Cognitive Mapping for Object Searching in Indoor Scenes

by

Shibin Zheng

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved November 2019 by the
Graduate Supervisory Committee:

Yezhou Yang, Chair
Wenlong Zhang
Yi Ren

ARIZONA STATE UNIVERSITY

December 2019

ABSTRACT

Visual navigation is a multi-disciplinary field across computer vision, machine learning and robotics. It is of great significance in both research and industrial applications. An intelligent agent with visual navigation ability will be capable of performing the following tasks: actively explore in environments, distinguish and localize a requested target and approach the target following acquired strategies. Despite a variety of advances in mobile robotics, enabling an autonomous with above-mentioned abilities is still a challenging and complex task. However, the solution to the task is very likely to accelerate the landing of assistive robots.

Reinforcement learning is a method that trains autonomous robot based on rewarding desired behaviors to help it obtain an action policy that maximizes rewards while the robot interacting with the environment. Through trial and error, an agent learns sophisticated and skillful strategies to handle complex tasks in the environment. Inspired by navigation procedures of human beings that when navigating through environments, humans reason about accessible spaces and geometry of the environment a lot based on first-person view, figure out the destination and then ease over, this work develops a model that maps from pixels to actions and inherently estimate the target as well as the free-space map. The model has three major constituents: (i) a cognitive mapper that maps the topologic free-space map from first-person view images, (ii) a target recognition network that locates a desired object and (iii) an action policy deep reinforcement learning network. Further, a planner model with cascade architecture based on multi-scale semantic top-down occupancy map input is proposed.

DEDICATION

This is dedicated to my parents, Benjun Zheng and Yun Xie, who support me, believe in me and inspire me at every footprint along the way.

ACKNOWLEDGMENTS

The very first gratefulness is given to Dr. Yezhou Yang, thanks to his long-term guidance and support. I feel blessed that I chose to join Active Perception Group (APG) led by Dr. Yang and feel fortunate that Dr. Yang agreed to serve as my advisor. I admit that I am not a student that is diligent enough to focus on a specific field and always try to learn as much as possible yet neglect the importance of concentration. However, Dr. Yang never losses patience on me and keeps encourage me along the way of my graduate studies. Secondly, I would like to appreciate my thanks to Xin Ye, who is a senior PhD student in the APG group. Along the way of collaborations, she instructed my studies and experiments, I was able to learn a lot from her and her assiduousness edified me deeply.

I would also want to express my thanks to Duo Lv, who is a senior PhD student who collaborates with our group, he always raises new ideas and possesses a broad view of researches which influence me a lot. I would like to thank Anshul Rai, Sree Gowtham Josyula Jacob Fang, Zhe Wang, Aadhavan Sadasivam, Tejas Gokhale, Mohammad Farhadi, Varun Chandra Jammula, Fengyu Yan, Changhoon Kim, Kausic Gunasekar and Rudra Saha from our group for the incredible experience along the path. Meanwhile, many thanks to the collaborators from Trident One project in the Luminosity Lab including Chase Adams, Tyler Smith, Sathish Kumar Katukuri, Trevor Lucero, Mitch Brown and John Patterson. I spent a fulfilling internship in the project and learnt a lot. Further, many thanks to Chenlu Sun for her long-term company. and Chenwei Zheng, Hongzhi Zhu, Jianwei Zhang, Siyu Zhou, Han Yu for their encourages.

In the end, I would like to thank my father and my mother, for everything.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

Vision-based navigation is one of the areas of research that remain vibrant for a long time. It intersperses a wide range of fields and is of great value for research and development. Apart from using other sensor modalities, vision becomes increasingly prevalent in autonomous mobile robots. Navigation can be defined as the procedure of ascertaining current position in the environment, planning on account of the environment and the target location and following the planned route.

As in most modernized factories and warehouses, autonomous mobile robots are playing important roles for assisting with factory works, e.g. Automated Guided Vehicles navigates in the warehouse transporting and reallocating cargos (Ullrich 2015). Even though the prevalence of industrial robots and the great efficiency they come with, the behaviors of the robots are factitiously programmed. Robots simply follow instructions to maneuver. A truly intelligent robot is not considered.

To develop a truly intelligent robot with vision-based navigation capability, it is inevitable to enable the robot to autonomously detect the target from vision input, sense about the environment and figure out how to approach the target based on acquired strategies. Inarguably, having autonomous vision-based navigation ability is one of the most fundamental requirements for robots to enter human life and assist with daily work. It is of great importance to solve the problem. An assistive robot is able to bring great convenience and release human's hands from rigid and repetitive drudgeries. Rescue

robot can search for survivors from a disastrous environment, or and elderly-care robot can provide assistance like fetching a stuff that the elder may find hard to reach.

With the emergency of reinforcement learning (RL) method, there is a high chance of speeding up the maturity of the product. Reinforcement learning is a method that trains agent in the environment and through trial and error, the agent gradually learns the action policy that optimize some objectives given in the form of maximizing the accumulative rewards while interacting with the environment, similar to a biological agent. By taking advantage of the method, the robot is able to learn the strategies to autonomously take actions in the environment.

In recent years, With the recent increase of computational capabilities, deep learning has been developing rapidly, machine learning experience remarkable improvement when learning from complex data. by taking advantage of this, deep reinforcement learning (DRL) combines both RL and deep learning to promote the evolution of a wide range of intricate decision-making tasks further that were previously out of reach for a machine. With the power of DRL, agent is able to learn mapping directly from complex state-space representations, e.g. image pixels, to actions.

1.2 Challenge

Despite the great advances in visual navigation research field, and unignorable progress RL methods have brought, the major difficulty of the approach is its generalizability. After trained in a batch of training environments, the performance in related yet unseen environments or pursuing new targets is the evaluation of generalization capability. The RL methods usually suffer from unsatisfiable

generalizability when transferring the previously learned model to finding a new target or interacting with a novel environment. It is impractical to train the model every single time in a new environment and especially in the real-world scenarios, the environment differs dramatically.

1.3 Motivation

There are many researches were trying to tackle the poor generalizability of the RL methods. As in the reinforcement learning, the agent is exposed to extremely complex state-space, simply deepening the neural networks works poorly and it is inefficient. However, it was shown by (Jaderberg, Mnih et al. 2016) that, by aggrandizing and forcing the RL agent to learn some auxiliary tasks, the performance in the sense of not only generalizability but also many other metrics improves noteworthily. In another word, the auxiliary tasks enhance the data-efficiency by training the agent to learn more general state representations.

It sets us to think, what could be a more general state representation? As humans, when we navigate through some environments, previous experiments were draw in similar conditions, estimate accessible areas which is the topology of the environment, searching for the desired location and then follow common sense of environment layouts or some time heuristics for moving over the desired location.

As an analogy of the human's behavior while navigating, we wander if while training an agent for visual navigation, augmenting with learning free-space map of the environment and target recognition as auxiliary tasks for the agent will boost the performance or not. The goal of this work is to design such a learning framework that

maps action policy from first-person view images while jointly learning the free-space map and target location representations and improve the performance of the network by demonstrating experiments in both virtual indoor environments and real-world scenario dataset and comparisons with previous researches under different metrics.

1.4 Contributions and Outline

In this work, we mainly develop neural network models that map directly from pixels to actions while intermediately learning the free-space map and object semantic segmentations. The model has both trained and test in virtual environments and real-world scenes.

In **Chapter 2**, we give a concise historical review on prior research on vision-based navigation.

In **Chapter 3**, we present a joint neural network architecture that has cognitive mapping and object semantic segmentations for visual navigation in indoor scenes. The methodologies behind the auxiliary tasks of mapping the free-space and target locations representation learning from purely images are demonstrated in detail. The applied deep RL method is also presented. Experimental setting is illustrated, quantitative results and performance analysis are continued at the end.

In **Chapter 4**, we present a novel idea of cognitive semantic mapping that maps from the first-person view to top-down semantic map. In the semantic map, both geometric information and semantic information are provided. We argue that it is an adequate and condensed modality as a state representation for a wheel-based robot as it encodes all the necessary information for navigation. Insisting on the principle idea of "a robot with

vision that finds object", we present a neural network model that maps from pixels to actions while jointly mapping the first-person view to a surrounding top-down semantic map.

In **Chapter 5**, we end this work with conclusion and give a short discussion about the future work.

CHAPTER 2

BACKGROUND

Visual Navigation has been actively studied during the past four decades. In a previous comprehensive survey (DeSouza and Kak 2002) sorted the conventional approaches for visual navigation problem, and divided the robot navigation into two subjects as indoor navigation and outdoor navigation. The two domains differ a lot mainly due to the accessibility to map and position signal and the illumination conditions. In indoor scenarios, the environments are more likely to be structured, position signal is not directly offered and illumination condition is gentle and even. While in outdoor scenarios, the environments are some time unstructured especially in off-road areas, the position signal is reachable from GPS (Global Position System), yet the light condition more extreme sometime cause failure to visual system.

This Chapter focuses on the methods that try to solve visual navigation in indoor scenes and gives a brief background of RL methods and those applied to solve visual navigation problem.

2.1 Conventional Methods

In a previous comprehensive survey of robot navigation (DeSouza and Kak 2002), the Indoor navigation is further divided into three sub-divisions, which are map-based navigation, map-building navigation and map-less navigation.

In map-based navigation, the global geometric information of the environment is directly provided. The frontier works, e.g. the occupancy grid methods (Borenstein and

Koren 1991, Oriolo, Vendittelli et al. 1995), try to solve the navigation problem by providing the discretized occupancy grid for robot navigation, and especially in (Borenstein and Koren 1989) improved the idea by combining the "virtual force field" where each grid was defined a repulsive force that pulls or repels the robot navigating within. Further, another work (Kim and Nevatia 1994) projected a three-dimensional spatial map of landmarks into 2D map and utilized the projected 2D map for indoor navigation. (Borenstein and Koren 1989) also incorporated uncertainties in the occupancy map that takes the sensor measurement errors into consideration, where the quantization error also contributes. Despite the fact that the global geometric information is directly given for robot to localize itself in the environment and building the map from visual cues is omitted, the occupancy grid methods still intrigues us since it initials the idea of sample the continuous space into grid world and simplifies the navigation problem.

Built upon the ideas of map-based methods, in map-building navigation, the robot navigates in the environment without a cut-and-dried model of the environment, however, builds such geometric model on the fly. The map-building methods release the necessity of possessing the map (or model) of the environment and ease the processes of generating such an environment description beforehand which could be hard especially metrical information is not given. The map-building methods enable autonomous robot to explore the environment and update knowledge of it. (Moravec 1980) made the first attempt to develop a remotely controlled mobile robot which is equipped with a single camera mounted on a 50 cm slider. It takes 9 images along the slider and extras features from the image for updating the 3D information that is projected to 2D occupancy grid, the environment is represented as 3D coordinates derived from the images. As the robot

navigates to a new location, the whole process repeats again for mapping the current surrounding environment. However, this kind of strategies usually suffer from inefficient computation at each mapping iteration. (Thrun 1998) extend the map-building idea further by combining both occupancy grids and topological graphs for indoor navigation. SLAM (Simultaneous localization and mapping) methods also belong to map-building category that generate map of the environment while simultaneously localize in the mapped environment based on the observation from on-board sensors.

In map-less methods for navigation, the movements of robots only depend on the perception information from the environment. In the system of using map-less methods, unlike map-building approaches, there is no need for creating a map before navigation to be operated. (Lucas and Kanade 1981) proposed a map-less method for navigation using optical-flow which estimates the surrounding occupied objects along the movements of the robot. (Matsumoto, Inaba et al. 1996) proposed an appearance-based method for visual navigation by storing a sequence of images for matching and navigation.

