

Efficient Robot Curriculum Approach for RGBD Indoor Robot Navigation Using DRL Method

Pooja Upadhyay¹, Dr. Bappaditya Jana²

¹Research Scholar, Sheat College of Engineering, Varanasi, India

²Professor and Head, Sheat College of Engineering, Varanasi, India

ARTICLE INFO

Article History:

Accepted: 07 May 2023

Published: 22 May 2023

Publication Issue

Volume 10, Issue 3

May-June-2023

Page Number

1187-1192

ABSTRACT

Training robots to navigate diverse environments is a challenging problem as it involves the confluence of several different perception tasks such as mapping and localization, followed by optimal path-planning and control. Recently released photo-realistic simulators such as Habitat allow for the training of networks that output control actions directly from perception: agents use Deep Reinforcement Learning (DRL) to regress directly from the camera image to a control output in an end-to-end fashion. This is data-inefficient and can take several days to train on a GPU. Our paper tries to overcome this problem by separating the training of the perception and control neural nets and increasing the path complexity gradually using a curriculum approach. Specifically, a pre-trained twin Variational AutoEncoder (VAE) is used to compress RGBD (RGB & depth) sensing from an environment into a latent embedding, which is then used to train a DRL-based control policy.

A traditional path-planner is used as a guide for the policy and the distance between start and target locations is incrementally increased along the A route, as training progresses. We demonstrate the efficacy of the proposed approach, both in terms of increased performance and decreased training times for the PointNav task in the Habitat simulation environment. This strategy of improving the training of direct-perception based DRL navigation policies is expected to hasten the deployment of robots of particular interest to industry such as co-bots on the factory floor and last-mile delivery robots.

Keywords: Mobile Robots, Autonomous Navigation; Collision Avoidance; Reinforcement Learning;

I. INTRODUCTION

Robots To go from point A to B in an indoor environment is challenging for a mobile robot. In the absence of GPS and using the visual/RGBD sensor

available on the robot, one has to map an environment & localize in it (SLAM) and then path-plan an obstacle-free route to get from a start to target location. This was the traditional approach to mobile robotics. Recently, Deep Reinforcement Learning

(DRL) has shown to provide more robust navigation policies compared to SLAM, if the robot (agent) is trained in simulation and exposed to an order of magnitude more experience [1]. This involves training navigation policies that regress directly from the camera image to a control output.

However, splitting this task into two: learning a compact state representation, termed “representation learning” and then using this representation to learn a robust control policy has the following advantages: (1) errors in policy learning will not affect perception as the latter is decoupled from the former, but the vice versa is not true; (2) once perception is learned, it can be reused to learn multiple policies for different tasks, which is not feasible in complete end-to-end training as the perception needs to be re-learned every time a new task is learned. These advantages have the potential to speed-up the overall learning of the task at hand.

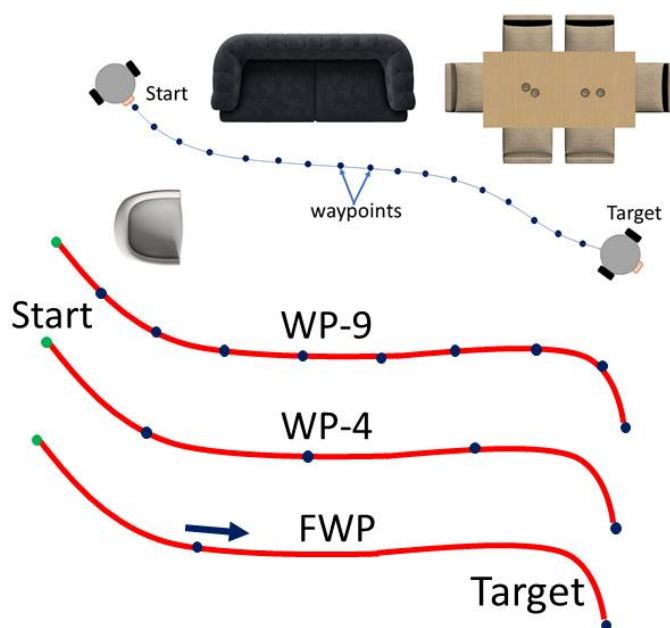


Figure 1: waypoints generated between desired start and target locations by the A algorithm. This work looks at assisting the training of a DRL-based robot navigation policy, by incrementally increasing the difficulty of the navigation task in a curriculum. Bottom: 3 curriculum training approaches with 9 and

4 discrete waypoints and a continuously moving waypoint: WP-9, WP-4 & FWP.

