Analyzing Different Unstated Goal Constraints on Reinforcement Learning Algorithm for Reacher Task in the Robotic Scrub Nurse Application

Clinton Elian Gandana[1], Joel D. K. Disu[2], Hongzhi Xie[3], Lixu Gu[4]
[1,2,4]*Institute of Medical Robotics, School of Biomedical Engineering, Shanghai Jiao Tong University*, Shanghai, China
[3]*Department of Cardiology, Peking Union Medical College Hospital*, Peking, China
[1]clintonelian@sjtu.edu.cn, [2]joel-disu90@sjtu.edu.cn, [3]drxiehz@163.com, [4]gulixu@sjtu.edu.cn

***Abstract*—The main objective paper is to make an empirical analysis of the effect of various unstated spatial goal constraints on reinforcement learning policy for the "reacher" task in the Robotic Scrub Nurse (RSN) application. This "reacher" task is an essential part of the RSN manipulation task, such as the task of picking, grasping, or placing the surgical instruments. This paper provides our experimental results and the evaluation of the "reacher" task under different spatial goal constraints. We researched the effect of this unstated assumption on a reinforcement learning (RL) algorithm: Soft-Actor Critic with Hindsight Experience Replay (SAC+HER). We used the 7-DoF robotic arm to evaluate this state-of-the-art deep RL algorithm. We performed our experiments in a virtual environment while training the robotic arm to reach the random target points. The implementation of this RL algorithm showed a robust performance, which is measured by reward values and success rates. We observed, these reinforcement learning assumptions, particularly the unstated spatial goal constraints, can affect the performance of the RL agent. The important aspect of the "reacher" task and the development of reinforcement learning applications in medical robotics is one of the main motivations behind this research objective.**

***Index Terms*—"reacher" task, spatial constraints, Robotic Scrub Nurse, Reinforcement Learning, Soft-Actor Critic, Hindsight Experiment Replay**

I. Introduction

Robotics research in the medical field is very challenging due to several uncertainties / unexpected events in the environment. The hospital has various workflows and procedures for each room, depending on the function and purpose of the room. For example, workflows and procedures in an operating room are different from workflows in a sanitary room. We also need to simulate a medical robot in an artificial environment to confirm its safety, one of the crucial aspects of medical robotics [1]. However, the involvement of medical robotics is undoubtedly helpful for our medical workers. Some of the works in medical robotics include application in an operating room [2], [3], a robotic arm with laparoscopic instruments [4], Da Vinci R, the most famous surgical robotic since the past decade [5], Robotic Scrub Nurse (RSN) [6], "pick and place" robotic system for surgical instruments by Peenelope CS [7] and Y. Xu [8], and Versius robotic arm for use in gynecology, upper GI surgery, collateral, and urology application [9]. The role of medical robotics in hospitals is essential and can reduce the burden of medical staff. The outbreak of the new virus (Covid-19) exposes an increasing demand for qualified medical staff in hospitals and clinics. We understand the need and importance of hospital automation to reduce the workload and improve the efficiency in hospitals and clinics. One of the solutions to address this issue is to take over the simple and repetitive task of a nurse (i.e., scrub nurse) into an automatic task. The scrub nurse works in the surgery / operating room (as displayed in "Fig. 1") to manage surgical instruments such as scalpels, retractors, hemostats, scissors, forceps, and needle holders. These scrub nurse works could be programmed into a robot called Robotic Scrub Nurse (RSN).

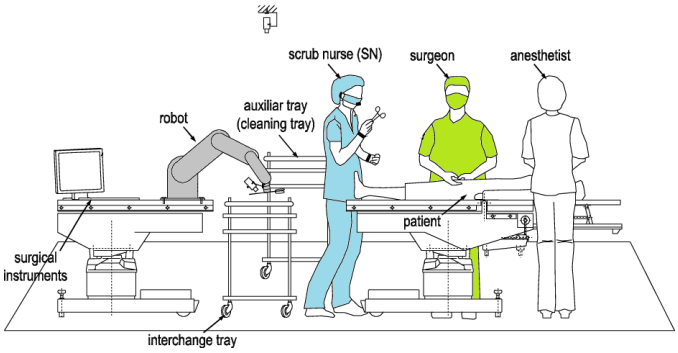


Fig. 1. Common configuration of the surgical room with scrub nurse (SN) staff, anesthetist, and surgeon. Adapted from [10]