2.2 Reinforcement Learning and Deep Reinforcement Learning

Even though the success of abundant research works on tackling the navigation with many evolutional approaches, the adaption of the methods from modeled environments to other environments is always limited. Hence, learning is necessary for robot to operate in different environments and behave properly. For this reason, roboticists have made efforts on developing learning methods for controlling robot. Reinforcement learning is more of a trend to be applied for solving a variety of problems.

The general reinforcement learning setup is illustrated in Figure 2.1. The agent is connected to the environment, at time t the agent observes in state S_t and receives the reward of R_t, based on action policy learnt so far and then takes action A_t in the environment falls into the next state S_{t+1} and obtains the R_{t+1} in the corresponding state. The loop continues until the then terminal state sent by the environment which ends the training episode.

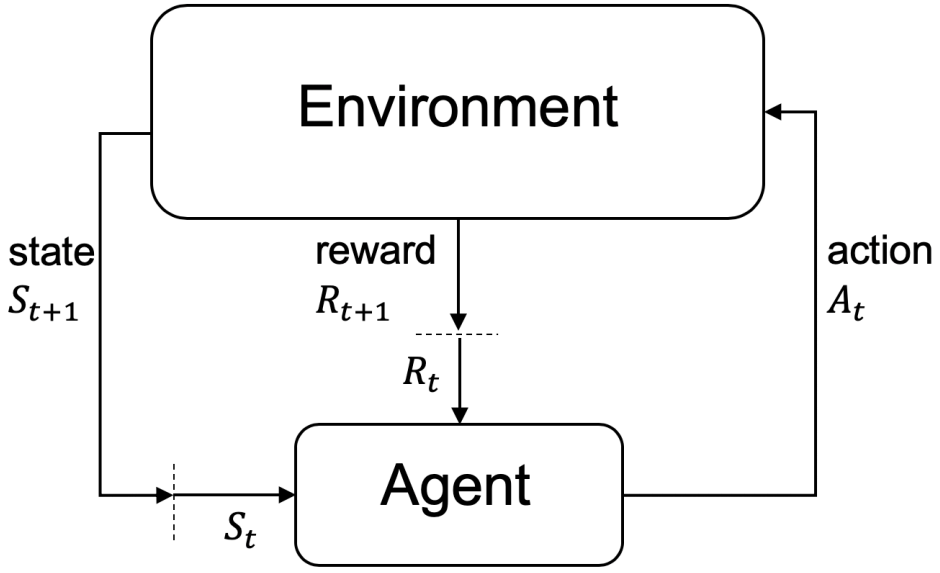


Figure 2.1 Reinforcement Learning illustration

Classical reinforcement learning methods usually build upon the mathematical framework called Markov Decision Process (MDP). MDP helps to model decision making in different states via observations from states. It is first introduced as known in (Bellman 1957) and consists of states S, actions A same as before, and transition probabilities T that indicates the consequence of state transition caused by the actions of robot in the environment. $T(s_{t+1},\ a_t,\ s_t)$ denotes the interactions between the agent and the environment. besides, the next state s_{t+1} only depends on the action a_t taken at the

current time stamp and current state s_t, no additional information is relevant however most of the RL approaches apply different strategies of reward function $R = R(s, a)$ that is dependent on the states and actions.

The goal of RL is to either find or approximate such a policy π^*that maps from observation at current state to the action space while maximizing an objective, which in the RL setting, the expected accumulated reward denoted as V_π^*. The relationship between the RL setup and MDP can be more elaborately explained by introduce the Bellman Equation (Bellman 1957), the equation are shown below:

$$\begin{aligned} V_{\pi^*}(s) &= \max_a Q_{\pi^*}(s, a) \\ &= \max_a \mathbf{E}_{\pi^*}[R_t \mid s_t = s,\ a_t = a] \\ &= \max_a \mathbf{E}_{\pi^*}[r_{t+1} + \textstyle\sum_{k=0}^{N} \gamma^k r_{t+k+2} \mid s_t = s,\ a_t = a] \\ &= \max_a \mathbf{E}_{\pi^*}[r_{t+1} + \gamma V^*(s_{t+1}) \mid s_t = s,\ a_t = a] \\ &= \max_a \textstyle\sum_{s'} P(s' \mid s, a)\left[R_{s',a} + \gamma V_{\pi^*}(s')\right], \forall s \in \delta \end{aligned} \tag{2.1}$$

The V_{π^*} is equal to the max value across all $Q_{\pi^*}(s, a)$ with possible actions at current state, where π^* denotes the optimal policy. Q_{π^*} is the expectation of the accumulated reward if taking action $a_t = a, \forall a \in A$. The accumulated reward is based on rewards accumulated in the future states and times by the discount factor γ, where N can be an infinite number. In the final stage, the $P(s' \mid s, a) = T(s', s, a)$ indicates the transition probabilities that by taking action a, the probabilities of falling into different states s'.

Various applications of RL boost the community. (Kohl and Stone 2004) proposed a policy gradient RL method to optimaize the gait of a quadrupedal robot for forward speed. (Kollar and Roy 2008) used RL approach to optimize the trajectory for robot to

explore an unknown environment and optimize the automatic data collection process. (Kim, Jordan et al. 2004) employed RL method to learn a fitted stochastic and non-linear model that enable the complex control for flight of an autonomous helicopter. (Guenter, Hersch et al. 2007) also utilized RL to enable robot with imitating capability to reproduce constrained human demonstrations. (Kormushev, Calinon et al. 2010) presented a RL approach that enables a robot to obtain adaptive motor skills by learning the correlations across motor parameters.

The problem of robot navigation in the environment is well suited for RL methods that train an agent in the environment to maximize the accumulated rewards while navigating to a desired position. Obstacle avoidance using RL (Michels, Saxena et al. 2005) was proposed that enables an autonomous vehicle utilizing monocular vision only to avoid obstacles while navigating in an outdoor environment. The agent was trained both in a supervised manner and RL. In the work (Duan, Cui et al. 2008), a TS fuzzy network and Q-learning combined approach is proposed to learn the reactive behaviors of robot and solve navigation problem. (Oßwald, Hornung et al. 2010) presented a RL approach to teach a humanoid robot to selectively choose correct navigation actions for reliably reach the goal position as fast as possible, while the trade-off between motion drifting and velocity is also taken into consideration. An variant method so-called quantum-inspired reinforcement learning (QiRL) algorithm is proposed in (Dong, Chen et al. 2010) that adopts a probabilistic action policy and a novel reinforcement method.

Recently, deep learning starts to thrive with its modeling capability, and deep learning approaches also demonstrated some success in navigation tasks, e.g. deep neural network models are trained with optimal state-action pairs to predict actions based on visual input

of an agent (Giusti, Guzzi et al. 2015). A major deficiency of these methods is the need for supervision yet collecting training data for navigation tasks is even more expensive than common regression problems.

However deep reinforcement learning (DRL) combines both deep learning and reinforcement learning which inherits the modeling capability to scale the reinforcement learning to high-dimensional state and action spaces. Remarkable successes haven been brought by the potential of DRL. (Mnih, Kavukcuoglu et al. 2015) kickstarted with a super-human model that outperformances human in Atari 2600 games. The model they applied is deep-Q-network that takes the video game pixels and the game score as inputs. It is a milestone of DRL that the capability of handling high-dimensional raw data is proven. (Silver, Huang et al. 2016) presented AlphaGo that defeated the world champion in Go, the model of AlphaGo is trained in a combination of supervised and reinforced manner. DRL is also naturally applied to robot controls. (Levine, Finn et al. 2016) and (Levine, Pastor et al. 2018) proposed methods that learn control policies directly from visual input from the real-world and collected 2 months for real-world experiments. As RL also generalize to real-world decision-making problems and optimization problems, plenty of other machine learning tasks can be approached by using RL.

2.3 Deep Reinforcement Learning for Visual Navigation

The employments of DRL to visual navigation has improve the approaches further, yet still suffers from low sample efficiency. Learning via trial and error within high-dimensional state space and action spaces, DRL is usually impeded by sparse reward in the environment, as the entire states of the environment is relatively large comparing to

the goal state which is quite unique in the environment sometime. Increasing interests have been provoked in object searching problem in indoor scenes, since the application of household robot is of necessity for assisting human with daily lives, and the debut of household robot may reform the current human lives and influence the society further. Comparing to conventional methods, the DRL has a learning manner that may enable the robot to adapt and transfer to most of the new environments so that increase the landing of the product.

To teach the robot with navigation ability. In (Xie, Wang et al.), authors propose an approach using RL for obstacle avoidance. Proposed as the method, first-person view images are sent to a fully convolutional neural network for depth estimation and then passed to a DoubleDQN (Van Hasselt, Guez et al. 2016) network for obstacle avoidance.

Further, in the seminal work (Zhu, Mottaghi et al. 2017), a target-driven deep reinforcement learning methods is proposed. two weight-share ResNet-50 (He, Zhang et al. 2016) are deployed for extracting presentations of both current first-person view image and the target-view image at the target state, then A3C deep reinforcement learning network is deployed for action policy learning. The reason to take the target as an input is to enable the generalizability to different targets and scenes. Results showed generalization abilities across both new targets and new scenes. (Chen, Moorthy et al.) extended the methods by several increments and helped the model to generalize better and achieve shorter path for navigation.

In the work (Ye, Lin et al. 2018), the task is extended from finding a specific target scene in the environment to finding a specific target object in the environment. Same as approach in (Zhu, Mottaghi et al. 2017), the proposed model takes an image of target

object as input for generalization ability purpose. The generalization ability across both new scenes and new targets is also proven. Also, a novel decaying reward function that improves the performance than regular intuitive reward functions. In the consecutive work (Ye, Lin et al. 2019), the generalization problem is further studied, by intermediately predicting semantic segmentation and depth information from RGB images, the model exhibits a boosting performance on transferring to both new scenes and new targets. The intuition is straightforward, by forcing the robot with auxiliary tasks and learning more general data representation that is independent from scenes and objects, the model performance better. Also worth to mention, alike in the work (Mousavian, Toshev et al. 2019), different combinations of data modalities including depth information, semantic segmentation and raw RGB images are tested in parallel with different network structures.

CHAPTER 3

COGNITIVE FREE-SPACE MAPPING FOR OBJECT SEARCHING

3.1 Introduction

The task of visual navigation is an intelligent robot that automatically navigates in either outdoor environments or indoor environments. The purpose of the navigation is to reach a desired or pre-defined goal state in the environment. the fundamental skills should be learnt by the robot to localize the target, avoid obstacles and approach the target location. As introduced in Chapter 2, remarkable achievements have been made by many research works. In this chapter we are presenting an approach that tries to tackle the same problem as the robot navigates in indoor scenes searching for targeting objects. In this section, we first give a short introduction about the principle idea about deep reinforcement learning (DRL) and describe the derivation of the proposed idea.

DRL methods become increasingly popular in the robotics field of studies. With the prevalence of deep learning, the DRL methods employ deep neural networks to approximate a policy function π that maps state to action, especially the convolutional neural network can be used to observe the state.

As illustrated in Figure 3.2, the DRL methods try to model the agent with a deep neural network (DNN) which maps the observation O_t in state S_t at time stamp t to the predicted action A_t with on the policy $\pi_\theta(s, a)$ with neural network parameters θ.

