A Reinforcement Learning Based Approach for Conducting Multiple Tasks using Robots in Virtual Construction Environments

Weijia Cai[1] and Zhengbo Zou[1]

***Abstract*— Construction robots are considered a promising solution for reducing onsite injuries and increasing productivity. One of the bottlenecks in deploying construction robots is solving the problem of robotic motion planning, considering the dynamic nature of construction sites. Specifically, current works in robotic motion planning for construction lack the generalization capacity for different tasks (i.e., a robot is generally optimized for a highly specialized task and fails to generalize when the task deviates slightly from its original form). In this paper, we proposed a reinforcement learning based approach for robotic motion planning using curriculum learning, which enables robots to conduct multiple construction tasks using a single trained agent. We tested our approach on three common construction tasks (ceiling installation, window installation, and flooring), resulting in an average success rate of around 80%.**

I. INTRODUCTION

Robotization of construction tasks has been deemed a promising path to reduce the risks of occupational injuries and increase the productivity of the onsite operations [1], [2]. This is because fundamental tasks onsite, such as reaching, carrying, and placing of construction materials and building objects (e.g., ceiling panel) are physically-demanding and repetitive operations which have the potential to obtain significant improvement through automation [1]. One of the main challenges of robotizing these tasks is the problem of robotic motion planning [3], which refers to producing an optimal trajectory to move onsite objects safely from a starting location to a goal placement.

Construction sites are considered dynamic environments, with constant movements of labor, equipment, and materials, which requires adaptive motion planning for construction robots [4]. In achieving this, two main types of solutions were proposed in previous works. The first type is to preprogram the motions of robots [5], [6]. Preprogrammed robots achieved a limited level of automation, especially when the robot itself [6] or parts of the tooling [5] are designed specifically for a task. These robots are designed to work in isolated and structured environments such as modular construction factories, where the movements and locations of the robots and operated objects are known at all times. However, these designed robots can only be used for the tasks they were designed for; hence lacking the capacity to generalize to other tasks even when they are similar. Another type is to implement and improve on classic motion planning methods such as sampling based motion planning (SMP) [7], [8], which attempt to obtain collision free trajectories by searching for an optimal series of sampled robot configurations given the geometric information of the robots and obstacles. However, the sampling process is computationally expensive if the geometries of the robot or obstacles are complex [9].

Reinforcement Learning (RL) has gained attention for improving the efficiency and generalization performance for robotic motion planning in dynamic environments [10]. In general, RL attempts to train an agent that produces an optimal policy, mapping the observed states (e.g., a robot arm's joint positions) to actions (e.g., forces applied to the joints). RL allows for generalization of multiple similar tasks through Transfer Learning (TL) [11]. TL techniques utilize the transferability of Deep Neural Network (DNN) to learn similarities among different tasks. For instance, many construction tasks can be seen as a special case of the fundamental pick-and-place task; therefore, it is feasible to train a single general agent for pick-and-place with embedded uncertainties such as starting and final placements, and then use the general agent to conduct multiple different but similar tasks such as ceiling installation and window installation.

In this study, an RL based motion planning approach is proposed to conduct three different construction tasks uisng a 6 Degree of Freedom (DoF) robotic arm in a Virtual Construction Environment (VCE). We first built a realistic VCE using a game engine (i.e., Unity3D) as the training and testing environment for the RL agent. We then trained the agent, named the general agent, for the fundamental pick-and-place task. Training of the general agent followed a designed training plan based on Curriculum Learning (CL) [12], which schedules the training by ascending difficulty. Lastly, we tested the generalization performance of the general agent on three common construction tasks, namely, flooring, window installation, and ceiling installation. To further prove the effectiveness of our approach, we trained three additional agents without CL as control group agents. The results were evaluated by the final cumulative reward and the success rate of each task.

II. METHODOLOGY AND EXPERIMENT DESIGN

In this section, training and testing of the general agent will be introduced (see Figure 1). We first provide details regarding the development of the VCE in Unity3D. Next, we introduce details about training of the general agents based on CL. Finally, we provide details regarding the

*We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), [funding reference number: ALLRP 570442-2021]

[1]Weijia Cai and Zhengbo Zou are with the Department of Civil Engineering, The University of British Columbia, V6T 1Z1 BC, Canada. andycai@student.ubc.ca, zhengbo@civil.ubc.ca

