BLINC: Querying the Internal States of Navigation Based Deep-Reinforcement Learning Models

Shurjo Banerjee*, Vikas Dhiman*, Brent Griffin & Jason J. Corso *
The Electrical Engineering and Computer Science Deparment
The University of Michigan
Ann Arbor, MI 48109, USA
{shurjo,dhiman,griffb,jjcorso}@umich.edu

Abstract

Deep reinforcement learning (DRL) algorithms have demonstrated strong progress in learning to reach a goal in challenging environments without the traditional requirement of performing SLAM or path-planning.

1 Introduction

Navigation remains a fundamental problems in mobile robotics and artificial intelligence **??**. The problem, traditionally called SLAM (Simulataneous Localization and Mapping), is classically addressed by separating the eventual task of navigation into *exploration* and *exploitation*. In the exploration phase, the environment is incrementally built and represented in some sort of *map* data-structure. In exploitation, this data structure is used for localization and path-planning to find an optimal path to a given destination based on desired optimality criterion. Although there have been many advances in this classical approach **?**, it remains a difficult challenge. *Either mention some examples or cite a paper that highlights the failiures of SLAM or both!*

More recently, end-to-end navigation methods—methods that attempt to solve the navigation problem without breaking it down into the separate parts of localization, mapping and path-planning—have gained traction. With the recent success of Deep Reinforcement Learning (**DRL**) **??**, these end-to-end navigation methods **?????** forego decisions about the details that are required in the intermediate step of map building.

Work by Mirowski et al. showcased agents that learned to navigate textureless environments to find desired goal locations trained on pure monocular vision - a feat that is still quite difficult for state-of-the-art monocular SLAM systems **?**. *Talk about the memory structures used - no need for any explicit path planning, slam or all that nonsense* The potential for simpler yet capable methods is rich on the surface.

Despite this potential and recent successes, state-of-the-art DRL based methods have been confronted with their own set of problems. In line with other Deep-Learning fallacies (*too negative?*), foremost among these is the difficulty in understanding the method limitations or the kind of patterns that these algorithms are understanding. The inherent black-box nature of these methods make them hard to study.

In this work, we attempt to pull back the lid of how these networks appear to be in fact be performing this navigation. We phrase these queries within the context of exploration and exploitation as is traditional in the SLAM world. Our contributions are two-fold:

1. We succesively blind state-of-the art DRL agents to gain a better understanding of these agents to convert short term actions to more long-term path planning.
2. We introduce BLINC - a new training paradigm wherein agents are incentivized to blind themselves during navigation that is applicable to any DRL based method. Extra incentives

*indicates equal contribution

are provided when this blind is chained together in a contigous sense. We showcase how agents trained via BLINC acheive better performance on state-of-the-art methods that is applicable to any state-of-the-art DRL method to improve reward scores.

3. We showcase how BLINC affords ways in which the internal understanding and representations of these agents can be queries. Specifically, upon blindfolding our agents we showcase we studying the amount of long-term path planning they are implicitly able to perform.

2 Related Work

Localization and mapping Robotic localization and mapping for navigation as a problem since the beginning of mobile robotics and sensing. Smith and Cheeseman **?** introduced the idea of propagating spatial uncertainty for robot localization while mapping and Elfes popularized Occupancy Grids **?** for mapping. In the last three decades, the field has exploded with variation of algorithms for different sensors like cameras, laser scanners, sonars, depth sensors, variation in level of detail like topological maps **?** for low level of detail to occupancy grid maps for high detail and variation in environment types like highly textured or non-textured.

All these approaches require huge amount of hand-tuning and design for adapting to different environments and sensor types. The level of detail of maps also needs to be decided before hand irrespective of the application and hence is not optimized for the application at hand.

Deep reinforcement learning Deep reinforcement learning (DRL) came back to the limelight **?** Check whether this citation should be here with Mnih et al. **??** demonstrating that their algorithms outperform humans on Atari games. Subsequently, the DRL algorithms have been extended **?** and applied to various games **?**, simulated platforms **?**, real world robots **?** and more recently to robotic navigation **??**.

The exploration into robotic navigation using deep reinforcement learning is a nascent topic, it has potential to disrupt the fields of simultaneous localization and mapping and path planning. Also, **?** train and test on the same maps which limits our understanding of the generality of the method. In fact, it is very common to train and test on the same environments in reinforcement learning based navigation works **??** with the only variation being in location of goal and starting point. In contrast, **?** do test on random maps but the only decision that the agent has to make is avoid a goal of particular color and seek other color rather than remembering the path to the goal. On similar lines, **?** test their method on unseen maps in VizDoom environment but only vary the maps by unseen texture. In this work, we take the study of these methods significantly farther with a thorough investigation of whether DRL-based agents remember enough information to obviate mapping algorithms or the need to be augmented with mapping algorithms.

3 Approach

4 Experiments

5 Analysis

6 Conclusion

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

References

References