II. OUR SOLUTION

The recently released Habitat simulator [1] has generated excitement in the field of RGBD vision-based robot navigation in indoor environments. In our solution, we train DRL agents for the problem of indoor robot navigation in the Habitat environment by separating perception (i.e., representation learning) and control (i.e., navigation policy). We use a VAE to encode RGB and Depth images, and use these latent encodings as well as a reading and heading angle for the target (from the PointGoal sensor), to learn navigation policies. Additionally, we use a traditional path-planner, A to assist the DRL agent during training, by following a pre-determined curriculum. A guides the agent by giving it shorter-distance goal locations (waypoints) between the original start and target locations.

We experiment with two different curriculum-based training of the DRL agents, one by decreasing the number of intermediate waypoints used (termed the SWP-N agent) or by moving the episodic goal farther away from the start position (termed the FWP agent). We describe the problem and our method in more detail in our paper. In summary, our contributions are as follows: (1) a principled approach to compare different navigation-agnostic VAE-based perception embeddings for their usefulness to a DRL in learning a subsequent navigation policy; (2) Using a traditional A path-planning algorithm in a curriculum fashion to assist in the training process of this navigation policy. This two-step procedure helps in the speed-up of the overall training of the policy network to learn robust navigation policies.

III. THE POINTNAV TASK

We use the Habitat simulator [1] to train our DRL to learn policies for the point-goal navigation task in the

Gibson environment [2]. The robot/agent is equipped with an RGBD camera, a point-goal sensor and a heading sensor. The point-goal sensor is like an indoor GPS: it provides the agent with its current position and the relative position of the target location. The heading sensor provides the current global heading angle of the agent. In the pointgoal navigation task, the agent is asked to navigate from the initial starting position to the required end position using only its RGBD, heading and point-goal sensors and without a map. These start and target locations are randomly initialized at the beginning of each episode, for which no straight line path is possible. The agent needs to learn navigation strategies that avoid obstacles and negotiate doorways since the start and target locations can be in different rooms.

IV. TWIN-VAE (REPRESENTATION LEARNING)

We pre-train perception in the environment by using a twin-VAE setup as shown in the figure below. RGBD cameras are initialized randomly in the environment, and at each location, RGB and depth images are collected. These images are used to train the RGB and depth encoder-decoder branches (blue and purple in the figure). Once the VAE is pre-trained, only its encoders are used for training the DRL policy. RGB and depth images are encoded to their respective embeddings, which are concatenated to provide the final visual embedding from the camera. This embedding is used for training the DRL policy.

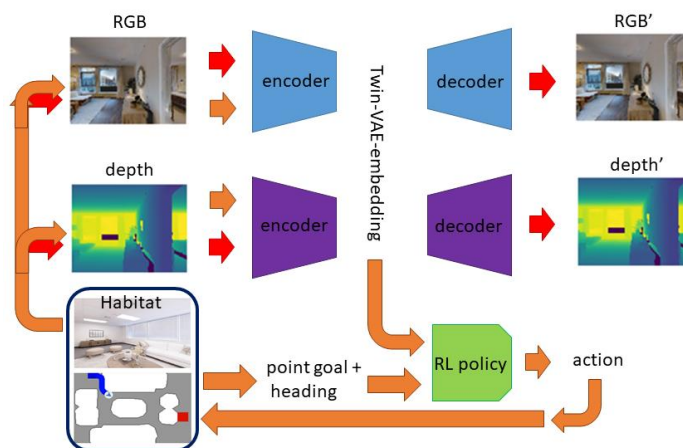


Figure 2: A twin (RGB-depth) VAE learns an embedded representation of the environment, which

is then used to train a navigation policy using DRL. Information flow during the VAE and DRL training are shown in red and orange respectively.

V. A CURRICULUM LEARNING

The task of learning the DRL policy is assisted by incrementally increasing the difficulty of the PointNav task. We do this during training by using A to determine an optimal path between start and target locations in the bird's eye view (BeV) map of the environment. A new sub-goal, a point to navigate to, that is on this A path, is provided to the DRL. This sub-goal is close to the starting location to begin with, and then as training progresses, gets farther and farther away from it. We test the following variants of curriculum learning based on discrete and continuous subdivisions of the path:

1) **WP-N**: In Way-point-N or WP-N, the A path is divided into N equidistant waypoints (WPs) including the target location. At the beginning of the training episode, the agent is asked to navigate to the first WP. When it reaches within 0.2m of this WP, the goal is revised to the next one and so on till the final target location. We investigate the number of intermediate waypoints required for successful navigation by experimenting with WP-10, WP-8, WP-6, WP-4, WP-3 and WP-2. WP-1 involves no subdivisions of the path and is the same as the original PointNav task.