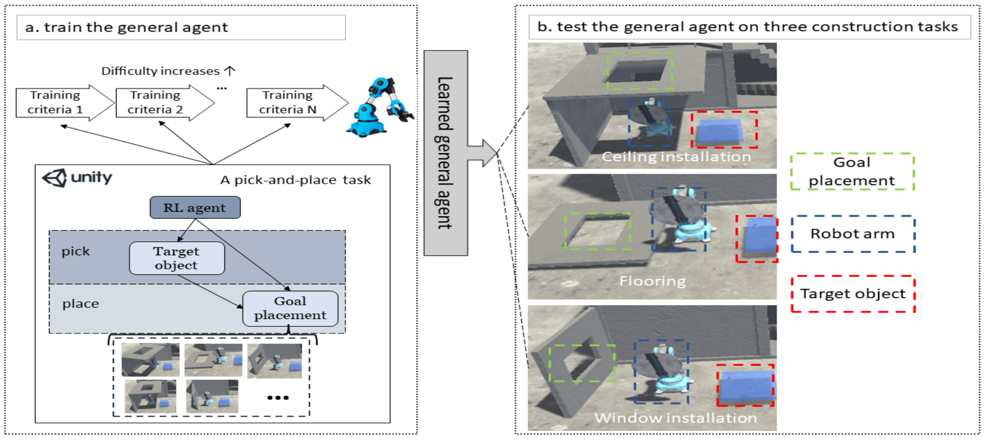

Fig. 1: Overview of the proposed method

experiments designed to test the learned general agent on three construction tasks.

A. *Virtual Construction Environment Development*

The development of the VCE has three main steps, as: 1) set up the physical characteristics (e.g., dimensions, weight) of objects in the VCE; 2) deploy virtual sensors for task completion detection; 3) establish a link between the RL agent and the VCE. For the first step, our VCE was designed as a construction site for a two-story building that includes a 6 DoF robot arm, a target object (i.e., ceiling panel, window, and flooring panel), three types of openings as the goal placement (i.e., goal position and orientation for the target object), and several barriers that make sure the target objects do not exceed the reachable area of the robot. We set the force limit of the robot to be 400 N, which is sufficient for lifting the target object weighted 15 kilograms. Examples of the three types of opening are shown in Fig. 1b.

Then, we set up the virtual sensors to detect whether the robot has completed the following sub-tasks: 1) picking up the target object and 2) placing the target object at a predefined goal placement. For the first sub-task, we used a ray-cast sensor in Unity3D, analogous to an ultra sonic sensor in real-world. The detection range of the sensor is used as the distance tolerance in our training plan. For the second sub-task, a ray-cast sensor with a angle detector is deployed to determine whether the target object is placed at the predefined placement. The angle detector measures the angular tolerance in the training plan.

Lastly, we linked the RL agent to the VCE to provide action control for the robot. In this study, we utilized the ML-Agents toolkits[13], a package that provides a ”brain” module for decision making and an ”agent” module for receiving observation and rewards from the environment. We randomized the starting location of the target objects and the goal placements during training, forcing the robot to gain the ability to adapt to different scenarios onsite.

B. *Training the General Agent*

1) Problem Formulation: An RL agent learns to obtain an optimal policy π^* that maximized the return G_t (i.e., the summation of the discounted future reward) overtime by interacting with the environment, as shown in the equation (1) [14]:

$$\pi^*(A|S) = \arg\max_{\pi} G_t = \arg\max_{\pi} \sum_{k=t+1}^{T} \delta^{k-t-1} R_k \quad (1)$$

In (1), A refers to actions. S refers to the observed state of the robot. The common approach to obtain π^* is to maximize a q value function $q(s,a) = E_\pi(\delta G_{t+1} + R_{t+1}|S_t = s, A_t = a)$ that refers to the expected reward given the current policy (see equation (2)).

$$q^*(s,a) = \sum_{\bar{s},r} \pi^*(a|s) T(\bar{s}|s,a)(r + max_{\bar{a}} q^*(s,\bar{a})) \quad (2)$$

In (2), T is a transition function that maps the current state s and action a to the next state $\bar{s}$; r is the reward from the environment at state s. Due to the complexity of updating the q value, modern RL algorithms such as Policy gradient (PG) methods [15] tend to use neural network to approximate the policy function and q value function. In this study, we used Proximal Policy Optimization (PPO) [16] for its stable performance.

2) General Agent Reward Design: The general agent aims to learn a fundamental pick-and-place task that normally has two sub-tasks: picking up the target object and placing the target object at a predefined goal placement. For picking, the agent aims to reach the target object within the distance tolerance. After picking up, the target object is attached to the gripper panel by adding a constant force from the target object to the panel. For placing, the agent attempts to place the target object at a predefined goal placement within the distance and angle tolerance.

