Spatially-Distributed Interactive Behaviour Generation for Architecture-Scale Systems Based on Reinforcement Learning

by

Daiwei Lin

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2020

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Daiwei Lin was the sole author of Chapter 1, 2, 4 and 6 which were written under the supervision of Dr. Dana Kulić and were not written for publication.

This thesis consists in part of one manuscript written for publication. Exceptions to sole authorship of material are as follows:

Research presented in Chapter 3 and 5:

This research was conducted at the Royal Ontario Museum by Daiwei Lin and Lingheng Meng under the supervision of Dr. Dana Kulić. Daiwei Lin implemented *Parameterized Learning Agent (PLA)*, while Lingheng Meng implemented *Single Agent Raw Action Space (SARA)* and *Agent Community Raw Action Space (SARA)* and contributed to the data analysis.

Meng, L., Lin, D., Francey, A., Gorbet, R., Beesley, P., & Kulić, D. (2019). Learning to Engage with Interactive Systems: A field Study. arXiv preprint arXiv:1904.06764.

Abstract

This thesis is part of the research activities of the Living Architecture System Group (LASG). LASG develops immersive, interactive art sculptures combining concepts of architecture, art, and electronics which allow occupants to interact with immersively. The primary goal of this research is to investigate the design of effective human-robot interaction behaviours using reinforcement learning. In this thesis, reinforcement learning is used adapt human designed behaviours to maximize occupant engagement.

Algorithms were tested in a simulation environment created using Unity. The system developed by LASG was simulated and simplified human visitor models are designed for the tests. Three adaptive behaviour modes and two exploration methods were compared in the simulated environment. We showed that reinforcement learning algorithms can learn to increase engagement by adapting to visitors' preferences and exploring with parameter noise performed better than action noise because of wider exploration.

A field study was conducted based on the LASG's installation Aegis, Transforming Space exhibition at the Royal Ontario Museum (ROM) from June 2nd to October 8th, 2018. The experiment was conducted in a natural setting where no constraints are imposed on visitors and group interaction is accommodated. Experimental results demonstrated that learning on top of human designed pre-scripted behaviours (PLA) is better at increasing visitors engagement than only using pre-scripted behaviours (PB). Visitor responses to the GodSpeed standardized questionnaire suggested that PLA is more highly rated than PB in terms of Likeability and interactivity.

Acknowledgements

First of all, I would like to thank my supervisor, Dr. Dana Kulić, for her patient guidance throughout my study. I feel very fortunate to have a supervisor from whom I can get countless inspirations and suggestions for my research. I would also like to thank all committee members, Dr. Rob Gorbet, Dr. Philip Beesley and Dr. Mark Crowley. This thesis would not have been possible without their support.

Secondly, I want to give my thanks to Philip Beesley Architecture Inc. (PBAI) and the Living Architecture Systems Group (LASG). Without their outstanding work, I would never have had the chance to conduct the experiment at Royal Ontario Museum. They have provided the physical system, equipment and other resources to help with the experiment.

I would also like to thank my colleague, Lingheng Meng, for all his efforts that made this project possible. He is an excellent researcher, always passionate and optimistic. Having conversations with him has been educational and inspiring. I want to offer special thanks to Adam Francey, for his generous help in bridging our work with the LASG and maintaining operation during the experiment.

I wish to thank all the members of the Adaptive Systems Lab (ASL) for your help and suggestions. Special thanks to Pamela Carreño, Jonathan Lin, Vladimir Joukov, Terry Taewoong Um, Brandon J. DeHart, Kevin Westermann, for all your support and advice.

Finally, I am very grateful for everyone who has provided help and support along the way. Thank all of you for accompanying me through this unforgettable journey.

Dedication

This is dedicated to my family and friends.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Human-Robot Interaction (HRI) addresses the understanding, design, and evaluation of robotic systems for use by or with humans. These robotic systems interact with humans through various forms of communication in different environments. HRI involves both physical and social interactions. The physical HRI focuses on application scenarios where direct contact between humans and robots is crucial. Research work such as cooperative robots[2][3], teleoperation surgery[4] and rehabilitation[5][6][7] consider physical HRI. On the other hand, the social HRI focuses more on the social, emotive, and cognitive aspects of interactions. According to the proximity between humans and robots, HRI can be categorized as remote or proximate[8], and most of the social HRI work belongs to the latter.

Interactive art sculptures designed by the Living Architecture Systems Group (LASG) and Philip Beesley Architecture Inc. (PBAI)[1] provide us with an opportunity to study HRI from the social perspective. Combining concepts in art, architecture and engineering, PBAI creates complex art sculptures at architectural scales which integrate computers, sensors and actuators allowing spectators to have immersive interactions. The aim of LASG's systems is to encourage humans to think differently about their environments. This thesis will be developed upon the Aegis installation at Royal Ontario Museum.

The aim of interactive systems such as the sculptures created by LASG is to engage humans. Generating and measuring human engagement becomes increasingly difficult when the interaction happens in a natural setting with a group of spectators. Responsive behaviours are often manually crafted by human designers, which is relatively easy for simple systems with small scales. However, for a complex system like Transforming Space

[1]PBAI/LASG website: http://philipbeesleyarchitect.com

that has hundreds of actuators and sensors, designing an optimal set of behaviours that maximize engagement is time-consuming and difficult. Therefore, autonomous generation of behaviours could be beneficial. At the same time, interactive systems like those of the LASG which require long-term, continuous interaction, face another challenge. During such extended periods of interaction, the characteristics of the humans interacting with the system and the environment may change by a great extent over time, e.g., viewers can be adults at one time and children at another. Therefore, being able to adapt to the non-stationary interaction environment is important. Additionally, invariable behaviours may become less attractive over time as spectators become more familiar with the system. This requires the system to change its behaviours autonomously over time in order to maintain viewer interest and engagement.

To address these challenges, we introduce reinforcement learning to the interactive system. Reinforcement learning solves problems that can be modelled as Markov Decision Processes (MDP). An MDP consists of a set of states S, a set of actions A, state transition probability T, rewards R and a reward discount factor γ. In reinforcement learning, an agent learns desired behaviours through trial-and-error interactions with its environment. At each time step t, an agent receives an observation of the current state S_t of the environment, and then chooses an action A_t as output. This action is then enacted on the environment, transitioning the environment into a new state S_{t+1}. A reward R_{t+1} is returned to the agent. The interaction diagram is provided in Figure 1.1. A single interaction experience is $(S_t, A_t, R_{t+1}, S_{t+1})$.

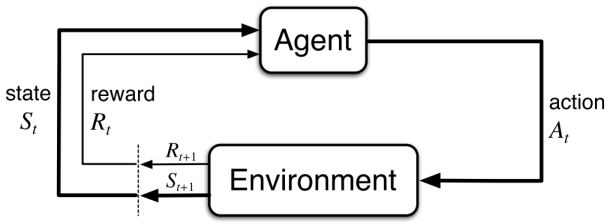


Figure 1.1: The agent-environment interaction.[1]

The interactive behaviour design is sophisticated and often requires knowledge of the visual representations of systems and the context of interactions. Human experts often embed such knowledge in their behaviour designs, which may be beneficial to utilize by the autonomous behaviour generating system. To take advantage of this expert knowledge, instead of automating the composition of interactive behaviours entirely (direct control over all actuators and sensors), we propose to use a reinforcement learning agent that acts on top of human designed interactive behaviours. We decompose and parameterize the key

components of human designed behaviours and let the learning agent learn the parameters for these behaviours. By doing so, we exploit the design of interactive behaviours built based on human expert knowledge, while enabling the system to constantly evolve and adapt to visitor preferences during the long term interaction.

Meanwhile, reinforcement learning is generally studied in static environments such as Atari[9] and OpenAI Gym[10] with stationary rewards and state transition probability. However, as stated above, social HRI proceeds in a natural setting where this static environment assumption does not always hold. Further, interactions in the natural setting happen less frequently and less regularly than in standard RL environments. This leads to sparse rewards for the learning agent and makes learning more difficult. By deploying RL on the living architecture system, we investigate the research problem of applying RL in a non-stationary environment with sparse rewards.

1.1 Contributions

The contributions of this thesis lie in the application of reinforcement learning to interactive environments in natural settings. They are listed in the following:

- A learning algorithm that uses human expert knowledge about interactive behaviours is formulated to bootstrap the learning and reduce the dimensionality of the exploration space.
- Simulations are performed for validation and we show that in a simplified interactive environment, learning agents are capable of adapting to visitor preferences. The ability to learn is affected by different adaptive behaviour generating modes.
- A field experiment was conducted on the Aegis installation together with Lingheng Meng, a PhD candidate and Adam Francey, a Master candidate at the University of Waterloo. In this joint work, Adam implemented a centralized network and prescripted behaviour. Lingheng implemented learning agents in two adaptive behaviour modes, *Single Agent Raw Action Space (SARA)* and *Agent Community Raw Action Space (ACRA)*, and was the lead on the data analysis. I implemented the learning agent in *Parameterized Learning Agent (PLA)* mode and helped with the data analysis.

1.2 Outline

The thesis is organized as follows:

Chapter 2 provides an overview of the related work regarding reinforcement learning algorithms and their applications in human-robot interaction, interactive arts and interactive systems in public spaces.

In Chapter 3, a detailed description of the testbed, Aegis installation, and the proposed method of generating interactive behaviours are given.

In Chapter 4, the design of the simulator and visitor models are described, followed by simulations of learning algorithms. Different exploration methods, adaptive behaviour modes and their relationship with different attraction models are compared in the analysis of simulation results.

In Chapter 5, we firstly explain the experimental and data processing procedures. Then we discuss the metrics used in evaluations. After that, analysis of the visitor engagement level and responses from the user study are presented.

Finally, conclusions and discussion of future work are presented in Chapter 6.

Chapter 2

Related work

In this chapter, we review the related work in reinforcement learning in discrete and continuous action spaces. Then, various application contexts of reinforcement learning in human robot interaction are reviewed. After that, we summarize the relevant works in interactive arts. A summary is provided at the end of the chapter.

2.1 Reinforcement Learning

Reinforcement learning (RL) is a machine learning technique where the learner learns to map situations to actions in order to maximize a numerical reward signal through trial-and-error interactions. RL is generally categorized into two classes, model-based and model-free. Model-based RL learns an environment model representing the state transition probability distribution associated with Markov Decision Process (see Chapter 1). Model-free reinforcement learning does not assume any environment model and learns the policy directly through interactions. Value-based and policy-based RL are the two approaches in model-free reinforcement learning. In value-based RL, the value function $V(s)$ indicates how good the current state s is for the agent. An alternative form of the value function is the Q-function $Q(s, a)$, which represents how good it is for the agent to take action a when in state s. On the other hand, the policy-based RL learns a policy function $\pi(a|s)$, which is a probability distribution $p(a|s)$ representing the chance of taking action a in state s.

State-Action-Reward-State-Action (SARSA) algorithm[1] and Q-learning[11] are examples of value-based approaches, in which the Q function is learned and used for deriving a policy. These two algorithms have been used in many HRI works so far[12][13][14][15][16][17][18].