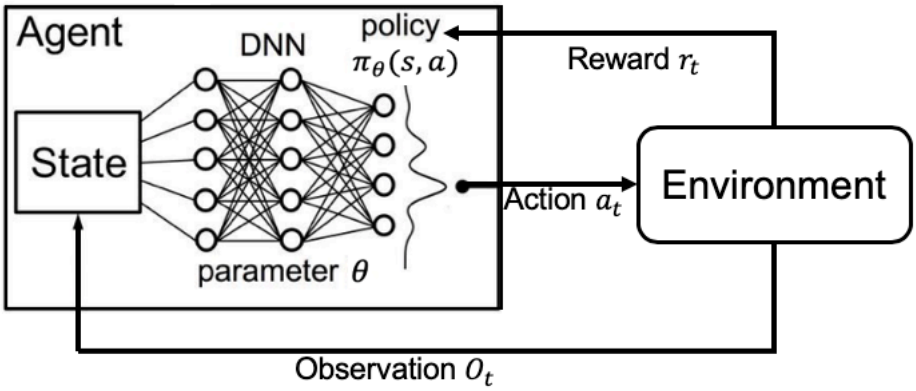


Figure 3.1 Deep Reinforcement Learning illustration

The methodology is well-suited for visual navigation problem: A robot with visual input explores to find a specified target in an environment. We can adopt the problem setting to the reinforcement learning setup with high reward assigned goal state being location of the target object or the closest location to the target object, based on which the agent is able to learn the policy to approach the target location. An illustration of fitting the visual navigation problem to reinforcement learning settings is shown in Figure 3.2.

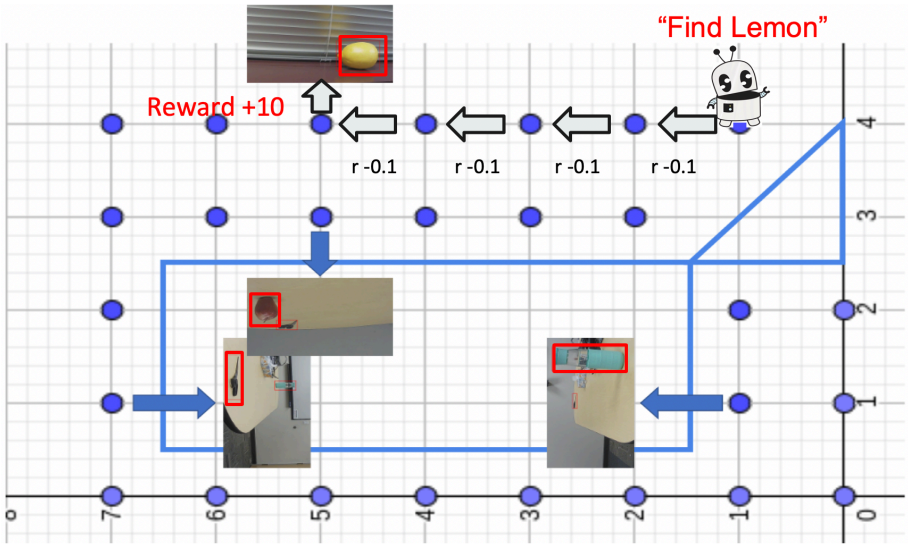


Figure 3.2 RL Setting for Object Searching

With the surge of DRL methods and their potentials of solving navigation problems, there emerges several works that make effort to tackle the problem. Seminal work (Zhu, Mottaghi et al. 2017) proposed a target-driven DRL model that trains the agent to find the target state of the target-driven scene in a environment. Further, the work (Ye, Lin et al. 2018) extended the method to a target object guided DRL model to enable the agent finding a specific object rather than a target scene in the environment.

Despite the success of these methods, the performance is constrained in trained environment however fails to retain in new environments, and re-training process is required for every new scene which is time-consuming and computational expensive. The unsatisfiable generalizability and inefficiency limit these methods especially while being implemented to real-world applications.

The reason behind such behavior is rooted in the training manner, when training in the environment, some high-level of scene dependent features are more easily learnt by the agent rather than general scene representations. However, the approaching ability should be a general ability.

We try to find the inspiration biologically from human navigation mechanism. As human navigating in environments, we reason about the geometry and topology of the environment, locate the target, move around blocking objects and approach the destination. As discovered in (Hafting, Fyhn et al. 2005), the spatial representation system in brains has two special types of neurons that correlate to the occupancy in the space. The two types of neurons are called place cell and grid cell. Place cell functions as a place recognition module that inform about familiar or previously seen places, and grid cell functions as an occupancy recognition module, the objects occupied places stimulate

the corresponding grid cell in the brain, e.g. in front of a person is a table, the table occupies the place in the front, and the corresponding grid cell representing the front fires to inform the brain about the occupancy in the front.

Besides, as the knowledge from multiple view geometry (Hartley and Zisserman 2003), human with binocular vision is able to accurately estimate the depth information nearly up to the end of arms, which is limited by the distance between two eyes. Inspired by the biological fact of human's binocular vision, we suspect that the depth information may not be necessary and as general as occupancy grid maps, the occupancy of the surrounding environment formed by objects that are more distant than an arm is actually understood by the brain as vulgarly speaking as "imagination".

Intrigued by the reasoning of human navigation mechanism, we reason about the two data representations of the environment, which are top-down free space map and semantic segmentation. Reasonably, household robot will appear in houses first as wheel-based robot, the interacting space for the robot is actually the 2D free space. Meanwhile providing the robot with the semantic mask so as to giving a sense of direction. With the geometric information of surrounding and the sense of direction. The robot is ought to learn how to approach the target and maneuver around the obstacles at the same time. An visualization of a robot on the 2D map and seeing a semantic segmentation is shown in the Figure 3.3.

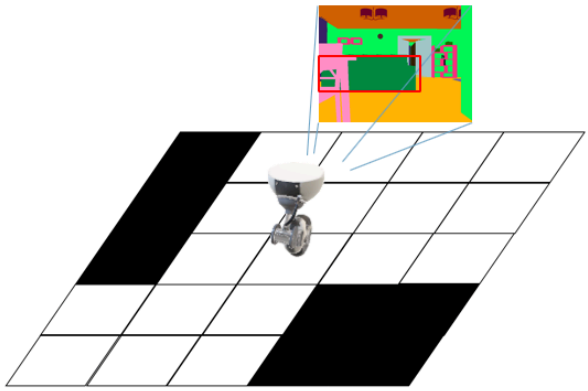


Figure 3.3 Visualization of a Robot on the Observed 2D Free-Space Map and Semantic Segmentation

We make effort on putting forward a novel approach that enables the agent to learn a generalizable approaching policy. We retain the manner of approximating the policy with deep neural network that maps from visual input to actions and keep the training paradigm of DRL. Differs from end-to-end training, we inherently enforce the network to encode the geometric information of the environment and also the target information. We present a method that first explicitly learns the top-down free-space map and masked semantic segmentation as data representations and treat them as input to the DRL network model for action policy learning. To validate the generalizability of the method, we conducted experiment on both virtual indoor-scene simulator: House3D (Wu, Wu et al. 2018) .The experimental results are reported in the Section 3.4.

3.2 Cognitive Mapping with Auto-Encoder Network Structure

In this section, we explain one of the major components proposed as a mapping function that maps from first-person view image to the top-down free-space map. As

explained in the previous Chapter, in the real-world applications, it is more practical to have wheel-based robot for indoor conditions, the robot navigation of the robot actually happens on the 2D map of the environments. We believe that, by encoding the top-down free-space map, which is a more general and condensed representation of the robots' state, in the model, it facilitates the generalizability of the model. As argued in previous chapter, auxiliary tasks help the performance of the reinforcement learning.

Encoder-Decoder (Autoencoder) is a specific neural network structure. The first half of the network maps the input information x to a lower dimension space and generate hidden features h, this part functions as an Encoder, which is formulated as function $h = f(x;\ \theta_{en})$. Then, the other half is called Decoder which restores the original input as close as possible, which is formulated as $\hat{x} = f(h; \theta_{de})$. The original purpose of the network is to reconstruct the input data x by minimizing reconstruction error $\mathcal{L}(x, \hat{x})$, which measure the distance between the reconstructed output with the original data. Figure 3.2 shows the basic structure of an autoencoder.

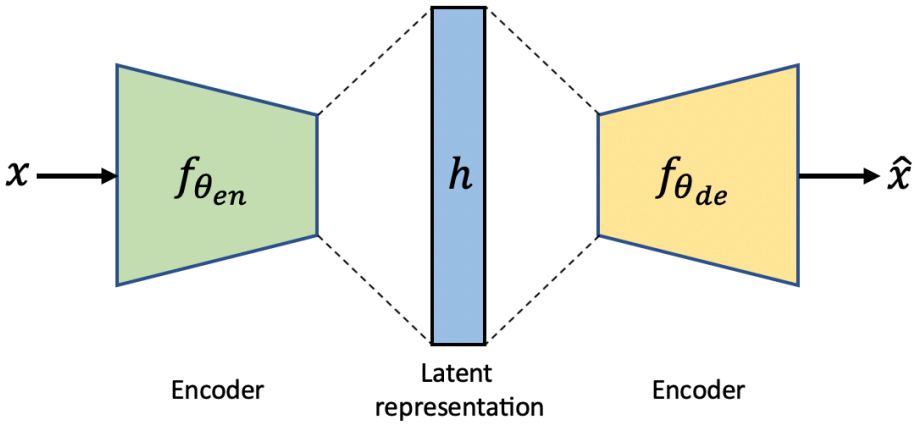


Figure 3.4 Basic Encoder-Decoder Model Architecture

Beyond, the Encoder-Decoder network structure is frequently used to construct other data modalities and is implement in many related computer vision problems such as image segmentation (Ronneberger, Fischer et al. 2015), image restoration (Lehtinen, Munkberg et al. 2018), depth estimation (Alhashim and Wonka 2018) and optical flow estimation (Dosovitskiy, Fischer et al. 2015).

The idea of mapping from RGB images to the top-down free-space map originated from above-mentioned tasks, from a distribution of data modality to another related data distribution. However, unlike the image segmentation, image restoration and depth estimation, in which the data distributes in the same space which is the 2D first-person view image plane projected from 3D spatial information, the mapping function we are proposing actually maps from first-person view images to top-down maps, spatial transformations must be somehow achieved.

Several pioneer works have proposed several network architectures to mapping with a capability of spatial transformation. In the work (Lu, van de Molengraft et al. 2019), the task is to map from first-person image to the semantic occupancy grid map in the front space. It is the very first work that is proposed to solve the occupancy grid mapping. The motivation to tackle such a problem also falls into the importance of occupancy grid map as a local metric map representation, and it is a powerful representation for the trending robotics tasks and also autonomous driving. Dataset the model is performed on is KITTI dataset (Geiger, Lenz et al. 2013). (Pan, Sun et al. 2019) also proposed a model with autoencoder architecture that maps from cross-view images to top-down semantic map. In this work, different data modalities including raw RGB image, semantic segmentation and also depth information are tested as input, number of cross views are also taken into

consideration that affects the performance of the model. An attracting module is introduced in the paper called view transformer that transformed the first-person feature space to another, the increases in the accuracy and intersection over union prove the effect of the module and conjunct with our previous hypothesis.

The cognitive mapper we utilize such Encoder-Decoder network structure for top-down free-space map estimation. The network structure is shown in Figure 3.5.