The robot scrub nurse (RSN) system can help medical staff to handle the surgical instruments and increase the effectiveness and efficiency in the operating room [10]. The RSN manipulation task mainly divided into grasping tasks and reaching tasks to control the flow of surgical instruments. There are many previous works on the grasping task in the robotic arm, as discussed in [12]–[14]. They developed a method for object picking to grasp and locomote certain objects. In this study, we focused on the "reacher" task because it is the fundamental structure of all kinds of robotic manipulation tasks, including handling the instruments in the operating room [2], [3]. Some prior research of the "reacher" task used motion planning algorithm [11], supervised learning methods, or even manually programmed or "scripted" robot. However, the motion planning approaches have some weaknesses in dealing with uncertainties in the real-world

978-1-7281-9336-6/20/$31.00 ©2020 IEEE

environment, whereas the supervised learning methods have difficulties in training the agent due to the lack of training data. O.I. Borisov et al. [15] discussed the method of controlling a robotic arm with trajectory planning, path planning, and a tracking system used in robotic applications. L.Barbieri et al. designed a master-slave approach to control the robotic arm to reach some predetermined target points [16]. These methods are very dependent on the target position or the determined trajectories and have difficulty if we change the target position or deal with an unseen target position. Moreover, the utilization of reinforcement learning reveals promising results for the manipulation task, as shown in the previous works [17], [18].

In reinforcement learning, the agent learns the learning policy about how to choose the best action from its interaction with the environment through a reward / penalty system. The interaction means, the agent chooses and performs an action for each time step at the reinforcement learning environment. Then, the environment reacts to the action and moves to the next state and gives a scalar signal to the agent named reward. This algorithm makes the robot more robust in dealing with the new environment or unexpected situation in a simulated environment. Moreover, one of the advantages of using reinforcement learning is that the robot arm can move automatically to reach the target point without manual input to control movement.

Reaching task is suitable to be a benchmark for investigating the performance of reinforcement learning algorithms since it is one of the fundamental aspects of robotic manipulation. We define the "reacher" task as a continuous control task to move the end-effector position of the robotic arm to a designed-target position. Deep reinforcement learning algorithm such as Soft-Actor Critic (SAC) [19] plays a role in choosing the best action for manipulation task.

The contribution of this research is to leverage the novel knowledge of reinforcement learning applications in the medical field. Our research provides a novel point of view of the reinforcement learning assumptions, i.e., unstated spatial goal constraints that could affect the performance of reinforcement learning agents to achieve its "reacher" target points.

II. Related Work

Shixiang Gu et al. [17] performed experiments on the robotic arm with reinforcement learning algorithms based on off-policy and Q-function to facilitate manipulation tasks. They compared the empirical performance of the trained agents using DDPG, NAF, and Linear-NAF. The experiments showed the unstated assumptions of RL training, such as variation of several axes, serves as confounding variables in the experiment results. The different settings of hyper-parameter, network architecture, number of agents, reward scaling, random seeds, and environment specifications have a significant impact on empirical learning performance. Cannon Lewis et al. [20] researched unstated general variants of simple manipulation tasks (such as variants of goal constraint region and number of joints) in the experimental setup that could have

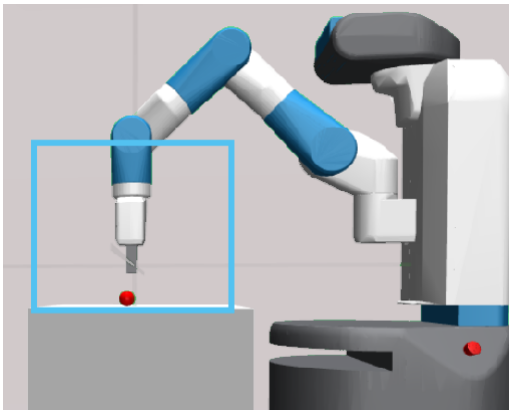
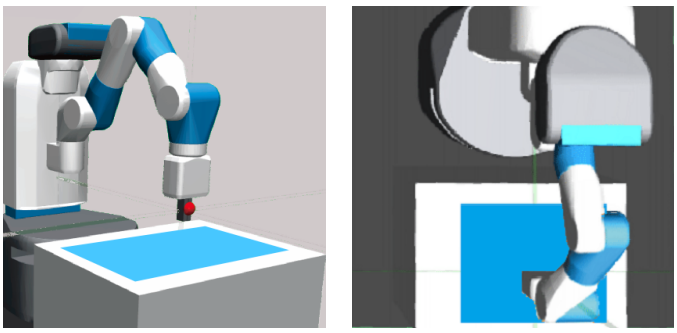