As for policy-based methods, the REINFORCE algorithm[19] is an instance of using a stochastic policy function consisting of differentiable functions with trainable parameters.

A combination of both value and policy functions has also been proposed, such as Actor-Critic[1]. In Actor-Critic, the *critic* is a state value function evaluating current states and the *actor* is a policy function producing actions. Asynchronous Advantage Actor-Critic (A3C)[20] is an improvement of Actor-Critic which uses multiple actors to collect experiences. In A3C, the critics learn the value function while multiple actors are trained in parallel and synchronize the global parameters from time to time. Here, the actors simultaneously work in different duplicates of one environment. However, in our application, it is impossible to have multiple copies of the interaction environment at the same time.

All works mentioned above are in discrete action space except the REINFORCE algorithm. In our case, the action space of the living architecture system is continuous. For continuous action space, Silver et al. proposed the Deterministic Policy Gradient (DPG) algorithm[21] that learns a continuous deterministic policy function instead. The exploration is realized by adding noise to generated actions.

Deep neural networks allow us to create more complex models as the state and action space grow larger and/or become continuous, which allow the agents to handle more complex situations and generate complex behaviours. In discrete action space, Deep Q-Network (DQN)[22] represents the value function in Q-learning with a deep neural network capable of handling large state spaces. Furthermore, DQN introduces a replay buffer to decorrelate the experience, and a target network to reduce instability of training. In continuous action space, Lillicrap et al. propose Deep Deterministic Policy Gradient (DDPG)[23] to improve learning stability based on DPG, using a replay buffer and target networks for both actor and critic. Other continuous action space RL algorithms include Trust Region Policy Optimization (TRPO)[24] and Proximal Policy Optimization (PPO)[25]. TRPO finds optimal updates within a small region (Trust Region) to improve sample efficiency while maintaining training stability. PPO replaces the Trust Region with a clip function that limits the difference between the new and old policy.

2.2 RL in Human-Robot Interaction

Reinforcement learning algorithms have been used in a variety of human-Robot Interaction contexts and the living architecture system in this thesis is designed for social behaviour interactions. In [26], kinetic movements of feather-like devices are adapted towards user

preferences through an RL framework which aims at raising positive emotions of the participants. In [27], Mitsunaga et al. use a policy gradient reinforcement learning (PGRL) algorithm on a robot to learn user preferences regarding interaction distance, gaze meeting, motion speed and timing. Barraquand et al.[12] create a robot that learns to behave politely in social circumstances. In this study, the reward is directly provided by human users. Chan et al. [28][29] develop a Curiosity-Based Learning Algorithm (CBLA) and apply it to LAS to automatically generate behaviour based on a computational notion of curiosity. The intrinsic motivation of CBLA, based on Intelligent Adaptive Curiosity (IAC)[30][31], allows for continuous evolution of behaviours. These studies were all conducted with one participant at a time, but in our case, we have to handle group interactions.

Various information can be extracted to measure engagement from humans' behaviours that naturally happens in HRI. The simplest way to measure engagement is to use proximity. For instance, Papaioannou et al. combine chat and task-based dialogue using RL to produce a more user-friendly experience[16][15]. The dialog system is loaded onto a Pepper robot that can detect a human's proximity and communicate. The agent learns to switch between chat and task dialogues and receives a penalty when users leave abruptly. Human poses are often captured for analysis of engagement as well. In [32][33], Ritschel et al. apply RL to vary a story-telling robot's level of extroversion. A user's pose is captured by Microsoft Kinect 2 and mapped to predefined pose categories. Then this information is translated into an engagement measure through a directed acyclic graph, which is used as a reward for RL algorithms. Facial expressions are also widely used in identifying humans' engagement levels. Kumagai et al. [26] convert facial expressions to level of delight by analyzing captured videos and calibrating with users' self-rating. In [18], facial expressions and voice are captured and converted into confidence scores of detecting smiles and laughs. Leite et al. [34] use facial expressions and task-related information to train empathic supportive strategies to help children improve chess skills. In [17], RL is used to assist children in learning a second language on a tablet, where facial expression is also used to evaluate the valance of children. A combination of above methods is also used in [27]. Psychological signals are also used to train a robot to adapt to children's preferences[35]. Despite the success of using facial expressions and human poses, these studies are all conducted in a one-to-one manner, where the agent interacts with one participant at a time. Accurately capturing facial and pose information is relatively easy in such cases. For our system that simultaneously deals with multiple visitors in a spatially-distributed public environment with uncontrolled lightning, accurate facial expression and pose analysis is technically challenging and may not be reliable.

In addition to extracting feedback from human behaviours, human feedback can be explicitly used to guide the learning. Thomaz et al. [14][13] present Interactive Rein-

forcement Learning (IRL) for training assistive robots through natural interaction, where a human coachs feedback is used to shape the predefined reward. In their study, the IRL agent learns to complete a cooking task containing multiple sub tasks. A human coach can award a scalar reward signal at any point in the operation of the learning agent. Additionally, at any point, a human trainer can give guidance by specifying an object in the environment and the agent will randomly choose an action that interacts with the specified object. Results show that human teachers tend to use rewards as implications of future actions and the guidance from human teachers leads to efficient exploration and performance improvement. Suay et al. apply IRL in a real-world robotic system[36]. In this study, humans teach a Nao robot to place objects in cups. The experimental result supports findings in [14][13] and further shows that this positive impact of human guidance increases as the state space size grows. In these works, humans are involved as teachers, and the objective of the robot is not necessarily to improve the human experience. In other words, humans are assumed to be expert teachers who are explicitly designing the reward functions. But for the social HRI application in this thesis, the human experience is the primary objective, and spectators may not be experts at designing robot behaviours.

2.3 Interactive Art

According to [37], interactive art is divided into four categories, *Static*, *Dynamic-Passive*, *Dynamic-Interactive* and *Dynamic-Interactive (varying)*. For Static art, art objects do not change and no interaction between the artwork and human observers happens. In contrast, dynamic art objects change over time and according to their evoking mechanism, they are further categorized into three types. A Dynamic-Passive artwork changes its form either with an internal mechanism, or by environmental factors. [38] is an example of such artwork, in which researchers transform the sound of music into multiple frequency bands and output them as tactile vibrations on a matrix of voice coils. The vibration of the voice coils changes as music is being played, and users sense the music through their skins. There is no difference in how Dynamic Passive artworks behave when humans are present or not. However, in the context of HRI, the human is an essential element which decides the responses from interactive systems.

On the other side, Dynamic-Interactive artworks can sense human activities and use these sensations to trigger variations. In Dynamic-Interactive (varying), the extent of interaction is furthered when visitor activities could change original specifications of the art objects. Many Dynamic-Interactive artworks have been created [39][40][41]. In [39], a swarm of agents is used to interact with users. User positions and objects placed in

space are captured by a video camera and processed as input to swarm control. The limitation of this work is that it only responds to a single visitor's position and cannot respond to multiple visitors. [40] creates a musical interface to encourage participants to create their own mixing of sound. The interface contains multiple speakers. Each speaker plays one sound track and participants alter the sound by placing tubes and balls on the top. This interface can be operated by multiple people at the same time and thus allows cooperative composition of sounds. Cubic polyurethane blimps are used as interactive objects in [41]. These cubes are capable of flying around in an enclosed environment and react to users' activity. Although the cubes autonomously move towards visitors when they are detected, no autonomous system is used during the following close-up interaction. Instead, the researchers project their faces onto the cube and communicate with visitors through speakers and microphones remotely.

Previous work by LASG[42][43][44] and the installation in this project are also dynamic-interactive systems. The previous Hylozoic Series interactive art sculpture aims at creating a perception of interacting with a living organism from visitors, using human designed interactive behaviours[42][43]. It is further developed in [28][29] with an intrinsically motivated learning algorithm. The interactive behaviour is autonomously generated by the learning algorithm. Because the learning algorithm uses intrinsic rewards only, it is not designed for adapting to visitor preferences. It is intended to engage visitors by continuously generating different interactive actions. Hylozoic Series were exhibited in a public area without learning system applied[1], and were tested during interaction with a single visitor in a controlled setting[28]. In this work, we use the interactive sculpture developed by LASG in the natural setting and with a reinforcement learning agent for autonomous action generation.

2.4 Interactive System in Public Setting

Social interaction studies also examine interactive systems in the public areas. Scheeff et al. [45] observed how people interact with a creature-like social robot in a science museum. Their study suggests that a rich sensory suite to sense users' locations and facial expressions is helpful.

Works have been done in short-term interactions between the system and humans. The RHINO robot was used to guide visitors in a museum[46]. The robot communicates with visitors using an on-board multi-media interface. It can also play music to entertain

[1]Epiphyte Chamber, Museum of Modern and Contemporary Art, Seoul, South Korea. Link: http://philipbeesleyarchitect.com/sculptures/1312_MMCA_Epiphyte-Chamber/index.php

visitors. MINERVA, an improvement of RHINO, uses memory-based RL[47] to choose the best interactive actions [48]. Even though the guiding robot could encounter multiple visitors at the same time, it typically interacts with only one of them each time. In [49], a mobile robot is placed in the transit area to entertain pedestrians. As the people in transit areas are often determined in walking to their destinations, the interaction time window is small. Others are applied in areas including the urban environment[50], the train station[51] and the shopping mall[52]. The interactions in these studies are not designed to last long. On the contrary, our system is exhibited at a museum for several months and visitors usually have more patience and time to experience and interact with our system. Thus the interaction time is significantly longer.

Other works investigate long-term interactions. Kanda et al. [53] deploy two humanoid robots in classrooms at an elementary school for English education. The field trail lasted 18 days and there was a sharp decrease in the interaction frequency between robots and children at the end of the trial. Even though hundreds of interactive behaviours are created and used in [53], the decrease in the interaction frequency reveals the need for a behaviour generation system that evolves to remain engaging.

2.5 Summary

This chapter provided an overview of related work on reinforcement learning, its application in human-robot interaction, interactive art and interactive systems in public settings. Methods in the discrete and continuous action space were reviewed. The applications in human-robot interaction are organized into different application fields. Categories of interactive arts are discussed and examples of each category are presented. Several studies that apply interactive systems in the public areas are also discussed. In our application, we focus on a dynamic-interactive system with continuous action space that aims for long-term user engagement. The RL algorithm with continuous action space is examined in simulation (Chapter 4) and deployed on the LAS at a museum for social interactions (Chapter 5).

Chapter 3

The Interactive Architecture and Behaviour System

In this chapter[1], we describe the physical system used as the testbed in this thesis, and the design of *Pre-scripted Behaviours (PBs)* that drive the interactive behaviour of the system. The PBs, which are designed by expert architects and interactive system designers, are the baseline we use to compare to the learning systems described in Section 3.2 below. Then we propose an adaptive behaviour mode, *Parameterized Learning Agent (PLA)*, to automatically generate interactive actions.

3.1 Living Architecture Testbed

3.1.1 Physical Living Architecture System

Our testbed is the installation called *Aegis*, as shown in the top-left sub-figure in Figure 3.1, which was exhibited at the Royal Ontario Museum (ROM) in Toronto, Canada from June 2 to October 8, 2018 (https://www.rom.on.ca/en/philip). The installation consists of the Canopy and Sphere, which are both publicly accessible to visitors of the museum.