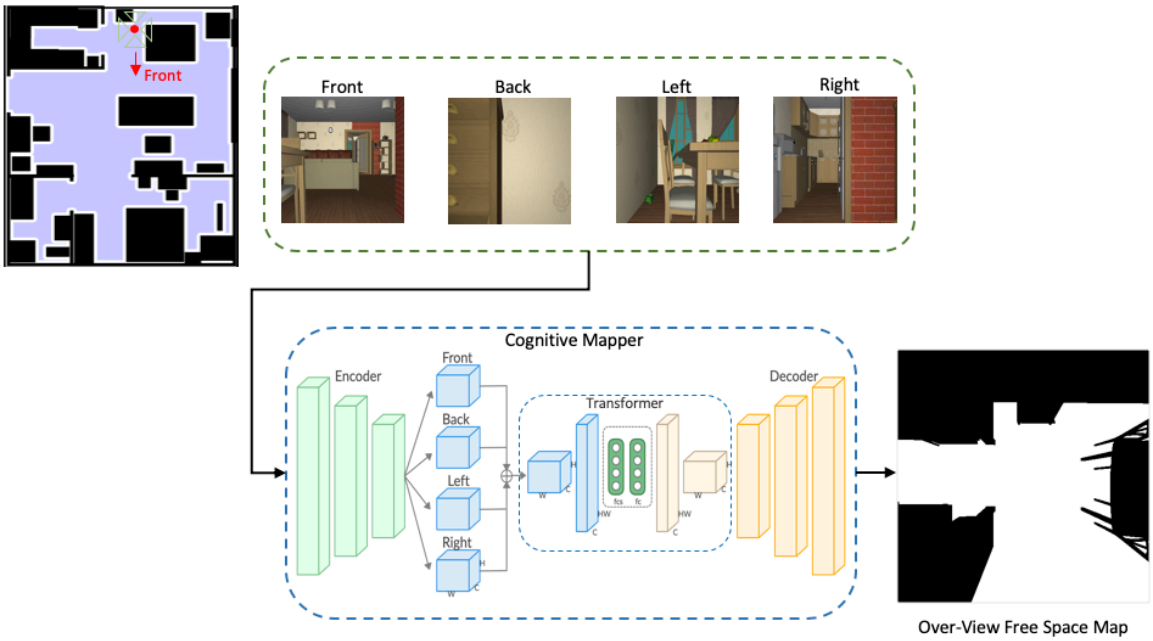


Figure 3.5 Cognitive Mapper with Encoder-Decoder Neural Network Structure

It is built upon the model proposed in (Pan, Sun et al. 2019). At every spatial location on the 2D map, we sample 4 first-person view images from front, back, left and right orientation so as to capture surrounding observations. Then, the first-person view images are encoded by a weighted-shared encoder, which we employ the ResNet-50 (He, Zhang et al. 2016). This Resnet-50 based encoder generates latent feature representation for each view, and we denote the function as $h_i = f(x_i;\ \theta_{en})$ for $i = 1, 2 \ldots, 4$. Then sum up the 4

features into 1 feature map, send to 2 fully connected layers and reshape back to the same size as the feature map. The decoder we employ is a spatial pyramid pooling module (He, Zhang et al. 2015) continued with up-sampling layers. Finally, we decode it to estimate the top-down free-space map.

We form the target and the output of the model to a dual-channel occupancy grid map, where the first channel index denotes the occupied grids and the second the channel index denotes the vacant space, we tailed the model with softmax classifier and apply the cross-entropy loss between the predicted free-space value $\hat{y}$ and the ground-truth free-space map y. As defined in Equation 3.1, the $y^{(i)}$ denotes the i^{th} occupancy grid in the ground-truth free-space map values and the $\hat{y}^{(i)}$ denotes the occupancy grid in the corresponding predicted free-space values.

$$\mathcal{L}(y, \hat{y}) = -\sum_{i=1}^{N} y^{(i)} \log \hat{y}^{(i)} \tag{3.1}$$

The training settings is listed in section 3.4. The predicted occupancy map is later sent as input to the action policy network as a data representation of the environment. Also, the predicted occupancy map is regarded as an inherently learnt model or knowledge of the environment.

3.3 Semantic Segmentation Module

As previous section introduced, many DNNs with variant autoencoder architectures are designed for predicting semantic segmentation (Garcia-Garcia, Orts-Escolano et al. 2017). Fully Convolutional Networks (Long, Shelhamer et al. 2015) is the milestone work that it given the possibility of training the DNN models end-to-end for the semantic

segmentation problem. The neural network architecture discards fully connected layers but uses only convolutional layers. The model significantly improves the performance of semantic segmentations. SegNet is proposed in (Badrinarayanan, Kendall et al. 2017), the decoder of which consists of upsampling and convolution layers, upsampling layer uses corresponding indices from the max-pooling layers in the encoder, then the dense feature maps are sent into convolutional layers, then tailed by a softmax classifier with size of input image and the depth in each pixel is equal to size of semantic category. Plenty of other works including extension of SegNet: Bayesian SegNet (Kendall, Badrinarayanan et al. 2015), (Yu and Koltun 2015) proposed a dilated convolution technique for upsampling, (Pinheiro, Lin et al. 2016) designed a novel skip-connection-like architecture for semantic segmentation and also the current state-of-the-art methods in *DeepLabv*3+ *(Chen, Zhu et al. 2018)*.

For generating the semantic segmentation from the image along with 2D top-down free-space map, we spawn another decoder branch based on *DeepLabv*3+ *(Chen, Zhu et al. 2018)*. The network structure is shown in Figure 3.6. the model also employs spatial pyramid pooling module but in the encoder for multi-scale contextual information.

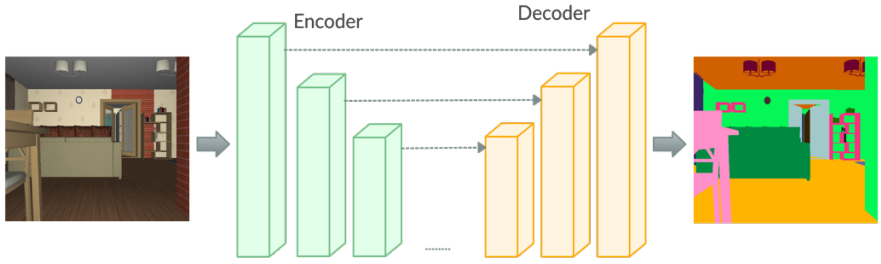


Figure 3.6 Semantic segmentation model of *DeepLabv*3+

The semantic segmentation is also a classification problem, we add a softmax classifier at the end of the network and apply the cross-entropy loss function for training the semantic segmentation model. The function is reformed in Equation 3.2, in which p_i is the one-hot vector of ground-truth semantic label with the index of the vector indicating the label at each pixel i, and ${p_i}^*$ denotes the predicted label probabilities over all semantic labels.

$$\mathcal{L}(p, p^*) = -\sum_{i=1}^{N} p \log {p_i}^* \tag{3.2}$$

Later the predicted semantic segmentation output is post-processed according to the target information by artificially add an attention mask over the corresponding semantic segmentation of target object. The attention mask is another representation besides the occupancy grid map of the agent's observations, we then input the generated representations to the policy network.

3.4 Action Policy Learning with A3C Network

There exists plenty of DRL methods, Deep Q Network (DQN) is the kickstart of DRL model that combines both DNN and RL and has performant results on playing video games, the approach is developed and proposed in (Mnih, Kavukcuoglu et al. 2015). Later a Double DQN is proposed in (Van Hasselt, Guez et al. 2016), by decoupling the estimation, Double DQN improves the stability and hence improves the performance.

The Asynchronous Advanced Actor-Critic Neural Network, also abbreviated as A3C (Jaderberg, Mnih et al. 2016) is a policy-based methods that has multi-thread of actor-critic (AC) networks (Konda and Tsitsiklis 2000). Deep neural networks with AC setup

combines benefits from both value-based models and policy gradient. It estimates a value function $V^{\pi}(s)$ (indicates how good a state s is to be in), and a policy function $\pi(s, a;\ \theta)$ (outputs action probabilities). Both can be tailed at the end of the network as fully connected layers.

Another important feature it possesses is the advantage value:

$$\mathcal{A}_{\theta}(s) = \ Q(s, a) - \mathcal{V}_{\theta}(s) \tag{3.3}$$

where $Q(s, a)$ can be approximate from discounted rewards:

$$R = \ -\sum_{i=1}^{N} \gamma^{i-1} r_i \tag{3.4}$$

The optimal value function $\mathcal{V}^{*}(s)$ is the expected rewards. The inside of using advantage is that it describes how good the action a is compared to the estimated value function based on the policy π. Intuitively, this allow the model to focus more on states where the model's fitting and predictions that deviate the most.

As shown in the Figure 3.7, The A3C network as multiple AC networks interacting with different environment. The advantage of this method is that it enables multiple agents to train in different environments and update the network asynchronously, however, increase the stability and generalization across different environments and targets and hence improve the performance. The multiple agent and asynchronous update setup enables the network to generalize in multiple environments and converges better than a single run.

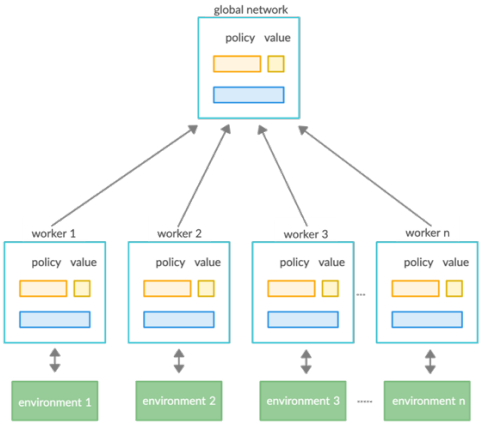


Figure 3.7 Asynchronous Advanced Actor-Critic Neural Network

We employ this DRL algorithm as the action policy network. As shown in Figure 3.8 the overall policy learning network structure, by taking in both predicted semantic segmentations and the generated top-down free-space map, the network learns action policy based on the representations of the environment.

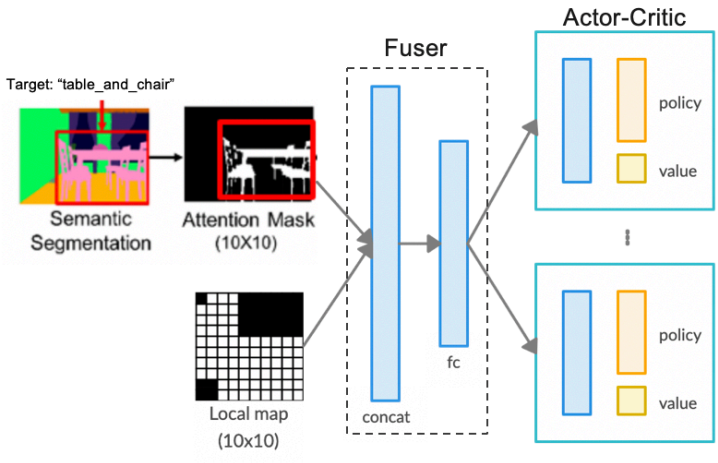


Figure 3.8 Action Policy Network Structure

State space: By following the principle idea that robot with only vision that finds object, the first-person view images are the only sources of information, which we denote as $I(s)$, where s indicates the state. Neither the agent location nor the target location is provided with to the agent. As the proposed idea, we enforce the model to first estimate the top-down free space map, annotated as $m(s) = \mathcal{M}\big(I(s)\big) = \mathcal{M}_{\theta^{(1)}}(s)$, where $\theta^{(1)}$ denotes the parameters of mapper model; secondly the semantic segmentations, and further generate the attention mask of the target object by taking both the semantic segmentation and the semantic label $(s) = \mathcal{S}\big(I(s)\big) = \mathcal{S}_{\theta^{(2)}}(s)$, where $\theta^{(2)}$ denotes the parameters of semantic segmentation network. In this way, the inherent state can also be denoted as the data representation from the observation of the environment as $\mathcal{O}(s) = \mathcal{O}\big(\mathcal{M}_{\theta^{(1)}}(s), \mathcal{S}_{\theta^{(2)}}(s)\big) = \mathcal{O}_{\theta^{(1)},\theta^{(2)}}(s)$. And the action policy and value function can be annotated as $\pi_{\bar{\theta}}(\mathrm{s}, \mathrm{a}) = \pi\big(\mathcal{O}(s);\ \theta^{(3)}\big) = \pi\big(\mathrm{s}, \mathrm{a};\ \theta^{(1)}, \theta^{(2)}, \theta^{(3)}\big)$ and $\mathcal{V}\big(s;\ \theta^{(1)}, \theta^{(2)}, \theta^{(4)}\big)$, where $\theta^{(3)}$ and $\theta^{(4)}$ are two separate branch at the end of the network. As for the **goal states** in the environment, we set them as the states have the first five largest attention masks.

