A Neural-Symbolic Approach for Object Navigation

Xiaotian Liu
Queen's University Canada
Kingston, ON, Canada
liu.x@queensu.ca

Christian Muise
Queen's University Canada
Kingston, ON, Canada
christian.muise@queensu.ca

Abstract

Object navigation refers to the task of discovering and locating objects in an unknown environment. End-to-end deep learning methods struggle at this task due to sparse rewards. In this work, we propose a simple neural-symbolic approach for object navigation in the AI2-THOR environment. Our method takes raw RGB images as input and uses a spatial memory graph as memory to store object and location information. The architecture consists of both a convolutional neural network for object detection and a spatial graph to represent the environment. By having a discrete graph representation of the environment, the agent can directly use search or planning algorithms as high-level reasoning engines. Model performance is evaluated on both task completion rate and steps required to reach target objects. Empirical results demonstrate that our approach can achieve performance close to the optimal. Our work builds a foundation for a neural-symbolic approach that can reason via unstructured visual cues.

1. Introduction

Object navigation refers to the task of finding specific scenes or objects in an unknown environment [2]. Solving the object navigation problem is necessary for building intelligent systems that can autonomously conduct tasks in any given environment. Object navigation is a natural ability possessed most animals, but it is not a trivial problem for artificial agents to solve [1]. The challenges of object navigation include achieving sub-goals, such as object recognition [2] and scene memorization [2]. An agent needs to navigate, discover objects, and memorize scenes while keeping track of its location. Recent advances in deep reinforcement learning have given rise to end-to-end RL agents that can conduct semantic navigation tasks using raw RGB images [10, 9]. However, these end-to-end systems require large neural networks with a significant amount of training steps. Additionally, the policies learned by these systems are distributed among its weights, which make transfer learning difficult [11]. In contrast, symbolic approaches, such as automated planning, can easily solve most object navigation tasks given a discrete environment.[8] The problem with symbolic approaches is that they requirement explicit defined symbols to represent environmental information.

We approach object navigation with a neural-symbolic method; taking raw RGB images as input and constructing a spatial memory graph containing both location and object information. The learned graph can be searched using any graph search algorithm. We demonstrate that our neural-symbolic approach is very sample efficient, and the object navigation is almost trivial once the graph is constructed after some exploration. This work also builds the foundation for applying neural-symbolic methods to complex long-horizon tasks for embodied agent reasoning.

2. Approach

To tackle the object navigation problem, our agent needs both a visual perception module and a memory unit. The proposed method has two major components: an object detection neural network and a spatial graph constructor. We chose to use YOLO [7] as our object detection method for its speed and simplicity. The graph is constructed using the Networkx library [4]. The nodes of the graph are the location and orientation of the agent, and the edges represent the possible movements. We use bidirected graphs to distinguish movement in opposite directions. Our agent uses object detection to gather object information from raw RGB images, and stores this information in the spatial graph as a simple lookup. With both modules, our agent can navigate to any object that has previously been discovered.

2.1. Object Detection

Our object detection method is based on YOLO [7]. It takes a raw image as input and outputs detected objects distribution with bounding boxes. Compared to two-step methods, such as RCNN [5], it is faster to train and easier to deploy. We artificially generated a dataset containing our test objects with random configurations in AI2-THOR. To ensure generalizability, the training object configurations are

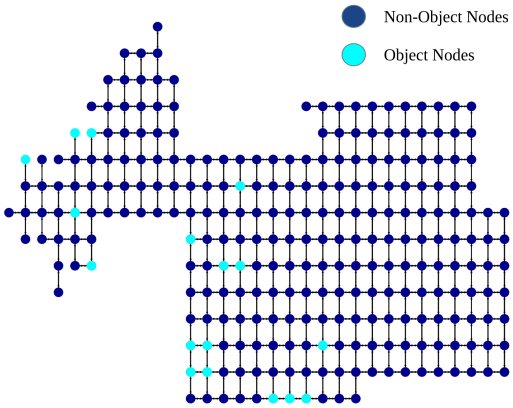

Figure 1. Example of a Learned Spatial Memory Graph

not present in the graph construction phase.