Since this work mainly used the Canopy part of the installation, we will describe the design of the Canopy in detail, and subsequently refer to the Canopy part of the installation as the Living Architecture System (LAS).

[1]Part of this chapter is adapted from [54]

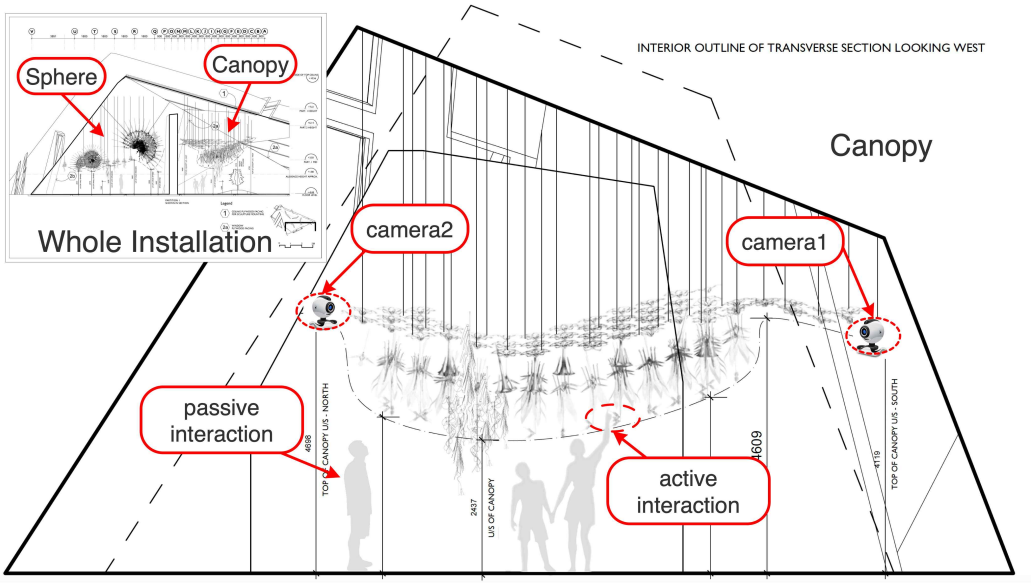


Figure 3.1: Installation Diagram and Interaction Types

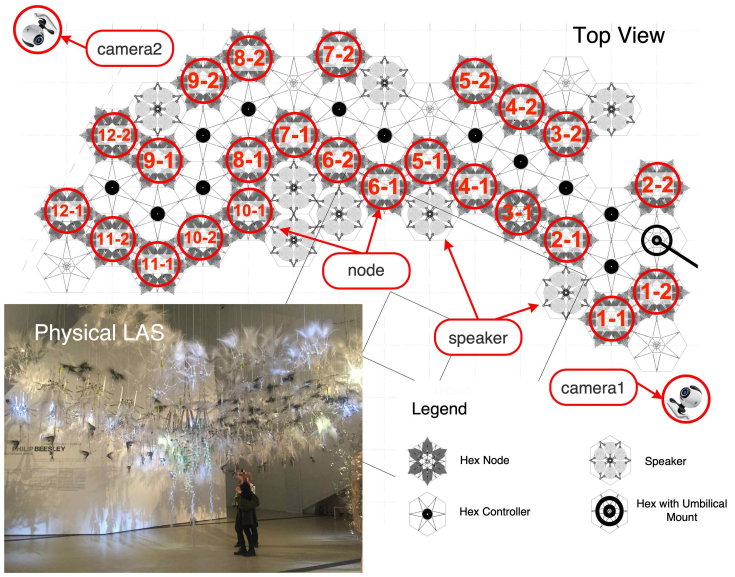


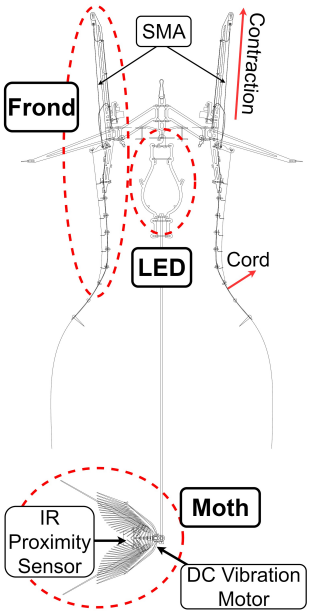
Figure 3.2: LAS: Canopy

The LAS hangs overhead within the Canopy space, with an approximate height of 1.8 meters. Figure 3.1 shows the front view of the LAS. The system is composed of eight speakers and 24 nodes. The arrangement of the speakers and nodes is illustrated in Figure 3.2, where the 24 nodes are highlighted by red circles. A photo of the physical LAS is shown in the bottom-left of Figure 3.2. The 24 nodes are at varying height levels. Specifically, nodes at the left and right edges are slightly higher than those in the middle of the LAS. This spatial arrangement distinguishes three types of visitor engagement with the system. When visitors observe the LAS but are not underneath the LAS, no IR sensor is activated, i.e., visitors are observing the LAS but cannot be observed by the LAS sensors. As shown in Figure 3.1, when visitors walk or stand underneath the LAS, which we name *Passive Interaction*, the IR sensors above them are activated, but the distance between the visitor and the system is still large, corresponding to a small reading of the IR sensor. Visitors engaging in *Active Interaction* might also reach their hand upwards to interact with the LAS, resulting in a higher activation value of the closest IR sensor.

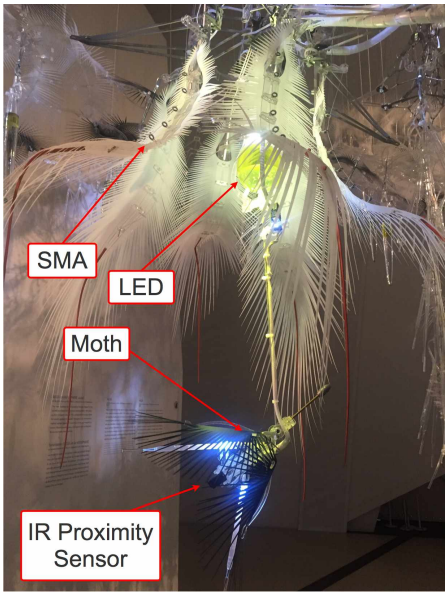
There are eight speakers distributed throughout the LAS. These speakers play two types of sound samples. The first sound is a background sound played on a continuous loop. The second sound is triggered by the IR sensors. These speakers are independently controlled by specialized software, so here we treat them as background behaviours.

Each node in the LAS consists of six Fronds, one Moth, and one high-power LED as its actuated systems and one infrared (IR) sensor, as shown in Figure 3.3. Each Frond includes a shape memory alloy (SMA) wire which contracts when voltage is applied, pulling a cord attached to a flexible co-polyester sheet, as illustrated in Figure 3.3a. The contraction generates a smooth and gentle movement, and when the applied voltage is removed the SMA slowly relaxes to its original shape. The Moth consists of wing-like flexible flaps attached to a small DC Motor that vibrates when activated, making the moth appear to be flapping its wings. The Moth also houses two small LED lights which illuminate during vibration. The single high-power LED located beneath the central flask can be faded to illuminate the coloured liquid in the flask. The IR sensor senses the proximity of visitors, and generates a continuous reading proportional to the distance between any part of the body of a visitor and the sensor location in the canopy.

Two web cameras (labeled Camera1 and Camera2 in Figure 3.1 and Figure 3.2) are used to record video during our experiment and to calibrate sensory data. These two web cameras are mounted on the wall in the front-right and back-left corners of the LAS space.



(a) Node Diagram



(b) Fully Assembled Node

Figure 3.3: Diagram of Node in the LAS

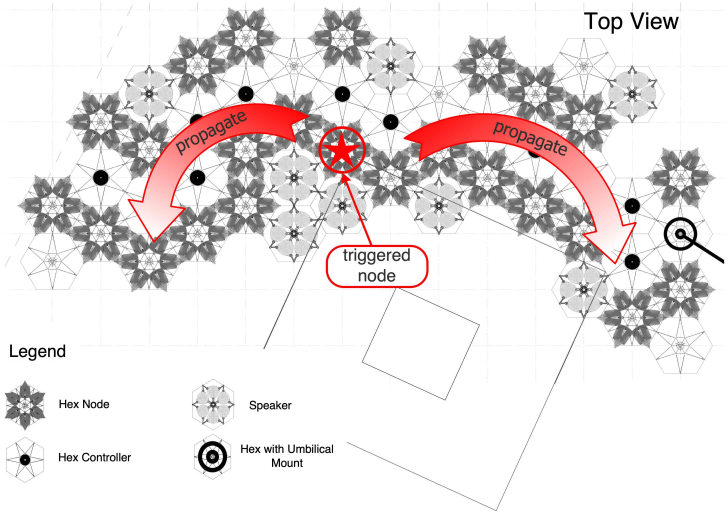


Figure 3.4: Pre-scripted Behaviour

3.1.2 Pre-scripted Behaviour

Pre-scripted behaviour (PB) is the spatially-distributed interactive behaviour manually designed by the architects, and it is also the baseline used for comparison with adaptive behaviours we will describe in Section 3.2. Within the PB mode, the system can be in two types of states: active and background, which are mainly controlled by 17 parameters (shown in Table 3.1) specified by the architects. The values in the Default and Range columns are used in the PB and PLA modes, respectively.

In PB, if any of the IR sensors is triggered, the system enters the active state. In this state, the node corresponding to the triggered sensor will first activate its *local reflex behaviour*. In the local reflex behaviour, the Moth, the LED and six SMAs attached to the same node as the triggered IR sensor will be activated. When a Moth is activated, it will gradually increase the vibration (T^m_{ru}) to its maximum intensity (I_{max}) and then keep vibrating for a period of time (T^m_{ho}). After that, it gradually stops (T^m_{rd}). After a waiting period (T^m_{gap}) following the sensor trigger, the LED on the same node is activated. It ramps up over time period (T^l_{ru}) to its maximum brightness (I_{max}), holds for a period of T^l_{ho} and then gradually dims (T^l_{rd}). At the same time, the SMAs are activated one after another separated by (T^{sma}_{gap}). A voltage is applied to contract the SMA, after which a cooling-down time is started during which this SMA will not be activated again. The activation profile of the SMA wires is fixed in order to protect them from overheating, so these are not included in the parameterization shown in Table 3.1. Meanwhile, this detected event will be propagated from the triggered node to neighbouring nodes (T^n_{gap}) until the edge nodes of the LAS are reached (shown in Figure 3.4), causing a cascade of local reflex behaviours at each node.

If no IR sensor triggering happens for a random time within (T^{min}_{bg},T^{max}_{bg}), the system goes into the background state. In this state, the LED and SMA will activate their local reflex behaviours every random amount of time (T_w,T_{sma}) with probability (P). The choice of activating LEDs and SMAs are independent.