Fig. 2. The illustration of goal constraints (in blue boxes) in the 'Robotic' environment from the side view



(a) (b)

Fig. 3. Illustration of flat surface goal constraint from side view (a) and top view (b), the plat surface is colored by blue

a significant impact on the performance of the trained agent. A.R. Mahmood et al. [21] developed a "reacher" task setup for reinforcement learning experiments on the UR5 robotic arm. The robot, as an RL agent, learns how to reach any arbitrary target positions by trial and error. The learning performance is profoundly affected by unstated setup specifications such as system delays and the choice of action spaces. F. Richter et al. [22] also experimented on the "reacher" task on Patient Side Manipulator (PSM) arm from the da Vinci R Surgical System using an open-sourced reinforcement learning environment. The research displayed that the agent learned the control policies effectively and able to transfer it to the real robots. Zhou T. et al. [6] performed research studies on the robotic arm for the scrub nurse robot application. They combined hybrid computer vision and robotic manipulation to handle the surgical instruments. However, their robotic system can not adapt to new environments or unexpected situations, such as the changes in the target point in delivering the surgical instruments. A trained robotic arm with a reinforcement learning algorithm could address this challenge. In this work, we researched various unstated spatial goal constraints effect on a robotic arm as an RL agent for the Robotic Scrub Nurse application.

978-1-7281-9336-6/20/$31.00 ©2020 IEEE

III. METHOD

A. Experimental Setup

We conducted our experiments using the 3D physics engine and the robotics environment available from the Open AI Gym [23]. Open AI Gym provides an RL environment as a benchmark platform for the algorithms. The Robotics gym environment [24] employs a task-based RL goal and uses the robotic arm as the main agent. Our experiment used a 7-DoF Fetch robotic arm manipulator with a gripper with three-dimensional reaching target points. We also use the Stable Baselines [25], which provides a set of implemented state of the art reinforcement learning (RL) algorithms that can be used in the Open AI Gym environment, i.e., Robotics environment. We implemented the SAC + HER algorithm and the hyper-parameters into the Robotics gym environment with modified spatial goal constraint regions.

We designed the training of the agent to reach the target point inside the spatial goal constraint in a Robotics gym environment (see "Fig. 2"). The target point has a tolerance range of 5 cm, which means the agent "$hits$" the goal if the end effector's position is 5 cm from the target point. In the Robotics gym environment, the state space comes from observing the robot state, such as the gripper state, the joint state, and the goal space. The gripper state consists of gripper positions and velocities. As for the joint state, it includes all positions, rotations, and velocities of the robot's joint. The goal space consists of the desired goal and the achieved goal. The desired goal is the target point that the agent wants to achieve, whereas the achieved goal is the target point that the agent achieved. The Robotics gym environment generates the Cartesian coordinate of the target point in a uniform distribution inside the spatial goal constraint.

The robot moves according to the chosen action space. The action space includes the changing position and rotation of the end effector. The action space has a shape of four dimensions, which consist of three dimensions of position control and one dimension of the gripper control. The three-dimension of position control are related to three dimensions of the Cartesian coordinate. The selected action space changes the array values in position control and gripper control with a maximum value of 0.05. For example, the chosen action space moves the position of the end-effector of the robot to $+0.05$ or -0.05 relative to the current Cartesian coordinate. The state space and the action space described above are the state space s and action space a as the input for the SAC algorithm.

The rewards are given when the effector reaches the target point or when it finishes it one complete episode. Our reward function is designed as '$sparse$ and $binary$' rewards, which means if the agent hits the target, it gets the reward of 0 , and if not, it gets the reward of -1. The reinforcement learning training aims to find the best optimal policies and the highest rewards, which are achieved by interacting with the environment. In our experiment, the RL policy controls the movement of robots by choosing the best course of action.