2.2. Spatial Graph Generation

The spatial graph serves as the "memory" of the agent during exploration. We encode location as the key for each node and visual observations as values. The edges are actions taken by an agent. The four actions we included are MOVEAHEAD, MOVEBACK, TURNLEFT and TURNRIGHT. The movement actions have a step size of 0.25 while the turning actions have a turn angle of 90°. We gave the agent a budget of 1000 steps to explore. For each observation, the probability of whether an object is present is calculated using YOLO. To make exploration more efficient, we implemented a queue to store locations visited. We also adopted a breadth search algorithm to facilitate exploration. The learned graph can be searched using any graph based algorithm for finding the optimal path. We used Dijkstra's algorithm for path planning as the default method [3]. We traversed through five environments to generate spatial graphs. An example of a generated graph is shown in Figure 1. On average, each graph has around 1231 nodes. However, only around 20 nodes have reachable objects. We setup a random walk agent to compare with our neural-symbolic agent. This random walk agent has equal probability of choosing each of the four actions.

3. Experiments

We use the AI2-THOR environment [6] to evaluate both training and testing environments. Our experiments include three object categories in a virtual kitchen environment. We choose seven different random location configuration for each object. Five of the configurations are used to conduct our experiments while the rest are used for training the YOLO classifier. We evaluated our approach using two criteria: the success rate and the average path length.

	Average Length	Success Rate
NS Agent	23	100%
Random Walk	2328	23.8%
Optimal	19	100%

Table 1. Summary of Object Navigation Results

We define success rate as the percentage of successful runs where our agent is able to find the target. The average path length is defined as the number of actions our agent needs to find a particular object. For baseline, we compared our agent against a random walk agent with uniform distribution among available actions. The experiments are repeated five times, and the average is calculated.

4. Results

Table 1 summarizes the results of our overall object navigation task. Our approach is able to conduct the object navigation task perfectly in all five settings. Our test results on the path length also confirm the effectiveness of our neural-symbolic method. The average length is 23 actions, which is slightly more than the 19 optimal lengths. Our method is reaching the optimal theatrical limit, which would require significantly more data and training steps in end-to-end RL approaches. Comparing our results to random walk, we can see a drastic improvement. An agent randomly searching in an environment with a given movement budget of 2000 can find an object 23.8% of the time. The average length for Random Walk agent is 2328, which takes 101 times more actions on average than our agent. Our results demonstrate a great deal of promise in using neural-symbolic approaches for the embodied agent setting.

5. Conclusion and Future Work

We outlined a neural-symbolic method for discretized object navigation tasks in the AI2-THOR environment. We adopted a YOLO-based object recognition network and integrated it with a spatial graph memory. The results demonstrate that our method can perform near-perfect object navigation tasks in a simple kitchen environment. Our neural-symbolic method is comparable with the optimal solution, and is significantly better than a random walk baseline. We demonstrated the effectiveness of a hybrid approach and have shown the importance of integrating neural networks with a symbolic reasoning engine.

One future direction is to solve more complex tasks that require higher level reasoning. Tasks such as rearranging items in a room, cooking according to a recipe are currently challenging problems for end-to-end neural networks [8]. We hypothesize that by converting a visual environment to a symbolic representation, we can perform the high-level cognitive tasks using optimized logical reasoning systems.

References

[1] Ramón Barber, Jonathan Crespo, Clara Gómez, Alejandra C. Hernámdez, and Marina Galli. Mobile robot navigation in indoor environments: Geometric, topological, and semantic navigation. In Efren Gorrostieta Hurtado, editor, *Applications of Mobile Robots*, chapter 5. IntechOpen, Rijeka, 2019. 1

[2] Jonathan Crespo, Jose Carlos Castillo, Oscar Martinez Mozos, and Ramon Barber. Semantic information for robot navigation: A surveyr. *Applied Sciences*, 10(2):497, 2020. 1

[3] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269, 1959. 2

[4] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. *https://www.osti.gov/biblio/960616*, 2008. 1

[5] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. *ICCV*, 2017. 1

[6] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, and Abhinav Gupta andAli Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017. 2

[7] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1

[8] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. *arXiv preprint arXiv:1912.01734*, 2019. 1, 2

[9] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. *ICLR*, 2019. 1

[10] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. *ICRA*, 2017. 1

[11] Zhuangdi Zhu and Kaixiang Lin andand Jiayu Zhou. Transfer learning in deep reinforcement learning: A survey. *arXiv preprint arXiv:2009.07888*, 2020. 1