2) **SWP-N**: Sequential WP-N or SWP-N involves keeping the number of WPs constant for a fixed number (few thousand) episodes. This is the same as WP-N, where the agent is asked to navigate from the 1st to the Nth waypoint within the same episode. However, N decreases episodically. Once the agent has mastered a higher N, requiring a smaller length sub-path traversal, the agent is subjected to a lower N, requiring a larger length sub-path traversal.

3) FWP: Farther Waypoint involves only one WP that moves farther and farther away from the start in continuous, linear increments, as training progresses. The training is commenced with the WP at 20% distance along the A path from the start. Over the course of training, this WP is moved farther and farther along the path until it is at 120% of the distance from the start to target after several tens of thousands of episodes, at which point the FWP problem is the same as the PointNav problem.

VI. DRL TRAINING (CONTROL LEARNING)

At any time instant, the sensory readings of the RGB and Depth images are encoded into one-dimensional vectors/embeddings using the pre-trained twin VAEs. These are then concatenated with the pointgoal sensor reading, and the heading angle, to obtain a compact representation of the state at time instant t , s_t . We use Deep Reinforcement Learning (DRL) to learn a policy π_θ that outputs action at the time t : $a_t = \pi_\theta(s_t)$, where the actions are one of three: (1) move forward by 0.25 m; (2) turn left by 10 degrees; (3) turn right by 10 degrees. A fourth action called "Done" is executed whenever the agent is within 0.2 m from the goal position. Specifically, the Proximal Policy Optimization (PPO) algorithm is used with the policy network being a neural network with fully connected layers and an LSTM for temporal information. See our paper for more details on the architecture used for the neural networks.

VII. RESULTS

The success-weighted path length (SPL) is the metric we use to assess the performance of the agents. It is a real number between 0 and 1, with 1 indicating the robot navigated from start to goal location using the most optimal path. See [1] for more details on SPL. We show below the training curves for three agents: PointNav, SWP and FWP. As evident, the SWP and FWP agents learn faster than the PointNav agent, and

so DRL agents learn faster when provided with a curriculum. While we have considered a fixed, pre-determined curriculum to follow, in the future one can consider a dynamic curriculum where the curriculum changes based on the progress made by the agent to learn the task.

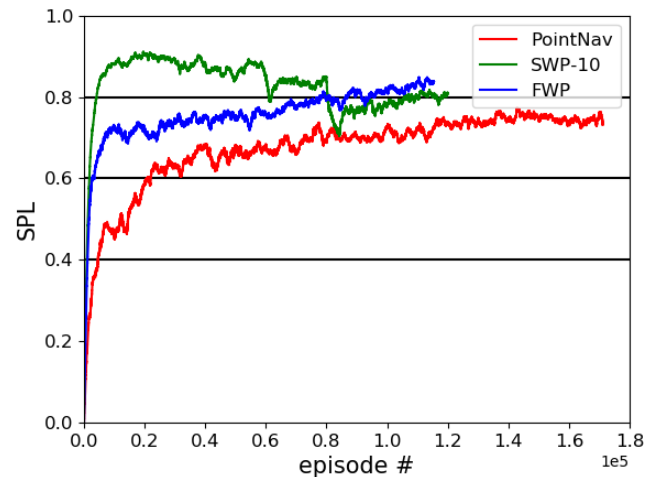


Figure 3: Success-weighted path length (SPL) over successive episodes.

Here are some top-down views showing sample trajectories of the three agents: SWP, FWP and PN (PointNav) for the same start and end locations. The trajectories taken are different and the final SPL, the metric we use to assess performance at test time, is superior for SWP and FWP agents compared to the baseline PN agent.

Test time paths traced by the PointNav, SWP-10 and FWP agents for different episodes. The start and target positions are represented by the green and blue squares respectively. The paths traced are shown in red for PointNav, yellow for SWP and purple for the FWP agents. The SPL values for the three agents for the respective episode are also shown in each sub-figure.

VIII. CONCLUSIONS

Mobile robot autonomous navigation in unknown indoor environments using DRL algorithms was presented and evaluated. The proposed DQL

algorithms are so powerful that there is no need for agents to know anything about the environment since they can still learn how to interact with the environment. The agents can work with any kind of mobile robot without requiring the kinematics and dynamics of the mobile robot because the proposed algorithms are model free. The simulation results show that the proposed autonomous navigation algorithms would allow the mobile robot to navigate autonomously and free of collisions to the target object location in both static and dynamic obstacles without the prior development of an environmental map.