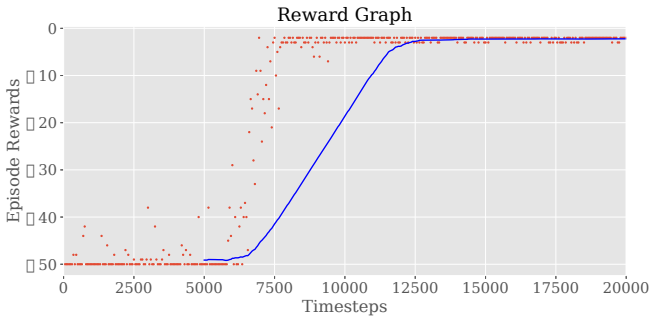


Fig. 4. Reward Graph of a Trained Agent of SAC + HER

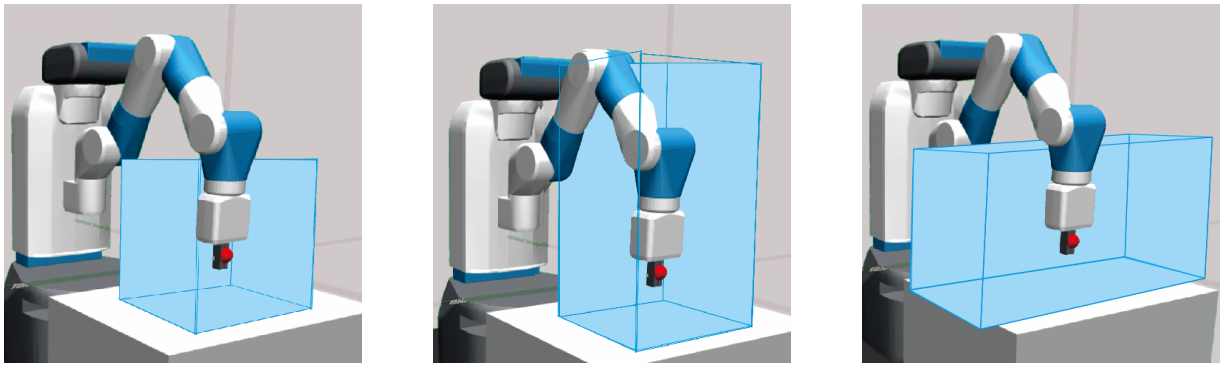
B. SAC & HER Algorithm

Soft Actor-Critic (SAC) [19] is a model-free policy-gradient algorithm, and suitable for solving continuous control problems for robotic applications, i.e., "reacher" task. This maximum entropy reinforcement learning framework uses an off-policy algorithm to optimize the stochastic policy. SAC algorithm has an actor-critic structure with different networks for each of policy and value function. SAC also uses entropy regularization to handle the trade-off between exploration-exploitation and could address the brittleness problem of the agent. SAC combines actor-critic training with a stochastic actor to enable more sample efficiency by reusing the obtained data. Previous research on this algorithm had shown a better efficiency performance compare with the prior policy methods.

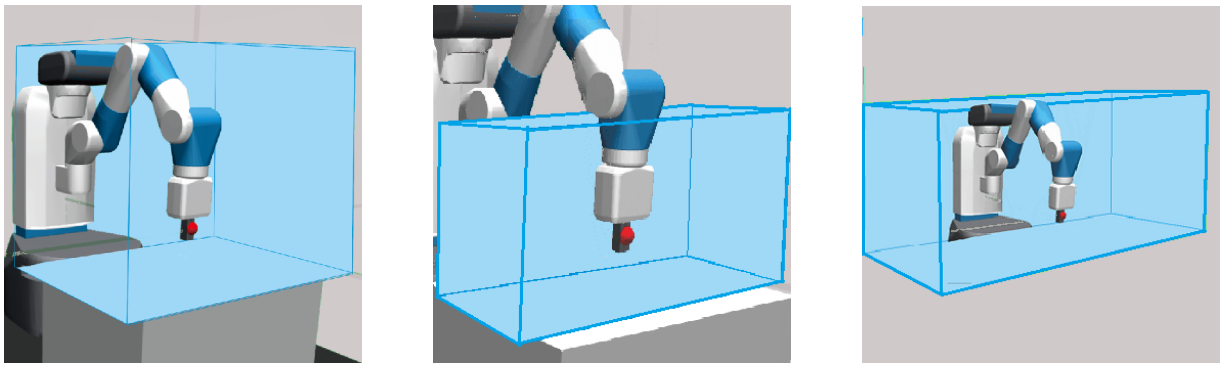
Hindsight Experience Replay (HER) [18] is a sample-efficient algorithm that can handle sparse and binary rewards well. The prior research has already been proved HER as a robust RL algorithm for complicated robotic behaviors (i.e., manipulation task). Besides improving the sample efficiency, HER algorithm could be able to learn policy even if the agent achieves an undesired goal. This ability to learn from the failures, inspired by observing the human ability since all of the experience can always be learned by us. By using this way, all the 'good' or 'bad' experience or trajectories is always useful for the agent. The HER algorithm replays each episode with a different goal than the desired goal and views this undesired goal as a pseudo goal in the process of learning. This method stores the experiences as an experience replay buffer for the off-policy RL algorithm. The previous research showed that the learned policy on the real robot could be used without any finetuning.

