliding average rewards of imitation learning algorithms trained on 10 teleoperated expert demonstrations.

spent tuning hyperparameters and training the models, it is suspected that neither method would approach SAC for this application. We suspect that SAC was able to accomplish the task due to its off-policy, exploration-heavy approach. By attempting to maximize both the lifetime reward and policy entropy, SAC was able to explore more possible state-action pairs without becoming distracted by local maxima, striking a nice balance between exploration and exploitation.

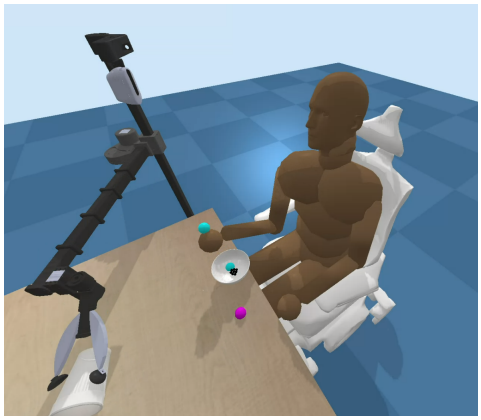
VII. Conclusion

Through our exploration of multiple reinforcement learning algorithms, it was found that SAC, the only off-policy algorithm that was tested, was able to successfully complete the task of delivering a beverage to a human agent. It is suspected that SAC was able to find a balance between exploration and exploitation, following the policy gradient in the behavior policy while fervidly exploring new state-action pairs in the update policy.

No other algorithm was shown to produce any meaningful result in the pick and place task. Advantage Actor-Critic

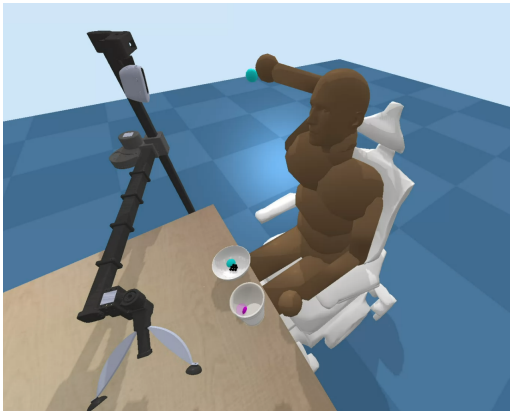


(a) SAC policy start position.



(b) SAC policy: gripping the cup.

(c) SAC policy: placing the cup.



(d) SAC policy: avoiding the human.

Fig. 11. Result rendering from the SAC policy.

and Proximal Policy Optimization, the on-policy algorithms that were tested, performed poorly and were observed to get stuck in local reward maxima and produce low variation in policy. Behavior Cloning and MARWIL, the imitation learning algorithms that were tested, produced results indistinguishable from a stochastic policy due to the failure of the Markov property to capture the complexity of the environment coupled with a lack of training data.

Robotic manipulation is a task performed in the 3D world, possessing rich, complex observations and continuous, high-dimensional action spaces. Navigating in such a complex world is extremely challenging, and we are fortunate to live in a world where the problem is becoming less and less intractable. It would be valuable to evaluate techniques that reduce the dimensionality of the observation and action spaces to simplify this problem, with the hope of improving robustness and allowing for transfer to the real world.

REFERENCES

[1] Harley Oliff, Ying Liu, Maneesh Kumar, Michael Williams, Michael Ryan, “Reinforcement learning for facilitating human-robot-interaction in manufacturing,” Journal of Manufacturing Systems, Volume 56, 2020, Pages 326-340, ISSN 0278-6125, doi: 10.1016/j.jmsy.2020.06.018.

[2] H. Modares, I. Ranatunga, F. L. Lewis and D. O. Popa, ”Optimized Assistive Human–Robot Interaction Using Reinforcement Learning,” in IEEE Transactions on Cybernetics, vol. 46, no. 3, pp. 655-667, March 2016, doi: 10.1109/TCYB.2015.2412554.

[3] Z. Erickson, V. Gangaram, A. Kapusta, C. K. Liu and C. C. Kemp, ”Assistive Gym: A Physics Simulation Framework for Assistive Robotics,” 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 10169-10176, doi: 10.1109/ICRA40945.2020.9197411.

[4] Lowe R, Wu YI, Tamar A, Harb J, Pieter Abbeel O, Mordatch I. Multi-agent actor-critic for mixed cooperative-competitive environments. Advances in neural information processing systems. 2017;30. Doi: 10.48550/arXiv.2103.01955

[5] Ha, H., Xu, J., & Song, S. (2021, October). Learning a Decentralized Multi-Arm Motion Planner. In Conference on Robot Learning (pp. 103-114). PMLR.

[6] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

[7] Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018, July). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In International conference on machine learning (pp. 1861-1870). PMLR.

[8] Wang, Q., Xiong, J., Han, L., Sun, P., Liu, H., & Zhang, T. (2018). Exponentially Weighted Imitation Learning for Batched Historical Data. NeurIPS.

[9] Kemp, C. C., Edsinger, A., Clever, H. M., & Matulevich, B. (2021). The Design of Stretch: A Compact, Lightweight Mobile Manipulator for Indoor Human Environments. arXiv preprint arXiv:2109.10892.

[10] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). Openai gym. arXiv preprint arXiv:1606.01540.

[11] Liang, E., Liaw, R., Nishihara, R., Moritz, P., Fox, R., Gonzalez, J., ...

& Stoica, I. (2017). Ray rllib: A composable and scalable reinforcement learning library. arXiv preprint arXiv:1712.09381, 85.

[12] Coumans, E., & Bai, Y. (2016). Pybullet, a python module for physics simulation for games, robotics and machine learning.

[13] Campo, C. F., Held, D., Wulfmeier, M., Zhang, M., & Abbeel, P. (2018). Reverse Curriculum Generation for Reinforcement Learning.

[14] Bousmalis, K., Irpan, A., Wohlhart, P., Bai, Y., Kelcey, M., Kalakrishnan, M., ... & Vanhoucke, V. (2018, May). Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In 2018 IEEE international conference on robotics and automation (ICRA) (pp. 4243-4250). IEEE.

[15] Fan, L., Zhu, Y., Zhu, J., Liu, Z., Zeng, O., Gupta, A., ... & Fei-Fei, L. (2018, October). Surreal: Open-source reinforcement learning framework and robot manipulation benchmark. In Conference on Robot Learning (pp. 767-782). PMLR.