In either state, a sweep of LEDs along the longer axis in either direction of the installation happens every random amount of time (T^{min}_{sw},T^{max}_{sw}). During the sweep, each LED activates local reflex behaviour and propagates in the direction of the sweep.

3.2 Proposed Approach

In this section, we will describe how an adaptive behaviour mode, *Parameterized Learning Agent (PLA)*, is designed to automatically generate interactive actions. The adaptive

Parameters	Meaning	Default	Range
T^m_{ru}, T^l_{ru}	ramp up time: the time it takes for the Moths or LEDs to fade up to their maximum value	1.5	$[0, 5]$
T^m_{ho}, T^l_{ho}	hold time: the time that Moths and LEDs are held at their maximum value	1	$[0, 5]$
T^m_{rd}, T^l_{rd}	ramp down time: the time it takes for Moths and LEDs to fade down to 0	2.5	$[0, 5]$
I_{max}	maximum percentage of duty cycle per PWM period	78	$[0, 100]$
T^m_{gap}	the time gap between the Moth starting to ramp up and the LED starting to ramp up	1.5	$[0, 5]$
T^{sma}_{gap}	the time gap between activation of each SMA arm on the nodes	0.3	$[0, 5]$
T^n_{gap}	the time gap between activation of each node	1.8	$[0, 5]$
T^{min}_{bg}	minimum time to wait before activating background behaviour	45	$[15, 60]$
T^{max}_{bg}	maximum time to wait before activating background behaviour	90	$[60, 100]$
T_w	time to wait before trying to pick a moth or LED	5	$[0, 10]$
P	probability of successfully choosing an actuator	40	$[0, 100]$
T_{sma}	time between choosing SMAs to actuate	0.7	$[1, 5]$
T^{min}_{sw}	minimum time to wait before performing sweep	120	$[5, 200]$
T^{max}_{sw}	maximum time to wait before performing sweep	240	$[200, 400]$

The unit of all time parameters is seconds, except I_{max} and P are percentages.

Table 3.1: Pre-scripted Behaviour Parameters

behaviour uses a standard reinforcement learning framework and an extrinsic reward formulation estimating the occupancy and engagement level of visitors based on IR sensors.

3.2.1 Parameterized Learning Agent: Learning on Top of Pre-scripted Behaviour

The proposed adaptive behaviour, PLA, is designed to learn on top of PB, i.e., parameterized action space, as shown in Figure 3.5. The motivation for this approach is to bootstrap learning by exploiting the designer's knowledge of engaging behaviour, where we hypoth-

esize the designer already has a good idea about what types of actions might be engaging to visitors and this can form a helpful starting point for the learner.

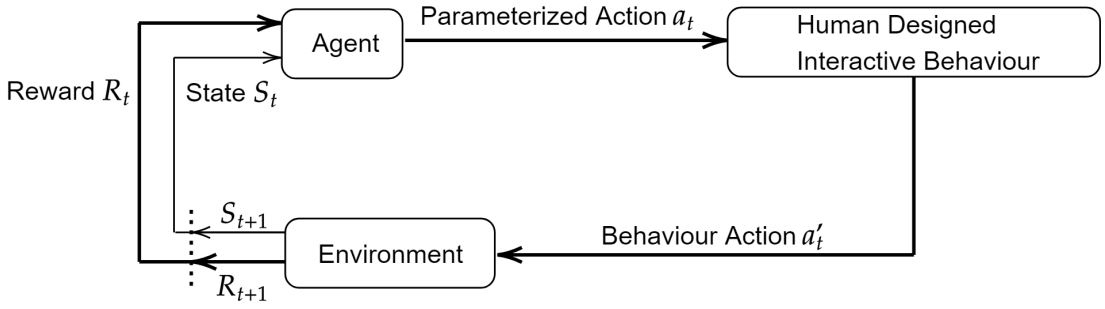


Figure 3.5: Interaction diagram of the learning agent acting on top of human designed behaviours (PB). The agent receives an observation of the current state S_t of the environment and outputs parameters of PB at each timestep.

Observation and Action Space Construction

For PLA, we select 11 parameters from Table 3.1 as the action space, i.e., the dimension of the action vector is 11. This is because some parameters don't take effect until a subsequent trigger or until the current propagation finishes. This could lead to obtaining an observation which is based on both previous and updated parameters. To avoid this issue, we exclude T_{bg}^{min}, T_{bg}^{max}, T_w, P, T_{sw}^{min} and T_{sw}^{max} from the action space. In this way, we make sure every observation is only related to the latest action. To attenuate IR sensor noise, the observation for PLA is an average over 20 IR readings as defined in Eq. 3.1:

$$\boldsymbol{obs}^{(t)} = \frac{1}{20}\sum_{i=0}^{19} \boldsymbol{ir}^{(t-i*\Delta t)} \tag{3.1}$$

where $\boldsymbol{ir}^{(t-i*\Delta t)}$ is the vector of 24 IR sensor readings at time $(t - i * \Delta t)$, $\sum$ is element-wise summation, and $\Delta t \approx 0.1s$ is the time to retrieve one set of 24 IR values from the physical LAS. Thus the dimension of the observation vector is 24 and each observation vector represents the average IR readings over 2 seconds. Based on the observation, the extrinsic reward for PLA is calculated according to Eq. 3.2 where n is 24.

The actions of the learning agent shown in Table 3.1 are scaled into $[-1, 1]$, where -1 corresponds to minimum and 1 corresponds to maximum. The IR readings are scaled into

$[0, 1]$ corresponding to the nearest object being at a detected distance of 80cm or more (no nearby humans detected), to the nearest object being 10cm (very close human detected).

Estimating and Using Engagement as a Reward for Learning

A key feature of our approach is the formulation of the reward function: we wish to learn and reward behaviours which foster visitor engagement. Specifically, the extrinsic reward is computed by summing over the IR observations, which can be regarded as a rough estimate of occupancy and engagement, because: 1) more activated IRs means more people are standing under the LAS, thus indicating higher occupancy; 2) closer distance between visitors and IR sensors implies more active interaction, e.g., looking very closely or raising hands, which are higher engagement behaviours. Therefore, higher occupancy and more active interaction will cause higher extrinsic reward. Formally, given a new observation at time $t+1$, $\boldsymbol{obs}^{(t+1)} = (obs_1^{(t+1)}, obs_2^{(t+1)}, \cdots, obs_n^{(t+1)})$[2], where the value of n depends on the specific behaviour mode, the reward $r^{(t)}$ for taking action $\boldsymbol{a}^{(t)}$ while observing $\boldsymbol{obs}^{(t)}$ can be expressed as Eq. 3.2:

$$r^{(t)} = \sum_{i=1}^{n} obs_i^{(t+1)}. \tag{3.2}$$

where n is the dimension of observation, e.g. $n = 24$ for PLA.

Implementation

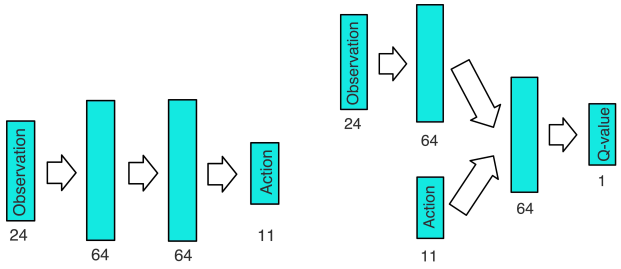
Given the observation and extrinsic reward, the optimal policy can be learned with an RL algorithm. In this thesis, learning is implemented using the Deep Deterministic Policy Gradient (DDPG) algorithm[23], a variant of Deterministic Policy Gradient [21], where both the *actor* and *critic* are approximated with deep neural networks.

We chose DDPG because: 1) the action space is continuous, and compared with benchmarks reported in the literature [55] [56] the dimension of the action space is relatively large, which means that even if we discretize the continuous action space, the dimension of the discretized action space will be very large; 2) DDPG is a well-known algorithm that has been successfully deployed in many applications.

The implementation of the learning algorithm is adapted from OpenAI's Baselines library[57]. The structure of the neural network is shown in Fig. 3.6. All layers are dense layers, with layer-norm applied and ReLu as the activation function for all hidden neurons.

[2]In this thesis, we will use normal lowercase for scalar and bold lowercase for vector.

The number under each layer indicates the neurons in that layer. tanh is used at the output layer of the actor, and linear activation is used at the output layer of the critic.



(a) Actor (b) Critic

Figure 3.6: Actor-Critic of PLA

We discuss the choice of hyperparameters for the simulation and the field study in Section 4.2.5 and Section 5.2 respectively.

Non-episodic Setting

In many reinforcement learning test environments, such as Atari games and OpenAI Gym, the environment is reset to initial states once a termination state is reached. The termination can be either tasks being completed or the death of game characters, usually indicated by a "Done" signal. In our application, we could not define such a termination state, as visitors can always interact with LAS and there is no final state. Therefore, the concept of episodes does not exist. However, we still use the notation, "episode", for two purposes: 1) we set a fixed number of steps per episode and the "episode" indicates how many interactions have happened so far; 2) we train the neural networks at the end of episodes and the "episode" represents the number of training iterations. Implementation details of the algorithm can be found in Appendix A.

Chapter 4

Simulation

In this chapter, we create a simplified simulation environment using Unity to examine the performance of the learning algorithm. The purpose of the simulation is to verify whether the learning algorithm can learn to adapt to visitors' preferences and understand the algorithm parameters that influence learning. Simple visitor models are used to facilitate the initial simulation analysis. Specifically, a single visitor environment and a multiple visitor environment are tested to verify the effectiveness of learning. Two exploration strategies, parameter noise and action noise, are examined and compared.