The HER algorithm can be used in conjunction with off-policy algorithms such as the Soft Actor-Critic (SAC) algorithm. SAC, combined with Hindsight Experience Replay (HER), improves the performance of the trained agent and the sample efficiency of the algorithm by using experiences to optimize the reinforcement learning policy. The HER algorithm can overcome the challenges of sparse and binary rewards in the continuous control tasks, i.e., manipulation tasks.

978-1-7281-9336-6/20/$31.00 ©2020 IEEE



(a) 0.3×0.3×0.3 goal constraint (b) 0.3×0.3×0.6 goal constraint (c) 0.6×0.6×0.6 goal constraint



(d) 0.6×0.6×0.6 goal constraint (e) 0.3×0.6×0.3 goal constraint (f) 0.6×1.1×0.6 goal constraint

Fig. 5. Illustration of the spatial goal constraints for testing experiments. Goal constraint regions of (a) - (f) for experiments in Table I No 1-15.

IV. Experimental Results

The results of our experiment consisted of two parts: an experiment on SAC + HER for the "reacher" task with a fixed size box-shape goal constraint, and an experiment on the "reacher" task with various goal constraints.

A. Investigation of the Performance of SAC + HER

We conducted our experiment using SAC + HER algorithms with the box-shape goal constraint with the dimension size of $0.3 \times 0.3 \times 0.3m^3$ (the illustration is shown in "Fig. 2"). We trained the agent to reach the target points for 20,000 training episodes. We demonstrated the performance of the RL agent in "Fig. 4". Higher rewards indicate that the trained agent performs well, while lower rewards represent the poor performance of the agent.

The SAC's reward graph increasing after 5,000 training episodes and reach stability after 12.500 time steps. The rewards tend to be stagnant and have little fluctuations near reward with a value of 0. That means the SAC-HER algorithm reaches the optimal policy at 12.500 time steps. Since we applied the sparse and binary rewards system implementation (score of -1 or 0) , the rewards score of -20 or higher shows the agent has explored the meaningful state spaces, got enough experience to move appropriately, and "hit" the target. The agent's policy gave an excellent performance and proved that the agent succeeds in reaching all 5000 random target points in the same constraint size successfully with a 100% success rate.

B. Goal Constraint Experiment

a) Training Phase: We designed our experiment to inspect the performance of the SAC algorithm regarding the unstated assumptions, i.e., spatial goal constraints in a robotic manipulation task. We would like to know whether this assumption could affect the performance of RL agents. In the testing phase, we examined the performance of the agent in carrying out the manipulation task that has one dimension higher than during the training phase. We evaluated the RL algorithm in dealing with this phenomenon in the continuous control problem.

In investigating the performance of SAC + HER, we trained and tested the agent for reaching the target points in the same dimension of goal constraints, which is a box-shape with has a size of $0.3 \times 0.3 \times 0.3m^3$. In this goal constraint experiment, we trained the agent in two-dimensional goal constraints and tested the agent in three-dimensional goal constraints for 20,000 episodes in the Robotics gym environment. The two-dimensional goal constraint is a flat surface on the table that has a dimensional size of $0.3 \times 0.3m^2(\boldsymbol{xy}$-surface), see "Fig. 3". The training was done with a $learning\text{-}rate$ of 0.001, a $buffer size$ of $1,000,000$, and a $gamma$ of 0.95 . In this training, we used $MlpPolicy$ and $SoftActorCritic$ as a

model class. Unlike the previous experiment setup, we tested the performance of the agent with various unseen target points in spatial goal constraints.

b) Testing Phase: We assigned the agent to reach any target points inside the boundaries of novel spatial goal constraint in the Robotics gym environment. We trained the agent in flat goal constraint because we want to examine an agent's performance when dealing with goal constraints that are dimensionally higher than in the training phase. We observed that the scrub nurse has to deliver surgical instruments from mayo tray to the surgeons. In actual cases, the position of the surgeon's hand can be within a certain distance at random. For this reason, we made the various spatial goal constraints as the set up of the testing phase. "Fig.5 " illustrates a picture of the spatial goal constraints for the testing experiments. The smallest spatial goal constraint has $0.3 \times 0.3 \times 0.3m^3$dimension, and the largest constraint has $0.6 \times 1.1 \times 0.6m^3$dimension.