Action space: To separate the problem from the physical control and probabilistic robotics, we discretize the continuous space into absolute grid spaces. The agent's transition from one grid to another is determinate and absolute without any noises and drift. In this way, the action space is also discretized into limited space which are namely the "move forward", "move backward", "rotate left", "rotate right", "shift leftward" and "shift rightward". The 6 actions enable the basic navigation ability of the agent in the environment. In the virtual environment, the actual moving distance or say the size of

each grid is fixed to 0.2 meters and the actual rotating angle is fixed as 90 degrees. All the movements are deterministic and noise-free.

Reward function: We employ the reward function proposed in (Ye, Lin et al. 2018). The reward obtained is positively related to the size of attention mask. However, the reward is only given only if the current attention mask is larger than any of the attention masks previously observed in a training episode. Otherwise, the reward is set to zero.

we set the r_t being the reward gained at time step t and a_t to be the size of attention mask observed at time step t. $r_t = \kappa a_t$, where κ is a non-negative variable and k is a positive constant, the value of κ follow the rules illustrated in Equation 3.5.

$$\kappa = \begin{cases} 0, & a_t \leq \max(a_i) \\ k, & a_t > \max(a_i) \end{cases} for\ i = 1, \dots, t-1 \tag{3.5}$$

The formulation of the accumulated reward function is given in Equation 3.6.

$$\mathcal{R} = \gamma^{i_1} r_{i_1} + \gamma^{i_2} r_{i_2} + \dots + \gamma^{i_t} r_{i_t} \tag{3.6}$$

Since the reward is positively correlated to the size of the attention mask, and reward is obtained by the agent only if the attention mask is increasing, so each accumulated reward follows the constrain $r_{i_1} < r_{i_2} < \cdots < r_{i_t}$ $(i_1 < i_2 < \cdots < i_t)$.

Loss Function: The A3C network combines both policy gradient method and value-base model. Once the worker experiences adequate iterations, we calculate the advantage and accumulated discounted reward according to Equation 3.3 and 3.4, along with the output of the model, i.e. action probabilities $\pi_{\bar{\theta}}(\mathrm{s}, \mathrm{a})$ and value $\mathcal{V}(s)$. The value loss function is given in Equation 3.7, where i indicates the i^{th} iteration:

$$\mathcal{L}^v = \sum_{i=1}^{n} \left(\mathcal{R}^i - \mathcal{V}^i(s) \right)^2 \tag{3.7}$$

The policy loss function is also given in Equation 3.8:

$$\mathcal{L}^p = -\sum_{i=1}^{n} \log \pi_{\overline{\theta}}(\mathrm{s})\, \mathcal{A}_\theta(s) \tag{3.8}$$

In addition to both value loss and policy loss, we also apply a penalty to the model by additional entropy item $\mathcal{L}^e$, shown in Equation 3.9

$$\mathcal{L}^e = \mathcal{H}(s) = -\sum_{i=1}^{n} \pi_{\overline{\theta}}(\mathrm{s}) \log \pi_{\overline{\theta}}(\mathrm{s}) \tag{3.9}$$

The effect of adding an entropy for the policy is to regularize the spread of action probabilities and more deterministic action policy will be learnt, which means at each iteration, the entropy loss encourages single action with a relatively much higher probability. The total loss is equal to the weighted summation of all loss functions, shown in Equation 3.9:

$$\mathcal{L} = \alpha \mathcal{L}^v + \mathcal{L}^p + \beta \mathcal{L}^e \tag{3.10}$$

Where α and β are hyper-parameters served as weights.

3.5 Experiments

3.5.1 Datasets

The dataset we are using is the House3D (Wu, Wu et al. 2018). It is a rich, extensible and efficient environment that contains 45,622 human-designed 3D scenes of visually realistic houses, ranging from single-room studios to multi-room houses, equipped with a diverse set of 94 fully labeled 3D objects, abundant sets of textures and human-designed scene layouts, and the House3D dataset is developed upon and based on the SUNCG dataset (Song, Yu et al. 2017).

3.5.2 Experiment Settings

Cognitive Mapper: 440 scenes selected from the House3D dataset, and further split the dataset into training set which contains 378 houses and validation set that contains 62 houses. The training set has nearly 203k pairs of data and the validation set has nearly 38k pairs of data. We discretize the room by 0.1m intervals and sample the data pairs only at moveable locations where, the center of the top-down view is always free-space. Besides, we collect the data pairs of input with 4 cross-view images in the front, back, left and right differ with 90 degrees and output with 1 top-view semantic image taken at 2.5m above the ground and laid with mask over the vacant space.

Semantic Segmentation Module: we collect 56k data pairs from 100 houses for training and shrink the semantic channel from original 94 labels to 77 labels with the rest being classified as "background".

Action Policy: we collected a set of 248 simulated houses that doesn't overlap with above dataset. Besides, to avoid ambiguity, we collect the objects that only have one instance in a house as the target. As explained in previous section, the state space are each 4 orientations at each sampled location in the house, the actions are determined as moving 0.2 meters or rotating 90 degrees every time. Besides, the collision doesn't account for a failed termination, the robot may get stuck in a state until the maximum steps are reached. Furthermore, the training process is ended manually by research observation, since the divergence across different workers in different environment where some targets are hard to reach and the branch of network converge slower than all others. Weights in Equation 3.10 are set to $\alpha = 0.5$ and $\beta = 0.01$.

3.5.3 Experiment Results and Analysis

The **Cognitive mapper** achieves 0.652 in Mean Intersection Over Union (mean IOU) and 85.7% in Pixel Accuracy. Qualitative results are shown in Figure 3.9.

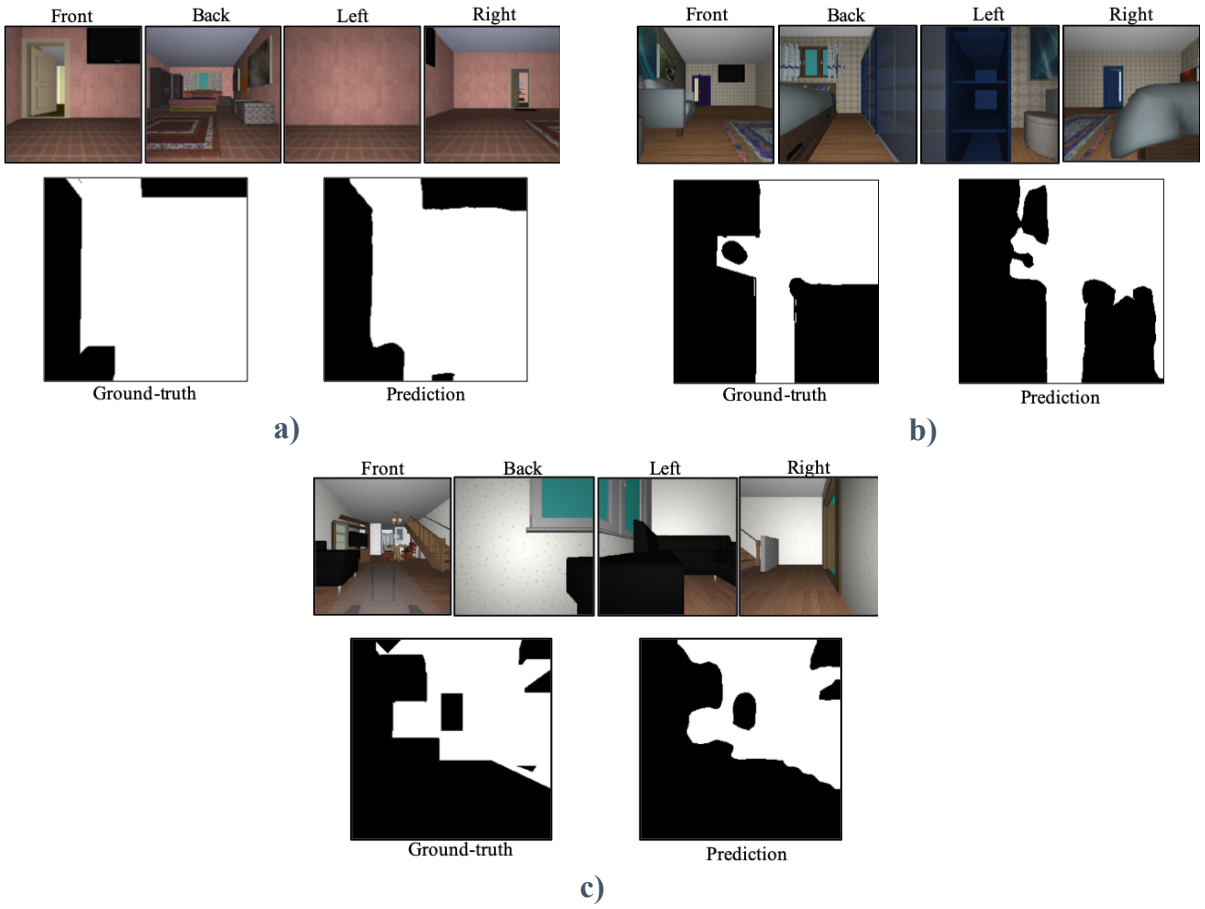


Figure 3.9 Qualitative Results of Cognitive Mapper

Even though the prediction is noisy, based on the free-space prediction, the performance of action policy network drops comparing to policy trained with ground-truth free-space map but still competitive with baselines.

The **Semantic segmentation module** achieves 0.436 in mean IOU. Qualitative results are shown in Figure 3.10. Since we shrink the channels of category, the walls and floors are set to "background" with color of white.

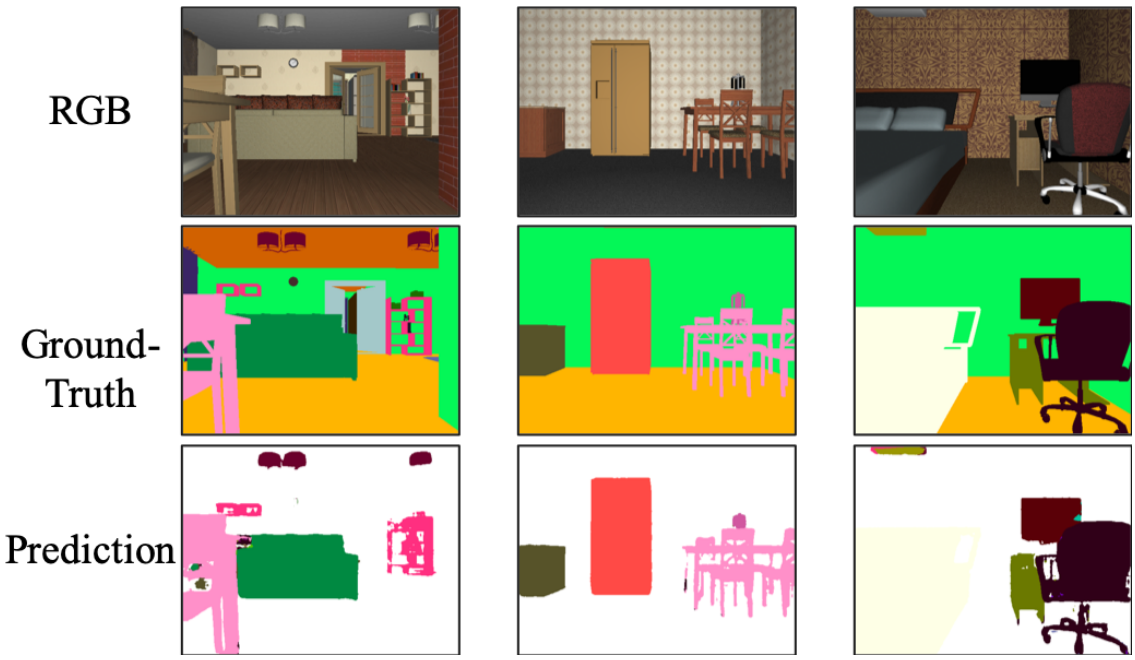


Figure 3.10 Qualitative Results of Semantic Segmentation Module

To demonstrate the generalizability of the model across both target objects and indoor environments, we maintain the experiment setting and compare the proposed method with the following baselines.