The reward function design is based on the goals of completing the above sub-tasks. Designing a successful reward function is one of the most important parts in RL [17] as it decides the success of the task completion. In this study, the reward function R is composed of two parts, including an extrinsic reward R^E and a curiosity-based intrinsic reward R^I [18], shown in Fig. 2.

```
Algorithm 1: Reward function
1 for episode $e = 1$ to $N$ do
2     for training step $t = 1$ to $MaxStep$ do
          # get intrinsic reward from the agent's exploration#
          $R^I_{e,t}$ = the distance between the next state's predicted feature and the ground truth
          # get extrinsic reward from the environment#
          $R^E_{e,t} = -\frac{1}{MaxStep}$
          if Grasped then
              $R^E_{e,t} = 1$ (only assigned for the first time)
              if Placed then
                  $R^E_{e,t} = 2$
                  # get total reward#
                  $R_{e,t} = R^I_{e,t} + R^E_{e,t}$
                  End current episode
          else
              continue
```

Fig. 2: The reward function design

3) Curriculum Learning based Training Plan: To make stable improvements during training, we designed a CL-based training plan for the general agent (see Table I). Table I contains three training criteria: 1) the distance tolerance for reaching the target object τ_{pick}; 2) the distance tolerance for reaching the goal placement τ_{place}; 3) the angle tolerance for placing the target object τ_{angle}. Every row in Table I represents a curriculum plan of the agent. The model parameters (i.e., policy function, q value function, and the intrinsic reward function) of the general agent starting from cp2 inherits model parameters resulted from the former curriculum plan.

TABLE I: Curriculum Plan (cp) for the general agent

Training criteria	τ_{pick} (m)	τ_{place} (m)	τ_{angle} (degree)
cp1	0.40	0.50	80.00
cp2	0.30	0.25	60.00
cp3	0.30	0.13	40.00
cp4	0.30	0.10	20.00

C. Testing Generalization of the General Agent on Three Construction Tasks

We tested generalization performance of the general agent on three construction tasks, namely window installation, ceiling installation, and flooring. These three tasks can be seen as a pick-and-place task but are different in the operations of placing the target object. More specifically, for window installation, the window should be placed vertically at the goal placement; for ceiling installation, the ceiling panel should be placed horizontally above the robot; for flooring, the flooring panel should be placed horizontally on the ground. To further test the effectiveness of our training plan, we trained three additional agents for the three tasks without the curriculum plan, as a control group. The control group agents (i.e., controlGroup4.1, controlGroup4.2, and controlGroup4.3) used the same training criteria as cp4 shown in Table I. We used the cumulative reward and the success rates of picking $\frac{N_{pick}}{N_{episode}}$ and placing $\frac{N_{place}}{N_{episode}}$ as evaluation metrics.

III. RESULTS AND DISCUSSION

In this section, we show the training results of the general agent and the comparison between the general agent and the control group agents.

A. Training Results of the General Agent

The training results of the general agent are shown in Fig. 3. Each line of the same color base refers to the cumulative extrinsic reward over two million action steps. The transparent line around the solid line represents the raw data while the solid line refers to the moving average of the raw data. From Fig. 3, we can see that cp1 and cp2 converged to a reward around 2.5 while the cp3 and cp4 converged to a slightly lower reward of around 2.1. This is because the acceptable placing of the target object for each type of goal placement was similar in cp1 and cp2. However, the acceptable placing became distinct for different goal placement in cp3 and cp4 with smaller angle tolerance. Overall, cp1 to cp4 all approached the upper bounds of the cumulative reward of 3 (i.e., 1 for picking and 2 for placing).

B. Comparison between the General Agent and the Control Group Agents

The cumulative reward results of cp4 and the control group agents are shown in Fig. 4. It can be observed that the final cumulative rewards of all control group agents were lower than that of cp4. This indicates that the training criteria of cp4 might be too difficult for the agents to learn without prior knowledge.

We tested the generalization performance of the general agent by executing the inference of the learned general agent for an additional 50K action steps for each construction task. The success rates of the testing (i.e., cp4) and the control group agents are shown in Table II. It can be that the general agent achieved higher success rates in all three tasks. To elaborate, the control group agents have an average picking success rate of around 63%, much lower than the average testing result (89%) of the general agent. For placing, the control group agents have an average success rate of around 1.7%, indicating that the curriculum plan is effective for stable improvement of the training. It can also be observed that the placing success rate (68.04%) for ceiling installation is lower than the other two tasks (84.82% and 85.17%). An important reason is that the motion planning for ceiling installation is more difficult than the other tasks. For instance, the agent is supposed to flip the robot's gripper to place the ceiling panel because the goal placement is above the robot. On the other hand, the agent can simply rotate the base joint to place the floor panel since the initial orientation of the panel remains the same as its goal placement.