c) Results: We present the $Success\ Rate$ in Table I. As seen in the table, the "reacher" points in various sizes of unstated" goal constraints were able to be reached by the trained agent. In experiment numbers 1, 2, and 3, the SAC + HER policy was able to reach 1,000 random "reacher" points in test episodes with a 100% success rate. Experiment number 4 has the same constraint size as experiment 3 but with a lower success rate, 94.51%. The difference in the values of success rate is caused by differences in the initial state when the agent starts learning. Experiment numbers 5-10 run on 5,000 test episodes. The highest success rates are 100.00% in experiment numbers 5 and 6. while the lowest is experiment number 10, with a 71% success rate. Experiment numbers 10-15 tested with 10,000 test episodes. The experiment numbers 11 and 13 have a 100.00% success rate with $0.3 \times 0.3 \times 0.3m^3$ and $0.3 \times 0.6 \times 0.3m^3$ goal constraints while the lowest success rate is 72.30% with $0.6 \times 1.1 \times 0.6m^3$ goal constraint. We see the correlation between the size of goal constraint and the value of the success rate. The larger the goal constraint, the lower the success rate. The experiments showed that the trained RL agent was able to reach most of the target in different goal constraints with success rates vary from 71.00% to 100.00%.

TABLE I
EXPERIMENTS WITH GOAL CONSTRAINT REGIONS

Table No	Goal Constraint Region			Test Episodes	Success Rate**
	*x axis range**	*y axis range**	z axis range*		
1	0.3	0.3	0.3	1,000	100.00%
2	0.3	0.3	0.3	1,000	100.00%
3	0.3	0.3	0.6	1,000	100.00%
4	0.3	0.3	0.6	1,000	94.51%
5	0.3	0.6	0.3	5,000	100.00%
6	0.3	0.6	0.3	5,000	100.00%
7	0.6	0.6	0.6	5,000	86.96%
8	0.6	0.6	0.6	5,000	81.51%
9	0.6	1.1	0.6	5,000	75.50%
10	0.6	1.1	0.6	5,000	71.00%
11	0.3	0.3	0.3	10,000	100%
12	0.3	0.3	0.6	10,000	95.21%
13	0.3	0.6	0.3	10,000	100%
14	0.6	0.6	0.6	10,000	73.81%
15	0.6	1.1	0.6	10,000	72.30%

*range in m **success number divided by total episodes

V. DISCUSSION AND CONCLUSION

In this work, we examined the performance of state-of-the-art reinforcement learning algorithm: Soft-Actor Critic (SAC) in performing the task which is related to the work of Robotic Scrub Nurse, the "reacher" task. This task is the main requirement for a robotic scrub nurse in managing the surgical instruments. We set up our experiment with various sizes of goal constraint regions and investigated the effect of this unstated RL experiment's assumption on the performance of the SAC + HER algorithm in executing the "reacher" task. We trained our agent in flat-surface goal constraint regions then tested the agent on various sizes of goal constraint regions. We got two conclusions from this work.

1) 1. The RL algorithm is fit for purpose and applicable for the reaching task in RSN application. From our observation, the main requirement for the RSN application is to deliver the surgical instruments from mayo tray to the hand of a surgeon within a specific range, which means the basic task is to reach the mayo tray and to reach the position of the surgeon's hand. These continuous state gradient policy algorithms are robust in dealing with this "reacher" task scenario. The result unveils that the robot can reach most of the target points (up to 100%) in various unseen spatial goal constraints even though the policy did not learn how to reach these points in the training phase. Our result proved that that SAC could deal with unexpected conditions, especially the new target points that the agent did not find during the training phase.

2) We discovered that the SAC + HER is suitable for a "reacher" task since the performance of the trained agent showed a good result. The RSN can use the learned policy since the "reacher" task is mainly used for sorting, pick, and place the task of surgical instruments. For practical surgeon-robotic scrub nurse collaboration, this experiment can utilize novel knowledge about the application of continuous control reinforcement learning algorithms in a medical robotic application, especially in the manipulation task.