Random: at each state, the virtual agent in the house randomly choose an action. The termination is either the agent reaching the target state, or exceeding the maximum step length, which in the case is 1000 steps.

AOP (Ye, Lin et al. 2018): as introduced in Chapter 2, the approach deploys ResNet-50 to extract feature from both observation at current state and the target object,

concatenates both features and a binary attention mask of the target, and sends to A3C network for action policy learning.

GAPLE (Ye, Lin et al. 2019): the approach deploys two autoencoder networks for generating both semantic segmentation and depth as inherent data representation of the environments and send both to A3C network. However, since the predictions of two autoencoder network may be noisy and inaccurate, to prove the general representation that leading to generalization ability we conduct experiments with both predicted and ground-truth semantic segmentation and depth as input to action policy network.

Our Method: as described in this Chapter, the model only takes RGB images as input and inherently learns the top-down free-space maps information and semantic segmentation as a general representation of the environment for action policy network. Like previous method, we also conducted experiments with both predicted and ground-truth top-down free-space maps and semantic segmentation.

We train and evaluate all methods with two setups for proving the generalization ability across both target objects and environments.

Setting 1): all approaches are trained in 1 environment finding 6 different targets.

Setting 2): all approaches are trained in 4 different houses and finding multiple new target objects.

Table 3.1 shows the results of Setting 1). It reports the success rate of different approaches denoting the generalizability across different targets. By comparing the results of c) with results of d) and e) with f), and the success rate of GAPLE method is higher than our methods, however, the drops from testing on the trained objects to new objects of our method are less sharp than which of GAPLE method. Besides, the much

higher success rate of GALPE method on trained objects could denote that the training overfits to training targets which means the GAPLE approach may have data representations that are less general.

Table 3-1 Successful Rates of Baselines and Proposed Approach within Multiple Times of Minimal Steps under the Setting **1**. ("n x ms" Stands for n Times of Minimal Steps, a) Random Method, b) AOP, c) GAPLE Model with predicted Semantic Segmentation and Depth, d) Our Method with predicted Semantic Segmentation and Free-space Map, e) GAPLE with Ground-truth Data Representation and f) Our Method with Ground-truth Data Representation.

Succ. Rate	New Targets [%age]					Trained Targets [%age]				
Minimal Steps*	1 x ms	2 x ms	3 x ms	4 x ms	5 x ms	1 x ms	2 x ms	3 x ms	4 x ms	5 x ms
a)	3.0	5.4	7.2	10.6	12.4	3.5	5.2	8.2	10.5	12.5
b)	24.8	30.0	37.2	44.4	48.2	50.0	80.0	89.0	91.5	93.0
c)	10.7	**25.1**	**44.4**	**47.4**	**51.2**	30.1	**48.3**	**49.9**	51.8	51.8
d)	**14.2**	24.3	39.6	44.5	47.0	**32.5**	40.8	45.4	**52.1**	**55.0**
e)	25.6	**46.2**	**55.2**	**63.6**	**70.2**	49.3	**79.8**	**89.0**	**91.7**	**93.3**
f)	**28.0**	40.1	55.0	59.2	63.5	**51.2**	62.3	70.0	70.0	75.5

Table 3.2 shows the results of Setting 2). It reports the success rate of different approaches denoting the generalizability across different targets. By comparing the results of c) with results of d) and e) with f), the performance of our method significantly outperforms all other methods within different times of minimal steps. The results show that, occupancy grid map could be a better representation of the environment than depth information for a wheel-based robot in indoor scenes. And by comparing the results in trained scenes with the new scenes, the success rate barely drops in our proposed method,

which means that the free-space occupancy map is a more general data representation across different environments.

Table 3-2 Successful Rates of Baselines and Proposed Approach within Multiple Times of Minimal Steps under the Setting **2**. ("n x ms" Stands for n Times of Minimal Steps, a) Random Method, b) AOP, c) GAPLE Model with predicted Semantic Segmentation and Depth, d) Our Method with predicted Semantic Segmentation and Free-space Map, e) GAPLE with Ground-truth Data Representation and f) Our Method with Ground-truth Data Representation.

Succ. Rate	New Scenes [%age]					Trained Scenes [%age]				
Minimal Steps*	1 x ms	2 x ms	3 x ms	4 x ms	5 x ms	1 x ms	2 x ms	3 x ms	4 x ms	5 x ms
a)	9.2	14.8	18.8	21.9	25.6	7.5	12.0	15.5	18.1	20.6
b)	29.6	34.0	37.0	39.5	42.2	30.6	39.6	46.2	49.2	52.0
c)	24.7	32.3	34.3	36.2	36.4	24.2	34.3	36.3	38.1	38.8
d)	**33.0**	**39.9**	**43.4**	**51.6**	**55.3**	**32.2**	**40.0**	**45.4**	**50.6**	**56.5**
e)	29.3	34.0	37.0	39.5	42.2	30.6	39.6	46.2	49.2	52.0
f)	**51.2**	**63.3**	**74.3**	**76.0**	**77.0**	**50.1**	**69.0**	**72.9**	**74.4**	**76.2**

To conclude the Chapter, our proposed model that inherently estimates both free-space map and semantic segmentation as data representation for action policy training improves the generalization ability and performance.

CHAPTER 4

OBJECT SEARCHING USING TOP-DOWN SEMANTIC MAP

4.1 Motivation

As illustrated in Chapter 4, the model maps first from the first-person view to top-down free-space map and utilizes the condensed data representation of the environment and the semantic segmentation of the target for object searching in the environment. The experiment results exhibit a significant increase in the success rate, which represents better learning ability of the model.

However, we are not stopped by the providing the action policy learning network only with top-down free-space map. As shown in Figure 4.1, a comparison between the top-down free-space map and the top-down semantic map. We can easily figure out the probable layout of the house from the right image that on the top-left part might be the kitchen, the center-right part is the living room and the white-colored object might be the bed in a bedroom. While in the left free-space map, it might be difficult to know the object layout of the house.

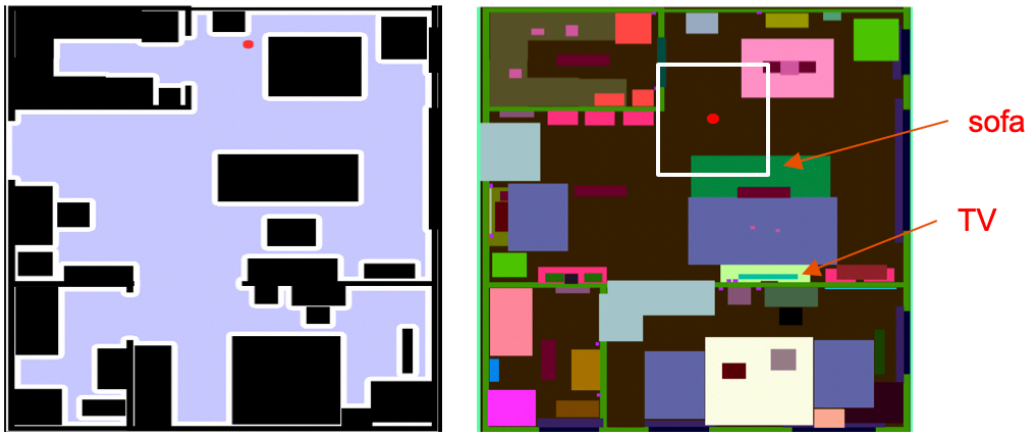


Figure 4.1 A Comparison between the Free-space Map and the Semantic Map

To extend the idea further, we want to not only enable the agent to learn the free-space map of the environment while learning action policy in the environment but also enable it to learn the top-down semantic map of the environment. The intuition behind this is, the semantic map not only gives the occupancy of a gird in the map but also provides with a semantic vector indicating the objects that occupy the gird. In this way, semantic map gives both geometric and semantic information at each grid. Further, the common senses layout in a house is provided. In a human house, objects are placed with certain common-sense rules, e.g. a sofa usually appears in the living room with a coffee table or a television. In another word, the relations between objects in a house might also be an essential cue for the agent to learn while training in the environment.

Besides, if the agent encodes the objects relation in the model, it can achieve a longer range of navigation, e.g. agent is commanded to find a television in the house, it doesn't see television either from the vision input or the top-down semantic map, however a sofa is within observation, via learning, it knows that sofa is commonly placed with a

television and by approaching sofa, it has a higher chance of seeing a television, so that the agent navigates to sofa first as exploration for discovering the television.

We finalize our motivation with two objectives: 1) enable the robot with performant object searching abilities and possibly better performance in both navigation success rate and achieving less steps and 2) The robot learns the common sense of object relations.

4.2 Relative Works

Top-down semantic segmentation is a novel research sub-division in the generative modeling fields. (Pan, Sun et al. 2019) proposed a novel spatial understanding task, so-called cross-view semantic segmentation, the objective of which is to segment the top-down view semantic masks from the first-person view images. (Lu, van de Molengraft et al. 2019) also tries to map from the first-person view images to the top-down semantic occupancy map in the front of the agent. Cognitive mapping and planning for visual navigation method is proposed by (Gupta, Davidson et al. 2017), a joint network that maps from the first-person view image to a top-down free space map and sends in a developed planner for action policy learning. (Chen, Gupta et al. 2019) proposed a learning-based approach that equip a mobile agent with RGBD camera and teach the robot exploration policy, the robot constructs the egocentric occupancy map from down-projected 3D points in depth image and extract features from the constructed egocentric occupancy map and RGB image using ResNet-18 backbone, then fuses the information from both features and passed into an recurrent neural network, which enable the robot with coherent behaviors with memory. This approach also generates the top-down egocentric occupancy map like our previous approach. However, we revisit the principle

idea of “robot with vision that finds object”, the top-down occupancy map is derived from RGB images while they directly generated from depth input.

An intriguing work proposed in (Tamar, Wu et al. 2016), in which a value iteration (VI) module is introduced and emulates the process of Bellman Equation (Todorov 2014) with convolutional computation process. The goal of the work is not only to providing a solution for planning in 2D grid world, but also more importantly using the VI module to automatically learn such a reward map and value map that indicating the rewards and values in each cell in the 2D grid world. The mechanism of the VI module works like attention map for vision tasks such as object recognition and semantic segmentations, it tends to highlight the important parts in the images that contribute and affect thee output the most. Like in the learnt reward map and value map from VI module, they highlight the target cell in the grid world and weaken the occupied cells. More results can be seen in the work (Tamar, Wu et al. 2016).