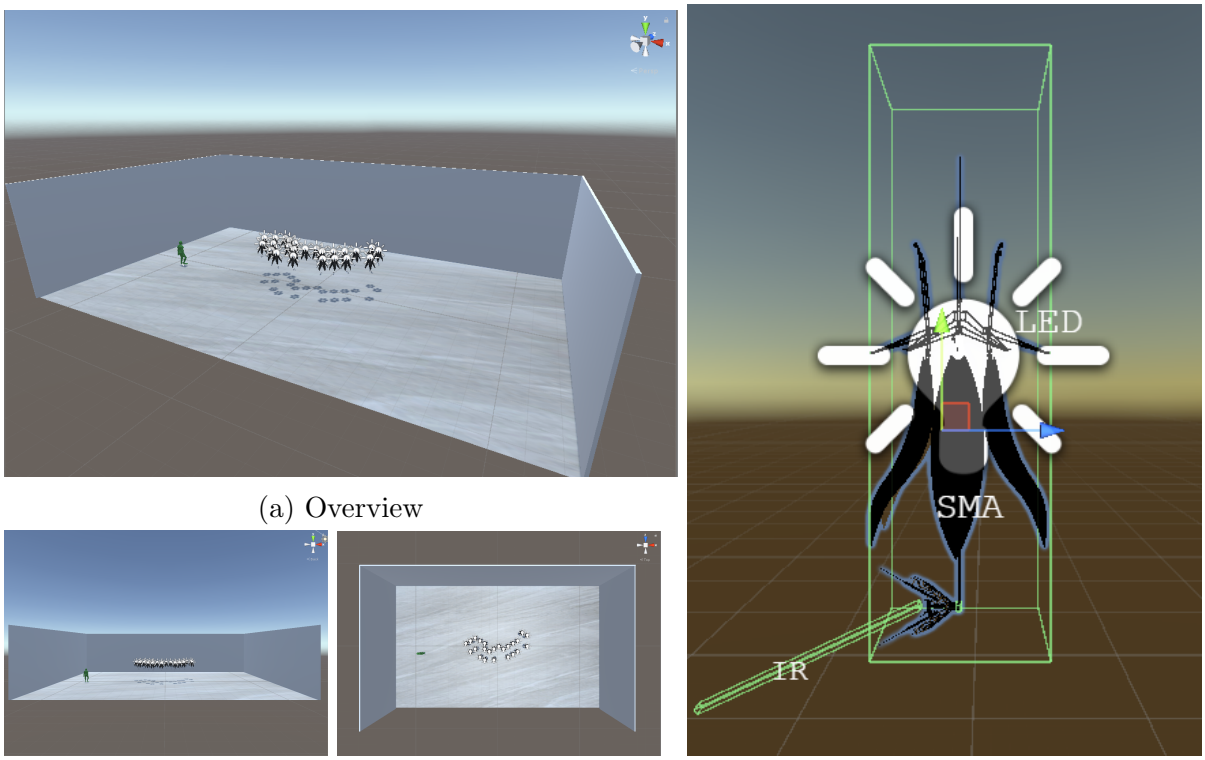
4.1 Simulator

4.1.1 LAS in Unity

A simplified replica of the Living Architecture System described in Chapter 3 is created using Unity engine. The simulator can be seen in Figure 4.1. Inside this scene, we have a $25m \times 15m$ platform surrounded by walls and simulated visitors can walk inside this area. The LAS system consists of 24 nodes and each node has 6 SMAs, 1 LED, and 1 IR sensor. Moths are not simulated for simplicity. Figure 4.1d shows a close-up view of a node. The whole LAS system is centered in the space and the relative position of each node is identical to the Canopy area of Transforming Space installation at ROM.

To simplify the simulation rendering, we replace the SMA actuation from curling to changing colors. Thus, the states of LAS are the colors of SMAs and the light intensity of LEDs.

In the LAS system installed in the ROM exhibition, the orientations of IR sensors are not fixed, and can be changed due to visitor actions. In the simulator, the orientations are randomly generated but fixed through all experiments to ensure consistent simulation results.



(a) Overview

(b) Front view

(c) Top view

(d) A single node

Figure 4.1: LAS simulation using Unity. (a),(b),(c) illustrate the platform and LAS. (d) is a close view of a single node. There are 24 such nodes in the LAS in (a),(b) and (c).

4.1.2 Visitor Simulation

As LAS is an interactive installation, simulating visitors is crucial to our simulation. As in real life, the simulated visitors can observe the current states of actuators in LAS. They will be engaged if they observe *attractions* and the *attractions* can be any particular actions generated by LAS. Based on the visitor's attraction model, each visitor selects the most interesting location by ranking the $Heat$ of each node. The $Heat$ of node i is the $attraction_i$

plus the sum of other nodes' $attraction_j, j \neq i$, divided by distances between node i and j. If all visitors share the same attraction model, it is very likely that multiple visitors will be attracted to the same location. To avoid this swarm behaviour, each visitor also takes travel distances into consideration. In other words, they favour those attractions that are close to themselves. Overall, $Heat$ of node i is defined in Equation 4.1.

$$Heat_i = (attraction_i + \sum_{j,j \neq i} \frac{attraction_j}{d_{ij}})/(D_{iv}) \tag{4.1}$$

where

$$\begin{aligned} d_{ij} &= \text{Distance between the center of node } i \text{ and } j \\ attraction_i &= \text{Attraction of node } i \\ D_{iv} &= \text{Distance between node } i \text{ and visitor } v \end{aligned}$$

Each visitor has its own measurement of the distance between himself and the destination. It considers himself as arrived if the distance is less than 0.1m. However, despite the fact that distances are taken into account, it is still possible that more than one visitor chooses the same destination. Those arriving late will be blocked by early arrivers, and never be able to reach the destination. To model visitors moving if a desired location is too crowded, a 10 second timeout is set. If visitors cannot get to a destination within this amount of time, they will choose another one. The choice of new destination still follows the ranks of $Heat$ defined in Equation 4.1, except that the previous destination is excluded.

The complete behaviour model of visitors is explained in Algorithm 1.

```
while True do
    Take an observation.
    The visitor selects the most interesting location as its destination according to
     Heat.
    while NOT arrived do
        if Timed out then
            break
        else
            Move toward destination.
        end
    end
    if Arrived then
        while attraction at Destination >0 do
            Remain at the destination until no more attraction can be found.
        end
    end
end
```

Algorithm 1: Visitor Behaviour

Overall, the visitor observation is defined in Equation 4.2.

$$\boldsymbol{obs} = (\boldsymbol{Act}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{IsArr}, \boldsymbol{IsTmOut}, \boldsymbol{vx}, \boldsymbol{vy}) \tag{4.2}$$

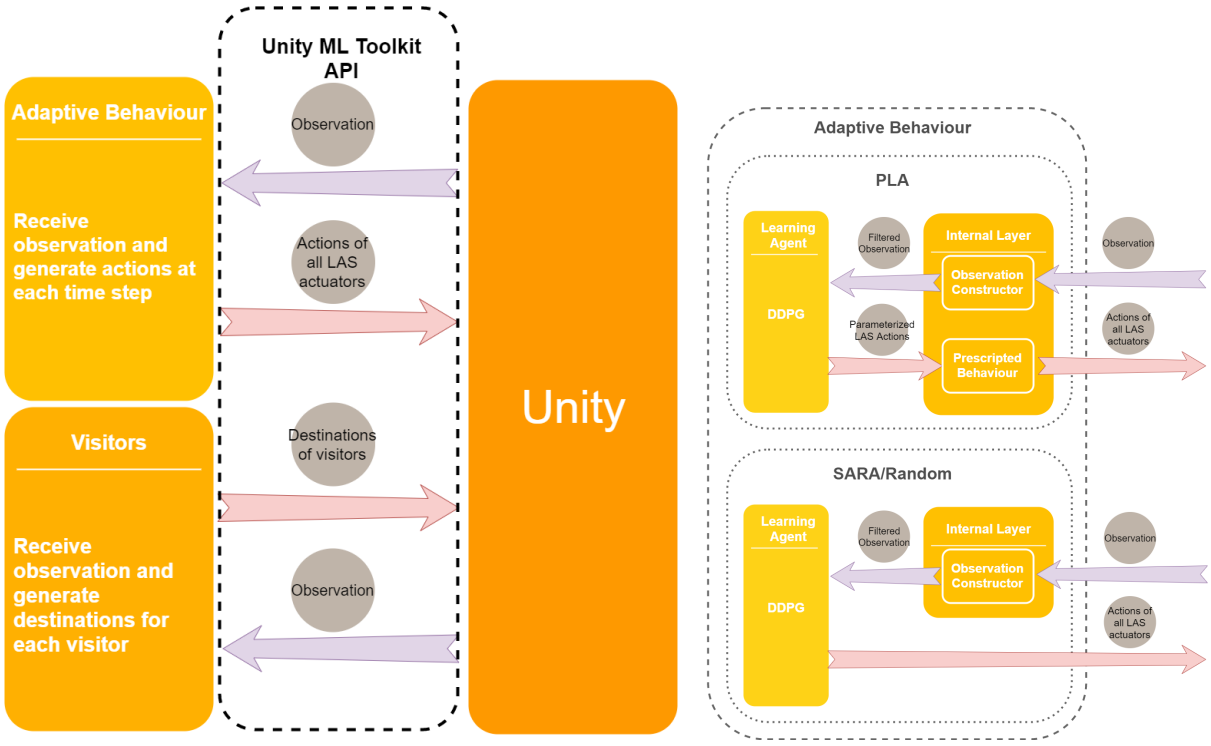
Where

$$\begin{aligned}
\boldsymbol{Act} &= (Act_1, ..Act_k) && = \text{Intensity of actuators (LED or SMA)} \\
\boldsymbol{x}, \boldsymbol{y} &= (x_1, ..x_{24}), (y_1, ..y_{24}) && = \text{Coordinates of center of all nodes} \\
\boldsymbol{IsArr} &= (IsArr1, ..IsArrv) && = \text{Whether each visitor has arrived at destination} \\
\boldsymbol{IsTmOut} &= (IsTmOut1, ..IsTmOutv) && = \text{Whether each visitor has timed out} \\
\boldsymbol{vx}, \boldsymbol{vy} &= (vx_1, ..vx_v), (vy_1, ..vy_v) && = \text{Coordinates of each visitor}
\end{aligned}$$

4.1.3 Interface Structure

A new simulator interface structure is designed to handle the communication between LAS and the learning algorithms, and to control simulated human visitors. We use Unity Machine Learning Toolkit[56] as the Python interface. The Unity Machine Learning Agents

Toolkit (ML-Agents) is an open-source Unity plugin that enables games and simulations to serve as environments for training intelligent agents. We adopt Python API into our simulator's interface, as shown in Figure 4.2. As shown in Figure 4.2a, the Unity ML Toolkit handles communication between Unity and two separate modules: adaptive behaviour module and visitor module. For the adaptive behaviour module, it passes observations to the module and returns actions to Unity. Figure 4.2b shows the adaptive behaviour module, where two types of structures are used. SARA and Random are two additional behaviour modes and will be detailed later in Section 4.2.1. The difference between the two types is that SARA and Random output actions directly into Unity, whereas PLA's actions are translated into raw actuator actions by Prescripted Behaviour described in Section 3.1.2. Meanwhile, the visitor module receives visitor observations and produces desired destinations of each visitor. Inside this module are the visitor behaviour models described later in Section 4.1.2.



(a) Overview (b) Adaptive Behaviour Module

Figure 4.2: Simulator interface.

4.1.4 Specifications

Each simulated visitor has a cylinder physics collider, which can trigger IR sensors and keep distances to other visitors. The radius of the cylinder is set to 0.1 m, close to the radius of a human's head. The reason for such a small radius is that due to the dimension of LAS, if we set visitors' collider cylinder radius to be larger, such as 0.5 m, then multiple visitors could not move freely within the area under LAS without blocking each other's path.

In the field experiment, we want the system to take actions as frequently as possible in order to generate more interactions, but at the same time not too fast for visitors to respond. This ratio r between human speed and LAS action frequency is defined in Equation 4.3. For the LAS of this project, the system takes 10 observations per second (10 Hz) and one action per 20 observations (0.5Hz). Human's average walking speed is 1.4 m/s. This gives the ratio r of 2.8.

$$r = \frac{\text{Human Speed}}{\text{Action Frequency}} \tag{4.3}$$

In the simulation, we would also like to keep the same ratio r. Therefore, when accelerating the simulation speed, we accelerate the visitor moving speed and adjust the number of observations per action accordingly. The physics render interval in Unity is by default 0.02 s, thus the system frequency is 50 Hz. The parameters related to timing in PLA are also adjusted accordingly. A comparison of parameter specifications is provided in Table. 4.1.