3) This research provides a novel point of analysis to compare RL algorithms. Our results (the reward graph and the success rate table) prove that the trained agents have successfully reached the random target positions in new various spatial constraints, which have a higher dimension size than in the training phase. There is an apparent effect on the

decrement in the success rate caused by the increasing size of goal constraints.

This research could be extended as a future work. We would like to support more advanced manipulation tasks such as grasping tasks, and "pick and place" tasks. We want to improve the simulation speed, especially in the training phase, and deal with a more complicated situation. We would like to test state-of-art RL algorithms against other "unstated physical and non-physical assumptions / parameters" such as the different number of joints in the RL training, the effect of different kinds of effectors to the agent and RL algorithms, sparse and dense reward function on continuous control algorithm, and others. The learned policies also could be transferred to the real-world environment, as discussed in the prior works [26]–[30].

We also see that the Fetch Robot is a mobile robot, whose navigation and exploration system can also be improved by using Deep Reinforcement Learning. One of the applications of future medical robotics is the development of reinforcement learning in mobile medical robots that can help and provide medical instruments to the medical staff. To the best of our knowledge, no prior research combines reinforcement learning in both manipulation tasks and navigation tasks for medical robotics in the operating room.

ACKNOWLEDGMENT

This research is partially supported by the National Key research and development program (2016YFC0106200), Beijing Municipal Natural Science Foundation (L192006), and the funding from IMR of SJTU as well as the 863 national research fund (2015AA043203).

REFERENCES

[1] C. Reardon, H. Tan, B. Kannan, and L. Derose, "Towards safe robot-human collaboration systems using human pose detection," IEEE Conf. Technol. Pract. Robot Appl. TePRA, vol. 2015-August, pp. 1–6, 2015, doi: 10.1109/TePRA.2015.7219658.

[2] H. Tan, V. Holovashchenko, Y. Mao, B. Kannan, and L. DeRose, "Human-Supervisory Distributed Robotic System Architecture for Healthcare Operation Automation," 2015 IEEE Int. Conf. Syst. Man, Cybern., pp. 133--138, 2015, doi: 10.1109/SMC.2015.36.

[3] H. Tan et al., "An integrated vision-based robotic manipulation system for sorting surgical tools," 2015 IEEE Int. Conf. Technol. Pract. Robot Appl. (TePRA)}, pp. 1--6, 2015, doi: 10.1109/TePRA.2015.7219664.

[4] A. Brodie and N. Vasdev, "The future of robotic surgery," Ann. R. Coll. Surg. Engl., vol. 100, pp. 4–13, 2018, doi: 10.1308/rcsann.supp2.4.

[5] J. Bodner, H. Wykypiel, G. Wetscher, and T. Schmid, "First experiences with the da VinciTM operating robot in thoracic surgery," Eur. J. Cardio-thoracic Surg., vol. 25, no. 5, pp. 844--851, 2004, doi: 10.1016/j.ejcts.2004.02.001.

[6] T. Zhou and J. P. Wachs, "Needle in a haystack: Interactive surgical instrument recognition through perception and manipulation," Rob. Auton. Syst., vol. 97, pp. 182--192, 2017, doi: 10.1016/j.robot.2017.08.013.

[7] "RST PenelopeCS," 2015. [Online]. Available: http://www.roboticsystech.com/.

[8] Y. Xu et al., "Robotic Handling of Surgical Instruments in a Cluttered Tray," IEEE Trans. Autom. Sci. Eng., vol. 12, no. 2, pp. 775--780, 2015, doi: 10.1109/TASE.2015.2396041.

[9] "CMR Surgical. Press kit," 2019. [Online]. Available: https://cmrsurgical.com/press-kit/. [Accessed: 20-Aug-2019].

[10] C. Pérez-Vidal, E. Carpintero, N. Garcia-aracil, J. M. Sabater-navarro, J. M. Azorin, and A. Candela, "Steps in the development of a robotic scrub nurse," Rob. Auton. Syst., vol. 60, no. 6, pp. 901--911, 2012, doi: 10.1016/j.robot.2012.01.005.

[11] S. X. Yang and M. Meng, "An efficient neural network approach to dynamic robot motion planning," Neural Networks, vol. 13, no. 2, pp. 143–148, 2000, doi: 10.1016/S0893-6080(99)00103-3.

[12] M. Breyer, F. Furrer, T. Novkovic, R. Siegwart, and J. Nieto, "Comparing Task Simplifications to Learn Closed-Loop Object Picking Using Deep Reinforcement Learning," IEEE Robot. Autom. Lett., vol. 4, no. 2, pp. 1549–1556, 2019, doi: 10.1109/LRA.2019.2896467.