As previous section discussed, we want to achieve two objectives including both planning and encoding object relations. The problem occurs that how to prove the learnt object relations. Intrigued by the learning ability of VI network, we want also such attention map that highlight not only the cell of target object but also the related objects in both the reward map and the value map. We employ the VI module and develop network that takes in top-down semantic maps and one-hot target indicating the target object in the training iteration, and follow the supervised training manner, we train the agent with optimal actions that leads to shortest path at each state space. Further details about the network and the training procedure are elaborately presented in the following two sections.

4.3 Planner with Value Iteration Module

First, we elaborate on describing the top-down semantic map. The top-down semantic map is illustrated in Figure 4.2. In each occupancy grid, a vector of semantic labels is given and indices of each vector indicate the corresponding semantic category of the object where the category is pre-defined for the semantic map generation. The top-down semantic occupancy map not only encodes the geometric information of the environment but also indicates the objects semantic information, which can be an alternative to both semantic segmentation and top-down free-space map of the previous method as general data representation of the environment.

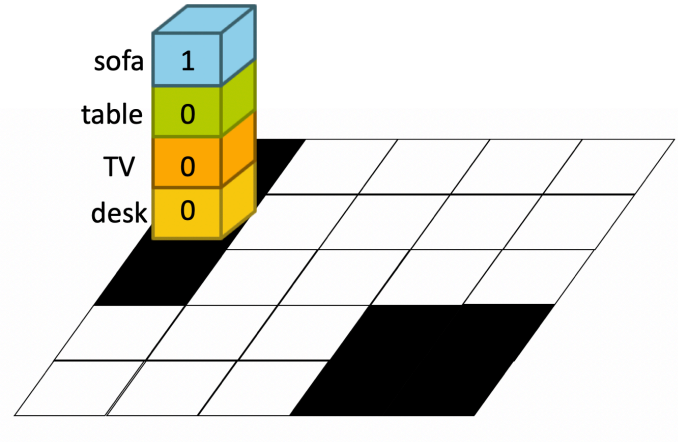


Figure 4.2 An Illustration of the Semantic Map Structure

By following the idea of "robot with vision that finds object", the top-down semantic map can't by directly given as conventional map-based methods. The generation of the top-down semantic map can be achieved in the similar way as the cognitive mapper we previously proposed in Chapter 3. However, in this Chapter we only discuss about the

planner network while omitting the cognitive mapper for top-down semantic map generation, future development of the module is necessary to fulfill such a joint network that remain the vision-only principle. the egocentric top-down semantic maps are manually generated and is input to the model as ground-truth value.

The planner has a cascade structure of multiple scales of map sizes, the very first scale encodes semantic and geometric information about the largest size of real-world map, and consequential scale has the size of half of previous scale. The purpose of having these different scales is to expose the agent with wider observation of the environment so as to enable a long-range planning ability and have a more detailed yet relatively small size of map for better and precise action policy learnings. The network architecture is shown in Figure 4.3. In total the network has three different scales with the first scale module encoding the largest physical space, which is a square of metrical size of 4 meters by 4 meters, second scale module encoding a square of metrical size of 2 meters by 2 meters, and the smallest scale encoding a 1 meter by 1 meter square.

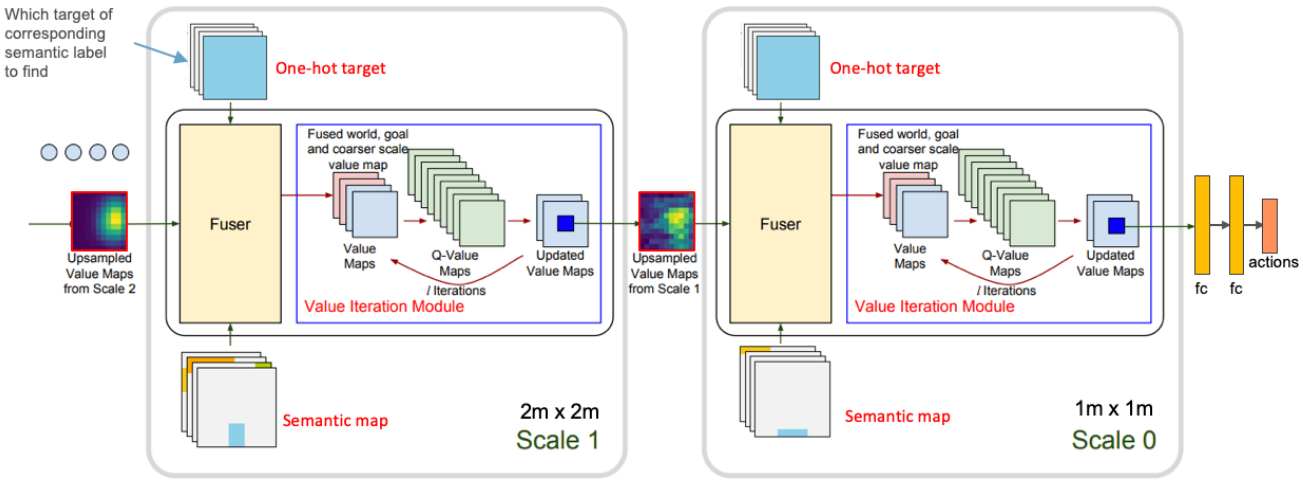


Figure 4.3 The Planner with Cascade Multi-scale Structure

Besides, at each scale module, the semantic map is fused with an one-hot target indicator that defines the target object and upsampled and cropped value map from previous scale, fusion of the triple inputs is than sent to VI module as the inception of reward map. We initialize the first value map at the headmost scale with all zeros, then at the tailed scale, the central-cropped value map is flatten and sent input two fully connected layers for action policy learning, at last, the model is connected to an additional action softmax layer for action probability prediction.

4.4 Experiment

4.4.1 Dataset

The dataset for training and testing the model stays the same with the dataset in Chapter 3, which is the House3D dataset. Keep the same collected houses for action policy training from Chapter 3, we generate dataset from the 248 houses. Differing from the top-down free-space map, we generate top-down semantic occupancy map by projecting all the occupied 3D points with semantic information of objects onto the top-down map. In each cell of the discretized grid world, we form a vector of semantic labels that indicating the objects that occupy the space. To remain the manner of supervised training for the sake of attention map generation, we collect 80k data pairs from 10 houses and 4 targets of top-down semantic occupancy maps at three scales and the corresponding optimal actions in the state which leads to shortest path to the target, to avoid multiple optimal actions in one state, we only choose one from all possible actions in each state.

4.4.2 Experiment Settings

State Space: same as the previous method, the state of the robot is represented as the observation of the environment, which in the case is the egocentric top-down semantic occupancy map at three different scales. We keep the same sampling strategies as the Chapter 3. Where the robot is constrained to perform discrete actions in these virtual environments, i.e., moving 0.2 meters or rotating 90 degrees every time. It also discretizes the environment into a set of reachable locations.

Action Space: with the discretization of locations, the action space is constrained to fixed number of actions. Same as previous methods, the action space includes actions namely “Move Forward”, “Move Backward”, “Shift Left”, “Shift Right”, “Rotate Left” and “Rotate Right”. Each action leads to a consequence of determined moving 0.2 meters or rotating 90 degrees without any drifts and noise.

We first run preliminary experiments to analyze the performance of the proposed model. We conduct experiment with settings of 1) finding multiple objects in a single house and 2) finding multiple objects in multiple houses. Then evaluate the model by testing in both new houses and finding new targets. success rate and success average length will be evaluated. The baseline we initially give is random method.

4.4.3 Experiment Results and Analysis

The hypothesis argues that first the searching performance improves and secondly the object relation can be learnt. The expected signs of learnt object relation is that learnt reward map and value map highlight corresponding targets and also the related objects in the grid world.

Table 4-1 Success Rate and Average Length Results of Methods under Setting 1).

Max step = 500	New Targets		Trained Targets	
Metric	Succ. Rate	Ave. length	Succ. Rate	Ave. length
Random	21.35**%**	470.13	19.71	480.32
Ours	**29.51%**	**338.24**	**100.0**	**34.27**

Under the setting 1) which the model is trained in one house finding multiple targets, the trained model is then tested on both trained targets and new targets also for comparison. By comparing the performance shown in the Table 4.1 between trained targets and new targets, we see that the success rate drops significantly, and the performance of our proposed methods just exceeds the random method. Hence, the generalization ability of the proposed method under the setting 1) is unsatisfying, which also means an overfitting performance to trained targets in the trained house.

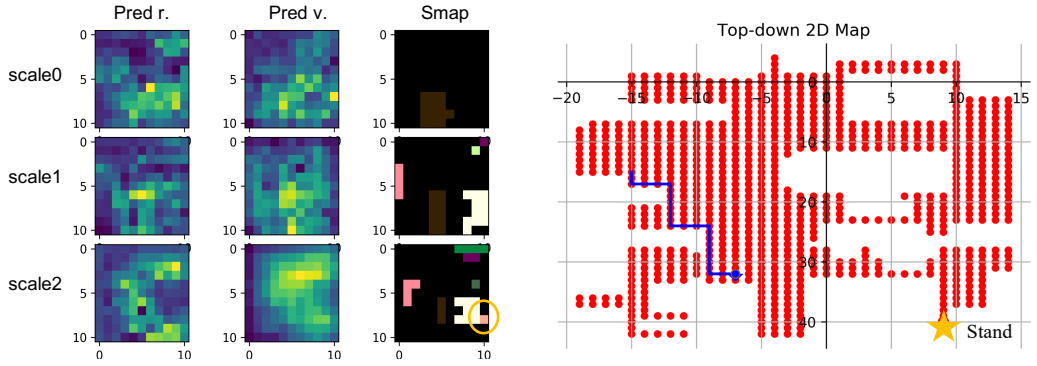
Table 4-2 Success Rate and Average Length Results of Methods under Setting 2).

Max step = 500	New Scenes		Trained Scenes	
Metric	Succ. Rate	Ave. length	Succ. Rate	Ave. length
Random	**25.62%**	463.18	21.46	433.45
Ours	18.75%	**370.18**	**99.0**	**25.58**

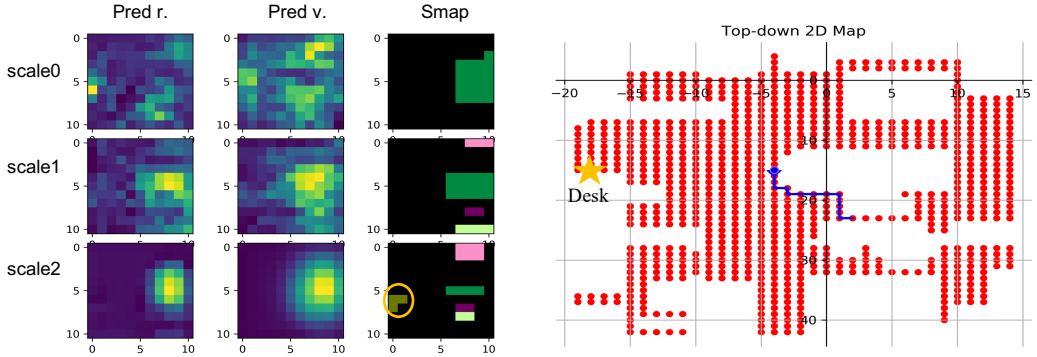
Under setting 2), which is training with multiple houses and multiple targets. The trained model is then tested on both the trained scenes finding same trained targets and new scenes finding trained targets. We draw conclusion that the model still overfits to the

trained scenes with the observation from the testing results in Table 4.2. The success rate of proposed methods is lower than the random method.