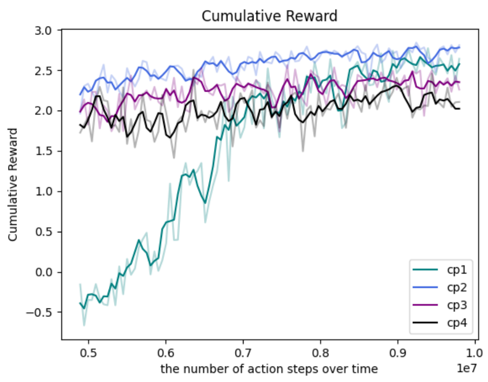

Fig. 3: Cumulative reward over the number of action steps for the general agent

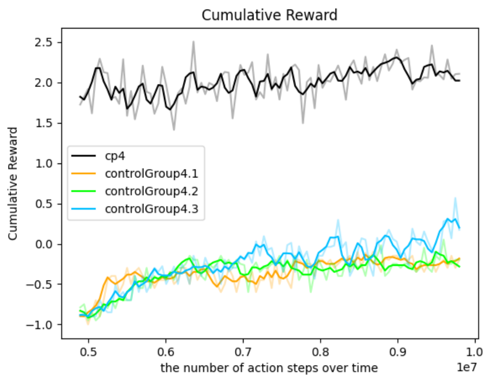

Fig. 4: Cumulative reward over the number of action steps for the cp4 and the control group agents

IV. CONCLUSION

In this study, we explored the generalization performance of the proposed RL based robotic motion planning approach in a virtual construction environment. We first built a realistic virtual construction environment. Then we trained the general agent following a designed curriculum learning based training plan. We showed the generalization performance of the general agent by testing it on three common construction tasks, namely ceiling installation, window installation, and flooring. Finally, we showed the effectiveness of the designed curriculum plan by comparing the results of the control group agents and the general agent. Only three common construction tasks were tested using the approach, but our approach can be easily extended to other construction tasks such as paneling and framing. We also noticed that the generalization performance of the general agent differs among the tested tasks due to the difference of the task difficulties, which is the main limitation of this study. In the future, we intend to design a more flexible algorithm to better distinguish the difference among tasks, since the current RL agent design only captures the similarity among the tasks.

TABLE II: The success rates of cp4 testing and the control group agents

Construction task	Training criteria	$\frac{N_{pick}}{N_{episode}}(\%)$	$\frac{N_{place}}{N_{episode}}(\%)$
Ceiling installation	controlGroup4.1/cp4	63.00/84.55	0.21/68.04
Window installation	controlGroup4.2/cp4	62.89/**95.32**	0.30/84.82
Flooring	controlGroup4.3/cp4	64.72/88.43	4.45/**85.17**

REFERENCES

[1] J. G. Everett and A. H. Slocum, "Automation and robotics opportunities: construction versus manufacturing," *Journal of construction engineering and management*, vol. 120, no. 2, pp. 443–452, 1994.

[2] T. Bock, "The future of construction automation: Technological disruption and the upcoming ubiquity of robotics," *Automation in construction*, vol. 59, pp. 113–121, 2015.

[3] K. M. Lundeen, V. R. Kamat, C. C. Menassa, and W. McGee, "Autonomous motion planning and task execution in geometrically adaptive robotized construction work," *Automation in Construction*, vol. 100, pp. 24–45, 2019.

[4] C. Feng, Y. Xiao, A. Willette, W. McGee, and V. Kamat, "Towards autonomous robotic in-situ assembly on unstructured construction sites using monocular vision," in *Proceedings of the 31th International Symposium on Automation and Robotics in Construction*, 2014, pp. 163–170.

[5] N. Kumar, N. Hack, K. Doerfler, A. N. Walzer, G. J. Rey, F. Gramazio, M. D. Kohler, and J. Buchli, "Design, development and experimental assessment of a robotic end-effector for non-standard concrete applications," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1707–1713.

[6] K. Iturralde, M. Feucht, R. Hu, W. Pan, M. Schlandt, T. Linner, T. Bock, J. Izard, I. Eskudero, M. Rodriguez, *et al.*, "A cable driven parallel robot with a modular end effector for the installation of curtain wall modules," in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, vol. 37. IAARC Publications, 2020, pp. 1472–1479.

[7] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[8] S. M. LaValle *et al.*, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[9] C. Yu and S. Gao, "Reducing collision checking for sampling-based motion planning using graph neural networks," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[10] S. Gu, T. Lillicrap, I. Sutskever, and S. Levine, "Continuous deep q-learning with model-based acceleration," in *International conference on machine learning*. PMLR, 2016, pp. 2829–2838.

[11] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Advances in neural information processing systems*, vol. 27, 2014.

[12] 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.

[13] A. Juliani, V.-P. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar, *et al.*, "Unity: A general platform for intelligent agents," *arXiv preprint arXiv:1809.02627*, 2018.

[14] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[15] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in neural information processing systems*, vol. 12, 1999.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[17] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 1.

[18] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *International conference on machine learning*. PMLR, 2017, pp. 2778–2787.