	In Reality	In Simulation using Simulator Time
Observations per Action	20	25
System Freq(Hz)	10	50
LAS Action Freq(Hz)	0.5	2
Human Speed(m/s)	1.4	5.5
Ratio r	**2.8**	**2.75**

Table 4.1: Parameters in simulation vs. parameters in reality

4.2 Experiment Design

4.2.1 Adaptive Behaviour Modes

Three adaptive behaviour modes are tested in the simulation. Apart from PLA described in Section 3.2.1, we consider two behaviour generating modes that act in the raw actuation space. The first mode is *Single Agent Raw Action Space (SARA)* and it has a single learning agent controlling all the individual actuators. SARA has the same observation space and reward formulation as PLA. For more implementation details, see Appendix B. The second mode is *Random* and it also directly controls each actuator, except that all actions are randomly generated. The purpose of using Random is to have an estimation of reward baselines. A summary of the three behaviour modes is given in Table 4.2.

Mode	Learning Agent	Observation Dimension	Action Dimension
PLA	Yes	24	11
SARA	Yes	24	168
Random	No	24	168

Table 4.2: Summary: behaviour modes in simulation.

4.2.2 Exploration Methods

Two exploration methods are compared in the experiment. As we know, DDPG is an off-policy algorithm and its policy π_θ is deterministic. The exploration is realized through perturbation of actions. A common practice for creating perturbations is to apply a Gaussian noise $\mathcal{N}$ on the action[21], which is described in Equation 4.4. We will refer to this method as *action noise* in the rest of the thesis.

$$a_t = \pi_\theta(s_t) + \mathcal{N}_t(0, \sigma) \tag{4.4}$$

An alternative way is to apply a Gaussian noise $\mathcal{N}$ to neural network parameters θ and get a perturbed policy $\pi_{\tilde{\theta}}$[58], where $\tilde{\theta} = \theta + \mathcal{N}_t(0, \sigma)$. This method will be referred to as *parameter noise*. The action is then given by new policy $\pi_{\tilde{\theta}}$ as in Equation 4.5.

$$a_t = \pi_{\tilde{\theta}}(s_t) \tag{4.5}$$

Because the impact of the scale σ in $\mathcal{N}_t(0, \sigma)$ on the noise range of action a strongly depends on the specific network architecture and its parameters, σ needs to be dynamic and

adaptive. It is adjusted by a constant α depending on the distance $d(\cdot, \cdot)$ of resulting actions (see Equation 4.6,4.7). The distance is estimated from a batch of experience sampled from the memory buffer.

$$\sigma_{k+1} = \begin{cases} \alpha\sigma_k & \text{if } d(\pi, \pi_{\tilde{\theta}}) \leq \delta \\ \frac{1}{\alpha}\sigma_k & \text{otherwise} \end{cases} \tag{4.6}$$

$$d(\pi, \pi_{\tilde{\theta}}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_s \left[(\pi(s)_i - \pi_{\tilde{\theta}}(s)_i)^2 \right]} \tag{4.7}$$

The resulting actions will have a relatively consistent exploration related to δ. Therefore, this parameter noise could lead to more consistent exploration and a richer set of behaviors. Thus, we would like to compare which method is suitable for our tasks.

4.2.3 Visitor Attraction Models

As mentioned in Section 4.1.2, visitors are simulated to exhibit behaviours based on *attractions*. Real humans' interests are complicated and modelling all possible stimuli is unfeasible. Hereby, we only consider two simple attractions based on actions generated by LAS. Firstly, we consider the intensity of an actuator as a basic attractor. Secondly, because behaviours are time-related actions, we consider an action sequence as another attractor. Therefore, two types of visitor attraction model, I and II, are crafted accordingly. To further simplify the design, we arbitrarily use LED in the attraction models.

Attraction model I is based on the simple relationship to the intensity of each LED: the brighter LEDs, the more attractive they are to visitors. The attraction of node i is defined as:

$$attraction_i = I_{LED_i}$$

Attraction model II uses the number of action sequences observed in a time window. The choice of time window length is arbitrary, and is set to 4 seconds. The attraction of node i is then defined as:

$$attraction_i = N_{\text{action sequence}}$$

The action sequence is a state transition from low to high, or vise versa. The duration of each state is at least 1 second. The state is defined as follows:

$$state = \begin{cases} LOW & I_{LED_i} \leq 0.5 \\ HIGH & I_{LED_i} > 0.5 \end{cases}$$

Since PLA is better at generating time-related action sequences, whereas SARA is better at generating time-independent actions, we expect these two attraction models to be helpful for comparing PLA to SARA. As visitors' preferences influence the performance of LAS and learning algorithms, the choice of attraction model could help reveal the distinctions between two behaviour generating modes.

4.2.4 Non-Episodic Simulation

As discussed previously in Section 3.2.1, the learning algorithm is used in a non-episodic setting. As a result, in the simulation, the states of LAS and visitors are carried over between episodes. No reset is applied in each run so that we simulate an interactive environment in which agents are continuously learning and evolving.

4.2.5 Hyperparameter Selection

We set the learning rate of the actor and critic to be 10^{-4} and 10^{-3}. The discount factor $\gamma = 0.99$ and the batch size is 64. We experimented with different neural network sizes: 64, 128 and 300 neurons in each layer of the two layer network (see Figure 3.6), and find no obvious influence on the results. Therefore, we only show the results using 64 neurons in each layer.

4.3 Single Visitor Environment

4.3.1 Setup

We firstly examine how learning algorithms perform in a simple situation by placing one visitor in the environment. The two attraction models are tested separately, and under each mode, two exploration methods are compared. The visitor is always spawned in a fixed location away from LAS, so that it cannot be immediately detected by LAS.

We run each case 5 times with the same initial state of the LAS and visitors, the length of each run is 1000 episodes and each episode is 25 steps. Because there is one visitor in the space and the IR sensor value can range from 0 to 1 (see Section 3.2.1), the maximum reward at every timestep step is 1, assuming the visitor could be only detected by one IR sensor in each timestep. Note that this assumption is not always correct as there exists

overlapping in IR detection areas. But the overlapping areas are all far away from sensors, therefore, the sum of two IR sensor readings is still less than 1. Then we estimate the total maximum reward of each episode to be $1 \times 25 = 25$.

4.3.2 Results

We report the average return of all runs (solid line) and the variance (shaded region) for each behaviour generating mode and noise method, and show visitor position histories in the space.

Attraction Model I

From Figure 4.3 we can see that SARA's returns are higher than those of PLA and Random. Using either exploration method, SARA reaches rewards close to maximum (25) and much higher than PLA's rewards. This matches our expectation, because attraction model I is based on the intensity of each LED. SARA has the ability to control individual actuators including LED and any LED with intensity $I_{LED} > 0$ will be an attraction to the visitor, so this task can be easily achieved with SARA. However, PLA can only control the parameters of behaviours in which LEDs must follow a ramp-up - hold - fade sequence. The difference in action space causes the difference in rewards. Further, the reward of PLA is higher than the reward of Random, showing that there is still a benefit to learn over taking random actions, even when the action space is not a good match to the visitor preferences.

With SARA, rewards when using parameter noise are noisier than when using action noise. This is because this task is relatively easy. The learning algorithm with action noise quickly finds and sticks to an optimal policy, while parameter noise explores the action space more thoroughly than action noise and produces sub-optimal actions that are less effective in attracting visitors during exploration. When comparing exploration methods in PLA, it can be seen that parameter noise exploration gives higher rewards and an improving trend, while action noise exploration has similar result as Random and no sign of improvement over the learning process. It is hypothesized that parameter noise has wider exploration than action noise, thus leading to a relatively higher reward in PLA.

Figure 4.4 shows a sample of visitor travel history during the entire training process using action noise. Similar results are also observed in using parameter noise and here we only show one of them. From the graph it is clear that the visitor travels much more in PLA than in SARA. In SARA, the visitor mainly stays under one single node after the learning algorithm quickly learns to keep the lights on. But in PLA, the visitor seems

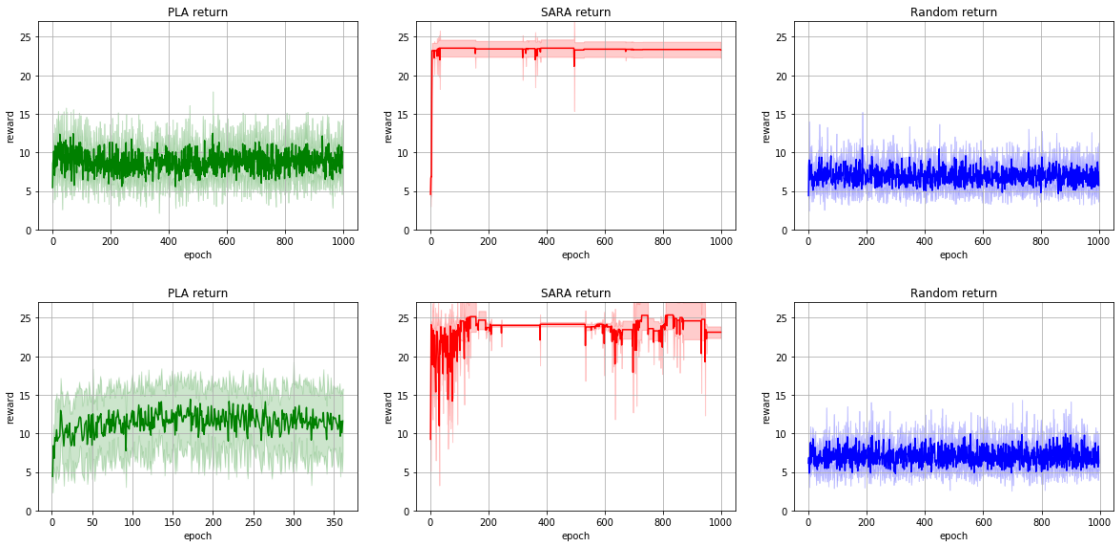


Figure 4.3: Attraction Model I: Cumulative rewards under PLA, SARA and Random. First row: action noise; second row: parameter noise.

to be constantly moving and switching destinations, concentrating within the area under LAS. This could explain the reward difference between SARA and PLA. If the visitor moves frequently in simulation space, their time within the IR detection range is reduced. Conversely, if the visitor tends to stay under one of the sensors and changes his destination less often, the rewards will be higher. This also shows that SARA is better at providing the visitor's desired action, which is determined by the match between the visitor attraction model and the learning agent's action space.

Attraction Model II

Figure 4.5 shows the accumulated rewards of LAS in the simulation in which a visitor is attracted by simple action sequences. The learning algorithm achieves approximately maximum rewards using parameter noise in both PLA and SARA. Meanwhile, SARA using action noise leads to a reward similar to Random at the end of training and PLA is slightly higher.