[13] Q. Wu, M. Li, X. Qi, Y. Hu, B. Li, and J. Zhang, "Coordinated control of a dual-arm robot for surgical instrument sorting tasks," Rob. Auton. Syst., vol. 112, pp. 1–12, 2019, doi: 10.1016/j.robot.2018.10.007.

[14] C. Choi, W. Schwarting, J. Delpreto, and D. Rus, "Learning Object Grasping for Soft Robot Hands," IEEE Robot. Autom. Lett., vol. 3, no. 3, pp. 2370–2377, 2018, doi: 10.1109/LRA.2018.2810544.

[15] O. I. Borisov et al., "Manipulation Tasks in Robotics Education**This paper is supported by Government of Russian Federation (GOSZADANIE 2014/190 (project 2118)).," IFAC-PapersOnLine, vol. 49, no. 6, pp. 22–27, 2016, doi: https://doi.org/10.1016/j.ifacol.2016.07.147.

[16] L. Barbieri, F. Bruno, A. Gallo, M. Muzzupappa, and M. L. Russo, "Design, prototyping and testing of a modular small-sized underwater robotic arm controlled through a Master-Slave approach," Ocean Eng., vol. 158, pp. 253–262, 2018, doi: https://doi.org/10.1016/j.oceaneng.2018.04.032.

[17] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep Reinforcement Learning for Robotic Manipulation with Asynchronous Off-Policy Updates," 2017 IEEE Int. Conf. Robot. Autom., pp. 3389–3396, 2017, doi: 10.1109/ICRA.2017.7989385.

[18] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba, "Hindsight experience replay," in Advances in Neural Information Processing Systems, pp. 5048–5058, 2017.

[19] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," in Proceedings of the 35th International Conference on Machine Learning, 2018, vol. 80, pp. 1861–1870.

[20] W. C. L. Ii, M. Moll, and L. E. Kavraki, "How Much Do Unstated Problem Constraints Limit Deep Robotic Reinforcement Learning ?," arXiv e-prints, vol. 1909.09282, 2019, doi: 10.25611/az5z-xt37.

[21] A. Rupam Mahmood, D. Korenkevych, B. J. Komer, and J. Bergstra, "Setting up a Reinforcement Learning Task with a Real-World Robot," IEEE Int. Conf. Intell. Robot. Syst., pp. 4635–4640, 2018, doi: 10.1109/IROS.2018.8593894.

[22] F. Richter, S. Member, R. K. O. Member, and M. C. Y. Member, "Open-Sourced Reinforcement Learning Environments for Surgical Robotics," Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst., p. arXiv:1903.02090, 2019.

[23] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI Gym," arXiv preprint. arXiv:1606.01540, 2016

[24] M. Plappert, M. Andrychowicz, A. Ray, B. McGrew, B. Baker, G. Powell, J. Schneider, J. Tobin, M. Chociej, P. Welinder, V. Kumar, and W. Zaremba, "Multi-goal reinforcement learning: Challenging robotics environments and request for research," arXiv preprint arXiv:1802.09464, 2018.

[25] Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher He…, "Stable Baselines", GitHub repository. GitHub, 2018.

[26] . 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," 2017

[27] J. Tobin et al., "Domain Randomization and Generative Models for Robotic Grasping," pp. 3482–3489, 2018.

[28] K. Fang, Y. Bai, S. Hinterstoisser, S. Savarese, and M. Kalakrishnan, "Multi-Task Domain Adaptation for Deep Learning of Instance Grasping from Simulation," 2018 IEEE Int. Conf. Robot. Autom., pp. 3516–3523, 2018, doi: 10.1109/ICRA.2018.8461041.

[29] P. D. H. Nguyen, T. Fischer, H. J. Chang, U. Pattacini, G. Metta, and Y. Demiris, "Transferring Visuomotor Learning from Simulation to the Real World for Robotics Manipulation Tasks," pp. 6667–6674, 2018.

[30] T. Haarnoja, V. Pong, A. Zhou, M. Dalal, P. Abbeel, and S. Levine, "Composable Deep Reinforcement Learning for Robotic Manipulation," Proc. - IEEE Int. Conf. Robot. Autom., no. 1, pp. 6244–6251, 2018, doi: 10.1109/ICRA.2018.8460756.