The object detector model would enable the mobile robot to recognize the target object in real time. A practical approach is proposed for collision avoidance and goal-oriented navigation tasks of mobile robots using DQN and DDQN agents. From the experiment, it has been seen that the DQN and DDQN agents trained in the Gazebo simulator can be deployed directly to the real mobile robot without tuning the parameters. The DDQN agent is more robust and better than the DQN agent in exploring the environment, avoiding collisions, and reaching the target object location from the simulation experiment in Gazebo. Thus, only the DDQN agent is used for real-world experiments in an unknown indoor environment.

The real-world experiment is performed in two different environments to demonstrate that the deployed DDQN policy without a learning algorithm is capable of operating in the real world. In conclusion, the proposed method has great potential for autonomous mobile robot navigation compared with SLAM, however, the reward design is the challenging part of DRL for autonomous mobile robot navigation. In the real-world experiment, the final model configuration will have a single board computer (NVIDIA Jetson TX2), which has excellent processing speed with a hex-core CPU, a 256-core

NVIDIA pascal GPU, and 8G LPDDR4 RAM, instead of a laptop as a controller unit.

IX. REFERENCES

1. Mustafa, K.A.A.; Botteghi, N.; Sirmacek, B.; Poel, M.; Stramigioli, S. Towards Continuous Control for Mobile Robot Navigation: A Reinforcement Learning and Slam Based Approach. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **2019**, *XLII-2/W13*, 857–863.
2. Truong, X.T.; Ngo, T.D. Toward Socially Aware Robot Navigation in Dynamic and Crowded Environments: A Proactive Social Motion Model. *IEEE Trans. Autom. Sci. Eng.* **2017**, *14*, 1743–1760.
3. Tai, L.; Paolo, G.; Liu, M. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, Canada, 24–28 September 2017; pp. 31–36.
4. Kalashnikov, D.; Irpan, A.; Pastor, P.; Ibarz, J.; Herzog, A.; Jang, E.; Quillen, D.; Holly, E.; Kalakrishnan, M.; Vanhoucke, V.; et al. QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation. *arXiv* **2018**, arXiv:1806.10293.
5. Andrychowicz, O.M.; Baker, B.; Chociej, M.; Józefowicz, R.; McGrew, B.; Pachocki, J.; Petron, A.; Plappert, M.; Powell, G.; Ray, A.; et al. Learning dexterous in-hand manipulation. *Int. J. Robot. Res.* **2019**, *39*, 3–20.
6. Rusu, A.A.; Večerík, M.; Rothörl, T.; Heess, N.; Pascanu, R.; Hadsell, R. Sim-to-Real Robot Learning from Pixels with Progressive Nets. In Proceedings of the 1st Annual Conference on Robot Learning, Proceedings of Machine Learning Research, Mountain View, CA, USA, 13–15 November 2017; pp. 262–270.

7. Heess, N.M.O.; Dhruva, T.; Sriram, S.; Lemmon, J.; Merel, J.; Wayne, G.; Tassa, Y.; Erez, T.; Wang, Z.; Eslami, S.M.A.; et al. Emergence of Locomotion Behaviours in Rich Environments. *arXiv* **2017**, arXiv:1707.02286.
8. Peng, X.B.; Berseth, G.; Yin, K.; Panne, M.V.D. DeepLoco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Trans. Graph.* **2017**, *36*, 41.
9. Shalev-Shwartz, S.; Shammah, S.; Shashua, A.J.A. Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving. *arXiv* **2016**, arXiv:1610.03295.
10. Sallab, A.E.; Abdou, M.; Perot, E.; Yogamani, S.K.J.A. Deep Reinforcement Learning framework for Autonomous Driving. *arXiv* **2017**, arXiv:1704.02532.
11. Qiu, X.; Wan, K.; Li, F. Autonomous Robot Navigation in Dynamic Environment Using Deep Reinforcement Learning. In Proceedings of the 2019 IEEE 2nd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE), Shenyang, China, 22–24 November 2019; pp. 338–342.
12. Zhu, Y.; Mottaghi, R.; Kolve, E.; Lim, J.J.; Gupta, A.K.; Fei-Fei, L.; Farhadi, A. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Downtown Core, Singapore, 29 May–3 June 2017; pp. 3357–3364.

Cite this article as :

Pooja Upadhyay, Dr. Bappaditya Jana, "Efficient Robot Curriculum Approach for RGBD Indoor Robot Navigation Using DRL Method", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 10 Issue 3, pp. 1187-1192, May-June 2023.
Journal URL : <https://ijsrst.com/IJSRST52310598>