Notice that the variance of rewards using action noise is significantly greater than parameter noise, because there is a huge discrepancy between different training runs. Figure 4.6 compares training runs using two exploration methods in SARA. As shown in the figure on the left, runs with action noise differ greatly from each other. In some runs,

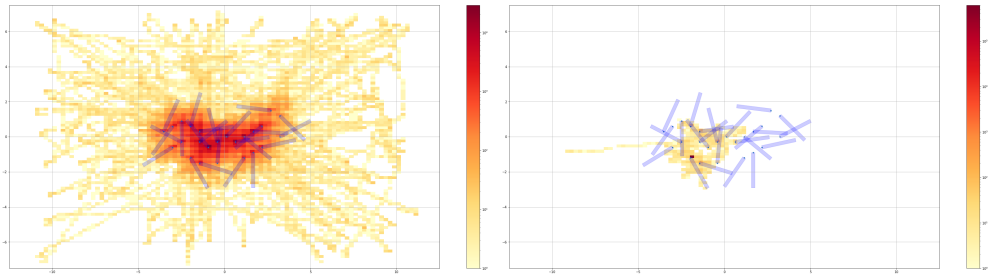


Figure 4.4: Attraction Model I: Sample visitor positions during the entire training process. Left: PLA; Right: SARA. 24 blue dots are the center of LAS nodes and light blue bars are IR detection ranges. Each pixel (from yellow to red) indicates the number of times the visitor occupied the area covered by the pixel.

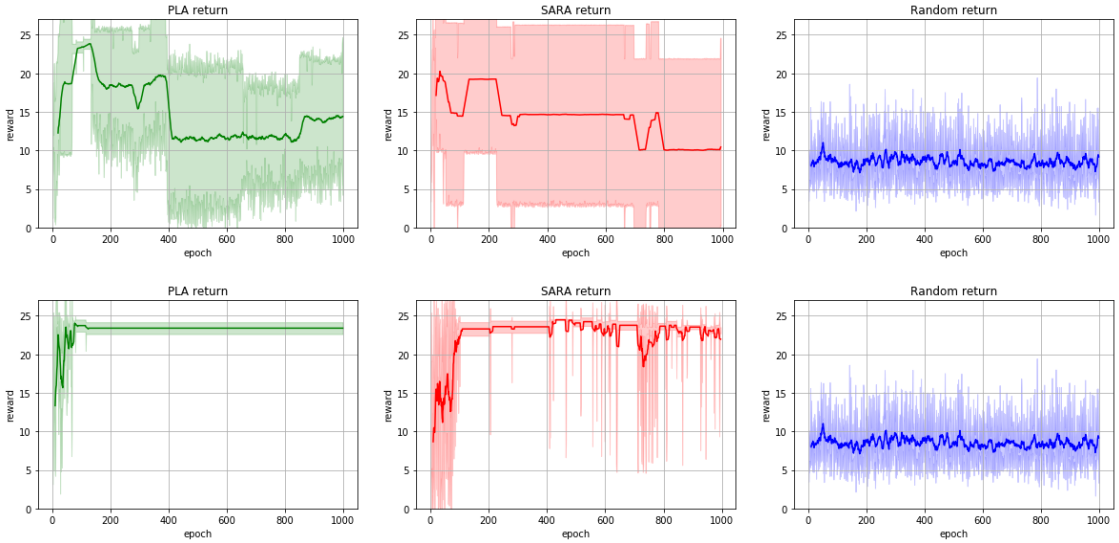


Figure 4.5: Attraction Model II: Cumulative rewards under PLA, SARA and Random. First row: action noise; second row: parameter noise.

they completely fail to attract visitors, while others might do well. This is because with action noise, the exploration is not as thorough as parameter noise and in a single visitor environment, the reward is sparse. In each run, the parameters of the neural network are randomly initialized (see Appendix A). If the network is initialized to a bad policy, it is extremely hard to learn with barely any reward. Only occasionally, such as run3, the system is initialized with a good initial policy which is capable of attracting visitors, then this

policy is reinforced over the training process. On the contrary, a policy using parameter noise could always generate attractions to visitors, regardless of the initial policy.

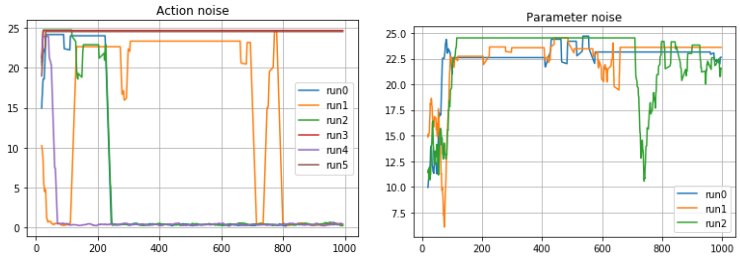


Figure 4.6: Comparison of training runs under SARA: parameter noise vs. action noise

Figure 4.7 compares the visitor's position history between PLA and SARA, both using action noise. We can see that in SARA the visitor either stays under nodes or randomly walks in the whole space. In contrast, with PLA the visitor's position is condensed near LAS. We compare the amount of time the visitor spent right under the nodes and within the LAS area using PLA and SARA. In Table 4.3, we can see that on average the visitor spends less time directly under the nodes in PLA than in SARA. This would cause an obvious gap in average rewards between SARA and PLA. However, we do not see such gap in Figure 4.5. On the other hand, if we look at the percentage of time within the LAS area in Table 4.3, this gap between SARA and PLA is narrower. Based on this, we believe the visitor is detected by IR sensors along the way while travelling under the LAS, and this compensates the loss in rewards obtained by the learning algorithm.

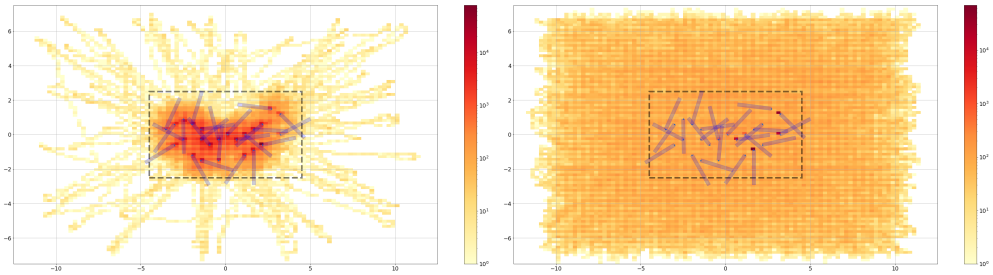


Figure 4.7: Attraction Model II: Sample visitor positions during the entire training process. Left: PLA; Right: SARA. 24 blue dots are the center of LAS nodes and light blue bars are IR detection ranges. Each pixel (from yellow to red) indicates number of times the visitor occupied the area covered by the pixel. The dashed line square outlines the LAS area.

	Percent of Time Under Nodes		Percent of Time In LAS Area	
	Avg	Std	Avg	Std
SARA	0.653	0.297	0.833	0.212
PLA	0.465	0.290	0.724	0.289

Table 4.3: Attraction Model II: Visitor position statistics of all runs using action noise.

4.4 Multiple Visitor Environment

4.4.1 Setup

In the real application, multiple visitors can be present within the space at the same time and we want to investigate how the learning system can perform in such a case.

We place 5 visitors inside the simulation space. All visitors share the same observations which are identical to those observed by a single visitor in Section 4.3. Two attraction models are inspected separately and all visitors share the same attraction model in each experiment. Like the single visitor environment, all visitors initially spawn within a fixed area away from LAS.

We run each case 5 times and each time the neural network is initialized randomly. The length of each run is 1000 episodes and each episode has 25 steps. Assuming each visitor can only be detected by a single IR sensor at each timestep, the estimated maximum reward per episode is $5 \times 25 = 125$.

4.4.2 Results

We report the average return/reward of all runs (solid line) and the variance (shadow region) for each behaviour generating mode and noise method, and show visitor position histories of all 5 visitors.

Attraction Model I

In Figure 4.8, the rewards of three behaviour configurations using two different exploration methods are shown. In both exploration methods, SARA obtains higher reward than PLA, which is consistent with the results of the single visitor experiment in Section 4.3.

Comparing the two different exploration methods, we can see that PLA has better results using parameter noise than action noise. In SARA, the final rewards are similar and the parameter noise method is marginally better. However, the speed of learning using action noise is much faster than using parameter noise.

We further examine the policy SARA learnt by plotting the agents' action history and IR observation history shown in Figure 4.9 and Figure 4.10. By comparing the actions in the first row against the second row in Figure 4.9, we can see that the agent learns to keep three nodes (circled in red) ON. However, as the reward structure does not penalize

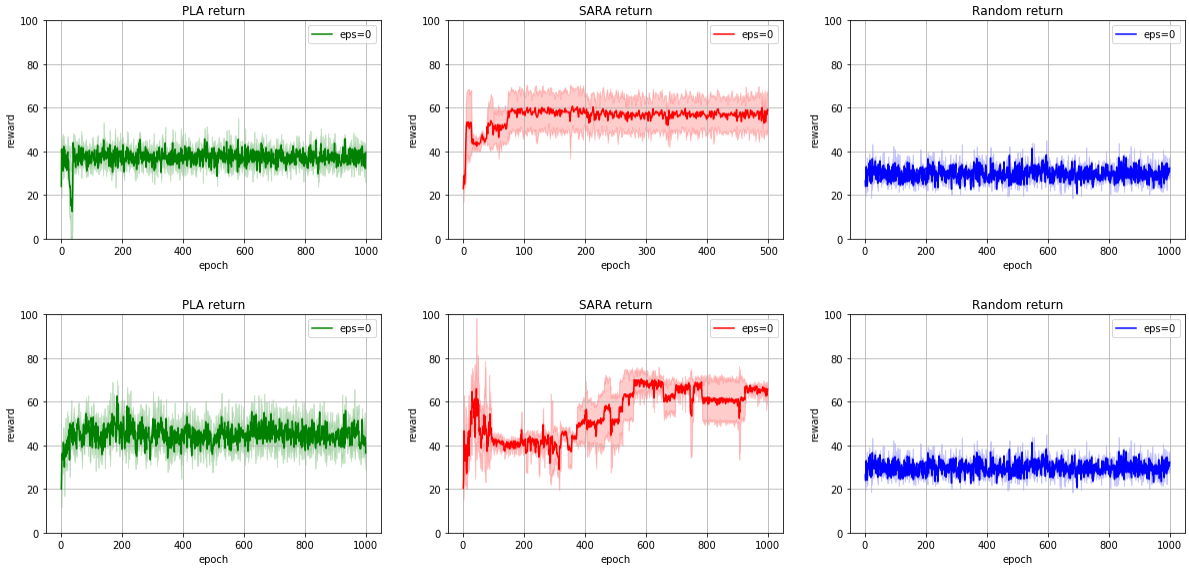


Figure 4.8: Attraction Model I: Cumulative rewards under PLA, SARA and Random. First row: action noise; second row: parameter noise.

turning on LEDs, therefore, there are other lights which are also kept on at the end of training. By observing the visitors' positions in Figure 4.10, we can see that at the end of training, visitors are all standing under or near the nodes which have their LEDs kept ON. This illustrates that the learning algorithm can learn visitors' preferences, which are turned ON LEDs. In most of the runs, the learning agent is only capable of attracting visitors to three nodes, with two nodes surrounded by more than one visitor. The reward is not maximized yet under this situation. The task of distributing all visitors to separate nodes seems to be much harder to learn.