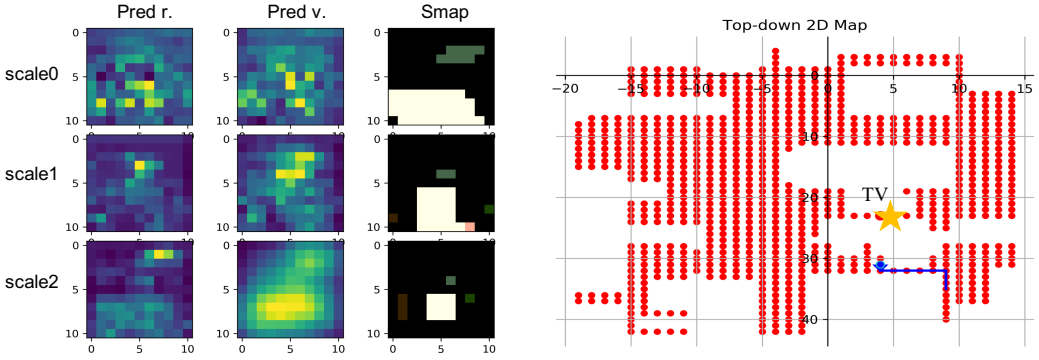
Further, some qualitative results are given and the visualization of learnt reward map and learnt value map are shown in Figure 4.4.



a) Result on Finding "Stand" in the House



b) Result on Finding "Desk" in the House



c) Result on Finding "TV" in the House

Figure 4.4 Learnt Reward Maps and Value Maps at Multiple Scales and the Trajectory

The VI module has shown the ability to learn such reward map and value map that highlights the important part in the occupancy map (Tamar, Wu et al. 2016). We can see from the learnt reward maps and value maps, not only the target location is highlighted but all the objects and especially large objects in the house are highlighted. From both the quantitative results and results of learnt reward maps and value maps, we argue that the method under both experiment settings: **i)** overfits to the geometric information of the trained environment and **ii)** not able to determine from the one-hot target indicator which in the semantic occupancy map is the target.

We analyze from both experimental setting and the model itself the reason why it fails to support our hypothesis. **i)** Through supervised learning, the model tends to learn and memorize the holistic geometric layout of the training houses but not correlate the target to the output actions or the general object relations in the given training house. Besides, another very important factor is that the supervised training unlike the reinforcement learning needs an adequate training set for learning yet the experiments only have limited training and testing sets. Last but not least, the human-designed houses in House3D dataset usually have multifarious layouts and some time provide no common-sense layouts which imply the common object relations. At the first stage, we would like a dataset of more strict layouts that object relations are evident. **ii)** The model's fuser fails to learn what the target is as the one-hot target indicator indicates. As input of VIN is the occupancy grid map with the highest value in target location so that the target location is directly referred. However, in our proposed method, we assume the fuser can figure out the target and estimate the reward map.

CHAPTER 5

CONCLUSION

We presented an approach policy training paradigm, through intermediately estimating the top-down free-space map and semantic segmentation as data representation of the environment for action policy. We tested this model on House3D environment and exhibit a significantly improved generalization capability. Further, a novel approach of using egocentric semantic map as the data representation of the environment is also presented. The method is also tested on House3D environment however yet shows a promising performance and supports our hypothesis. Further development is required.

Besides, the dataset we used is all virtually rendered images. However, at the end of the day, we want to transfer the research work to real-world applications, so that further real-world experiments are necessary to prove the practicability. Further, in the experimental settings, the space is discretized and transition of robot is deterministic while in the real-world, drifts are always introduced by motions of physical robot. Nevertheless, the our proposed approaching method is still potential to be transferred to real-world object searching application, since the model does not require either a deterministic location for representing the state or the deterministic moving distance or rotating angles, it made decisions only based on current observations which are images in our case. To aid the performance of the second method, we need to first increase the dataset for supervised training, and then transfer to a real-world dataset that object relations are stronger and last but not least, elaborate on the modifying the fuser in the network for target recognition on the semantic map.

REFERENCES

Alhashim, I. and P. Wonka (2018). "High Quality Monocular Depth Estimation via Transfer Learning." arXiv preprint arXiv:1812.11941.

Badrinarayanan, V., et al. (2017). "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." IEEE transactions on pattern analysis and machine intelligence **39**(12): 2481-2495.

Bellman, R. (1957). "A Markovian decision process." Journal of mathematics and mechanics: 679-684.

Borenstein, J. and Y. Koren (1989). "Real-time obstacle avoidance for fast mobile robots." IEEE Transactions on systems, Man, and Cybernetics **19**(5): 1179-1187.

Borenstein, J. and Y. Koren (1991). "The vector field histogram-fast obstacle avoidance for mobile robots." IEEE transactions on robotics and automation **7**(3): 278-288.

Chen, L.-C., et al. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. Proceedings of the European conference on computer vision (ECCV).

Chen, L.-H., et al. "Imitating Shortest Paths for Visual Navigation with Trajectory-aware Deep Reinforcement Learning."

Chen, T., et al. (2019). "Learning exploration policies for navigation." arXiv preprint arXiv:1903.01959.

DeSouza, G. N. and A. C. Kak (2002). "Vision for mobile robot navigation: A survey." IEEE transactions on pattern analysis and machine intelligence **24**(2): 237-267.

Dong, D., et al. (2010). "Robust quantum-inspired reinforcement learning for robot navigation." IEEE/ASME transactions on mechatronics **17**(1): 86-97.

Dosovitskiy, A., et al. (2015). Flownet: Learning optical flow with convolutional networks. Proceedings of the IEEE international conference on computer vision.

Duan, Y., et al. (2008). Robot navigation based on fuzzy RL algorithm. International Symposium on Neural Networks, Springer.

Garcia-Garcia, A., et al. (2017). "A review on deep learning techniques applied to semantic segmentation." arXiv preprint arXiv:1704.06857.

Geiger, A., et al. (2013). "Vision meets robotics: The KITTI dataset." The International Journal of Robotics Research **32**(11): 1231-1237.

Giusti, A., et al. (2015). "A machine learning approach to visual perception of forest trails for mobile robots." IEEE Robotics and Automation Letters **1**(2): 661-667.

Guenter, F., et al. (2007). "Reinforcement learning for imitating constrained reaching movements." Advanced Robotics **21**(13): 1521-1544.

Gupta, S., et al. (2017). Cognitive mapping and planning for visual navigation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Hafting, T., et al. (2005). "Microstructure of a spatial map in the entorhinal cortex." Nature **436**(7052): 801.

Hartley, R. and A. Zisserman (2003). Multiple view geometry in computer vision, Cambridge university press.

He, K., et al. (2015). "Spatial pyramid pooling in deep convolutional networks for visual recognition." IEEE transactions on pattern analysis and machine intelligence **37**(9): 1904-1916.

He, K., et al. (2016). Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition.

Jaderberg, M., et al. (2016). "Reinforcement learning with unsupervised auxiliary tasks." arXiv preprint arXiv:1611.05397.

Kendall, A., et al. (2015). "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding." arXiv preprint arXiv:1511.02680.

Kim, D. and R. Nevatia (1994). Representation and computation of the spatial environment for indoor navigation. 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE.

Kim, H. J., et al. (2004). Autonomous helicopter flight via reinforcement learning. Advances in neural information processing systems.

Kohl, N. and P. Stone (2004). Policy gradient reinforcement learning for fast quadrupedal locomotion. IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004, IEEE.

Kollar, T. and N. Roy (2008). "Trajectory optimization using reinforcement learning for map exploration." The International Journal of Robotics Research **27**(2): 175-196.

Konda, V. R. and J. N. Tsitsiklis (2000). Actor-critic algorithms. Advances in neural information processing systems.

Kormushev, P., et al. (2010). Robot motor skill coordination with EM-based reinforcement learning. 2010 IEEE/RSJ international conference on intelligent robots and systems, IEEE.

Lehtinen, J., et al. (2018). "Noise2noise: Learning image restoration without clean data." arXiv preprint arXiv:1803.04189.

Levine, S., et al. (2016). "End-to-end training of deep visuomotor policies." The Journal of Machine Learning Research **17**(1): 1334-1373.

Levine, S., et al. (2018). "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection." The International Journal of Robotics Research **37**(4-5): 421-436.

Long, J., et al. (2015). Fully convolutional networks for semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition.

Lu, C., et al. (2019). "Monocular semantic occupancy grid mapping with convolutional variational encoder–decoder networks." IEEE Robotics and Automation Letters **4**(2): 445-452.

Lucas, B. D. and T. Kanade (1981). "An iterative image registration technique with an application to stereo vision."

Matsumoto, Y., et al. (1996). Visual navigation using view-sequenced route representation. Proceedings of IEEE International conference on Robotics and Automation, IEEE.

Michels, J., et al. (2005). High speed obstacle avoidance using monocular vision and reinforcement learning. Proceedings of the 22nd international conference on Machine learning, ACM.

Mnih, V., et al. (2015). "Human-level control through deep reinforcement learning." Nature **518**(7540): 529.

Moravec, H. P. (1980). Obstacle avoidance and navigation in the real world by a seeing robot rover, Stanford Univ CA Dept of Computer Science.

Mousavian, A., et al. (2019). Visual representations for semantic target driven navigation. 2019 International Conference on Robotics and Automation (ICRA), IEEE.

Oriolo, G., et al. (1995). On-line map building and navigation for autonomous mobile robots. Proceedings of 1995 IEEE International Conference on Robotics and Automation, IEEE.

Oßwald, S., et al. (2010). Learning reliable and efficient navigation with a humanoid. 2010 IEEE International Conference on Robotics and Automation, IEEE.

Pan, B., et al. (2019). "Cross-view Semantic Segmentation for Sensing Surroundings." arXiv preprint arXiv:1906.03560.

Pinheiro, P. O., et al. (2016). Learning to refine object segments. European Conference on Computer Vision, Springer.

Ronneberger, O., et al. (2015). U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention, Springer.

Silver, D., et al. (2016). "Mastering the game of Go with deep neural networks and tree search." Nature **529**(7587): 484.

Song, S., et al. (2017). Semantic scene completion from a single depth image. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Tamar, A., et al. (2016). Value iteration networks. Advances in Neural Information Processing Systems.

Thrun, S. (1998). "Learning metric-topological maps for indoor mobile robot navigation." Artificial Intelligence **99**(1): 21-71.

Todorov, E. (2014). "Markov Decision Processes and Bellman Equations."

Ullrich, G. (2015). The History of Automated Guided Vehicle Systems. Automated Guided Vehicle Systems, Springer**:** 1-14.

Van Hasselt, H., et al. (2016). Deep reinforcement learning with double q-learning. Thirtieth AAAI conference on artificial intelligence.

Wu, Y., et al. (2018). "Building generalizable agents with a realistic and rich 3d environment." arXiv preprint arXiv:1801.02209.

Xie, L., et al. "Towards monocular vision based obstacle avoidance through deep reinforcement learning. arXiv 2017." arXiv preprint arXiv:1706.09829.

Ye, X., et al. (2019). "GAPLE: Generalizable Approaching Policy LEarning for Robotic Object Searching in Indoor Environment." IEEE Robotics and Automation Letters **4**(4): 4003-4010.

Ye, X., et al. (2018). Active object perceiver: Recognition-guided policy learning for object searching on mobile robots. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE.

Yu, F. and V. Koltun (2015). "Multi-scale context aggregation by dilated convolutions." arXiv preprint arXiv:1511.07122.

Zhu, Y., et al. (2017). Target-driven visual navigation in indoor scenes using deep reinforcement learning. 2017 IEEE international conference on robotics and automation (ICRA), IEEE.