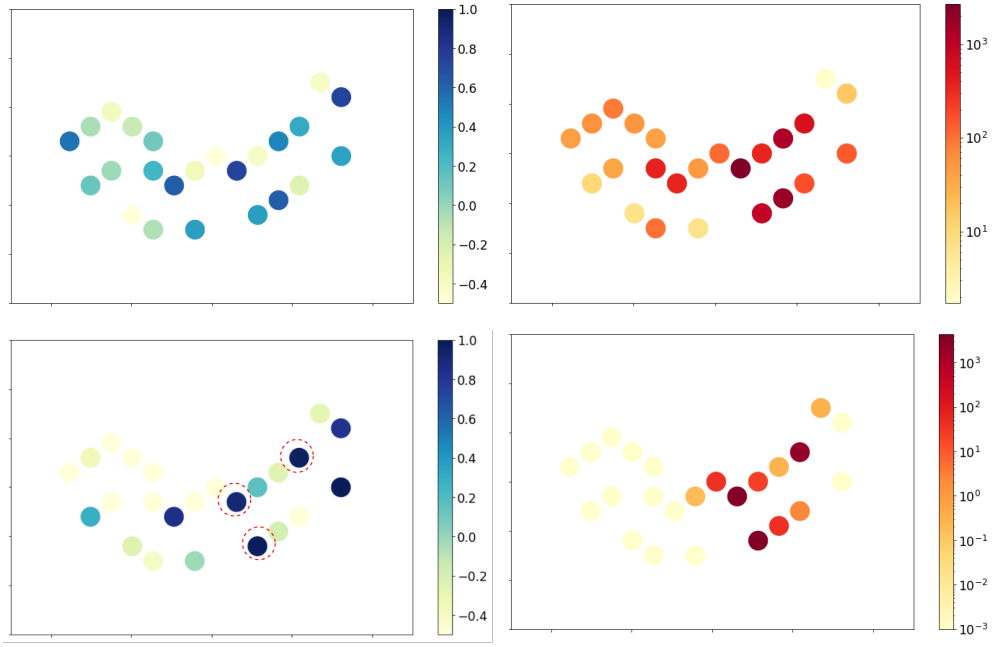


Figure 4.9: Upper-left: Average of first 5000 actions; Upper-right: Log sum of first 5000 IR observations; Lower-left: Average of last 5000 actions; Lower-right: Log sum of last 5000 IR observations.

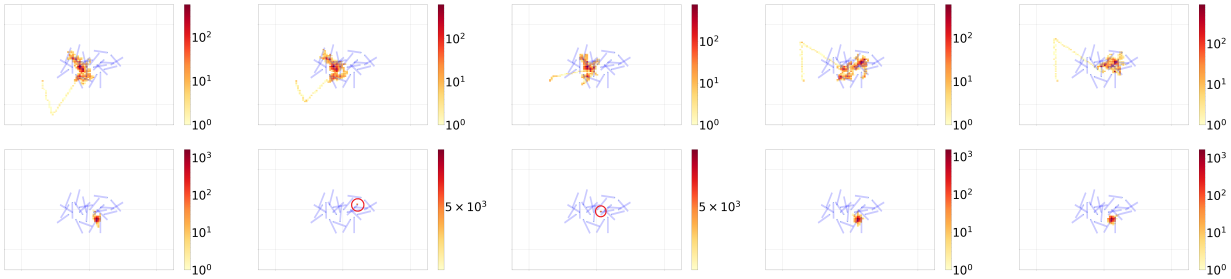


Figure 4.10: Attraction Model I: Top row: Heat map (log scale) of 5 visitors for first 5000 timesteps; bottom row:Heat map (log scale) of last 5000 timesteps. Blue dot is the location of IR sensor and its detection area is represented by light blue bars. Each pixel (from yellow to red) indicates number of times the visitor occupied the area covered by the pixel.

Attraction Model II

From Figure 4.11 we can see that PLA has higher rewards than SARA in both exploration methods, while Random receives the lowest rewards. This demonstrates that PLA is better at generating visitors' desired action sequence than SARA and Random. In fact, the attraction model II is a subset of PLA's behaviours. For SARA, however, given a learnt deterministic policy, the action sequence can only be produced with the help of action noise. The chance of SARA generating a desired sequence is reduced compared to PLA. The random policy can generate any sequence of actions, but the requirement of minimum duration of each state is hard to satisfy. Thus Random obtains the worst rewards among the three adaptive behaviour modes.

According to Figure 4.11, parameter noise exploration decreases the learning speed of PLA and SARA, which is consistent with the results with attraction model I. Final reward levels of two exploration methods are similar in both PLA and SARA, except parameter noise exploration is marginally better in SARA.

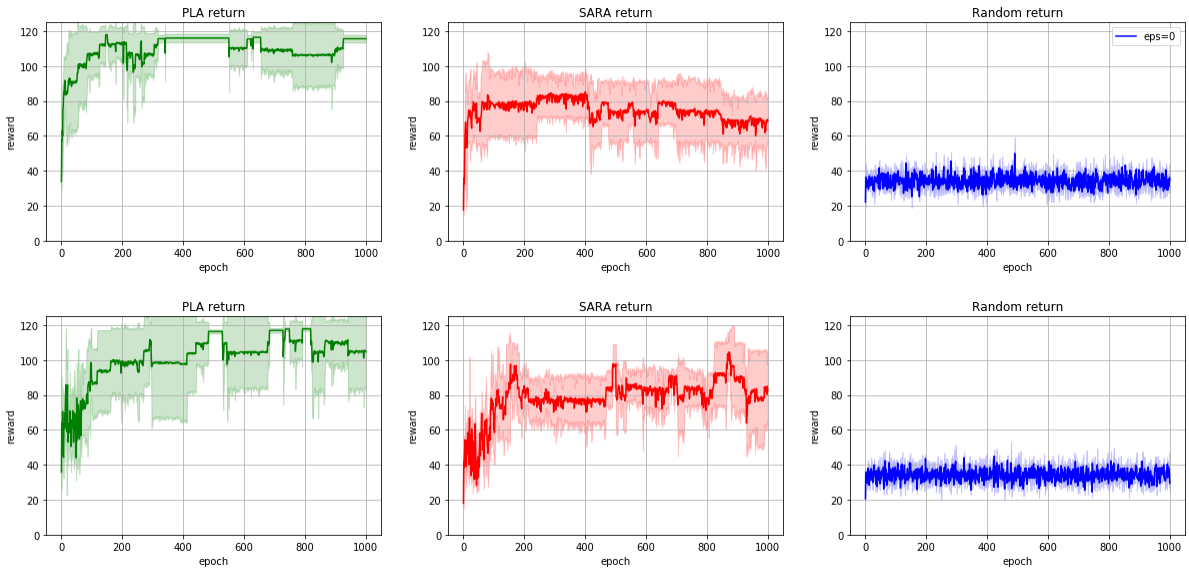


Figure 4.11: Attraction Model II: Cumulative rewards under PLA, SARA and Random. First row: action noise; second row: parameter noise.

We notice that with PLA, the learning algorithms achieve rewards close to maximum by the end of training. We show the learnt policy by plotting the visitor position history

at the beginning and the end of training process in Figure 4.12. Both exploration methods have similar results, so we choose action noise for demonstration. From the figure we can see that initially visitors move around under LAS. By the time training finishes, all visitors found their own attractive spots (circled in red) and stay there. They keep triggering the IR sensor of the node above them, and LED in the node keeps making ramp-hold-fade action sequences to maintain occupancy. Further, we notice a phenomenon that is different from SARA in attraction model I. In this case, each visitor stands under a different node, which optimizes the rewards of LAS. We hypothesize that the propagation behaviour in PLA helps distribute visitors to separate nodes.

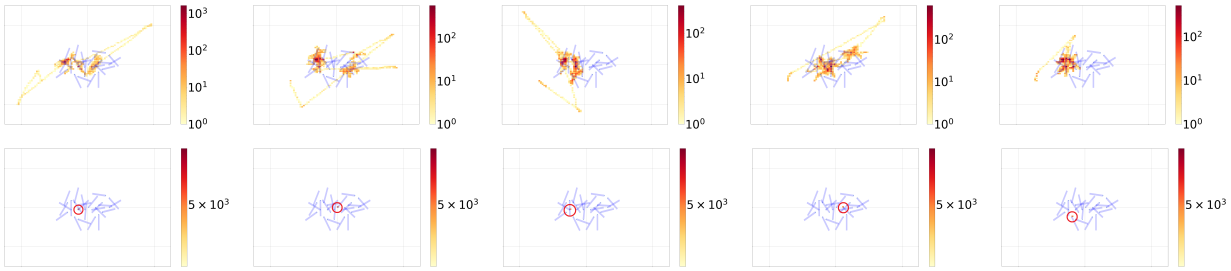


Figure 4.12: Attraction Model II: Top row: Heat map (log scale) of 5 visitors for first 5000 timesteps; bottom row:Heat map (log scale) of last 5000 timesteps. Blue dots are locations of IR sensors and light blue bars represent their detection areas. Each pixel (from yellow to red) indicates number of times the visitor has appeared in the area covered by the pixel.

It is worth mentioning that attraction model II has differing results between single visitor (Figure 4.5) and multiple visitor environments (Figure 4.11). Specifically, there are two differences: **(i)** rewards using action noise **(ii)** difference between PLA and SARA.

- **(i)** Compared to the failure in the single visitor situation, the learning algorithm using action noise did acquire a policy that attracts visitors in the multiple visitor setting. We believe the learning algorithm benefited from the abundance of rewards brought by more visitors. There is a higher chance of receiving rewards when more people are present in space. Further, we hypothesize that the increase in number of visitors helps shaping reward gradient which is beneficial to learning. Reward gradient refers to a characteristic of reward formation that could guide learning into a correct direction. The numerical gradient of rewards could inform learning algorithms about which actions or states are more preferable. This is a similar idea to the potential-based reward shaping function proposed by Andrew Y. Ng. et al.[59]. In the multiple visitor environment, different number of visitors detected by IR sensors can form a gradient

and tells learning algorithms to attract as many as possible, which can be considered as a subgoal-based heuristic for potential-based shaping function in [59]. In the single visitor case, however, the reward is sparse and no gradient can be utilized. The task is thus harder to learn.

- **(ii)** In the single visitor experiment, results of PLA and SARA are equally good or bad. But when multiple visitors are present, PLA is better than SARA using both exploration methods. The rise of difficulty exposes the advantage of PLA in generating action sequences over SARA.

4.4.3 Stochastic Visitor

To make the simulation closer to reality, we introduce randomness into visitors' behaviour. Every time a visitor determines its destination, it has a probability of ϵ to go to a random position. ϵ is set to different levels, $\{0, 0.01, 0.05, 0.1, 0.3\}$, representing different levels of unpredictability of human beings.

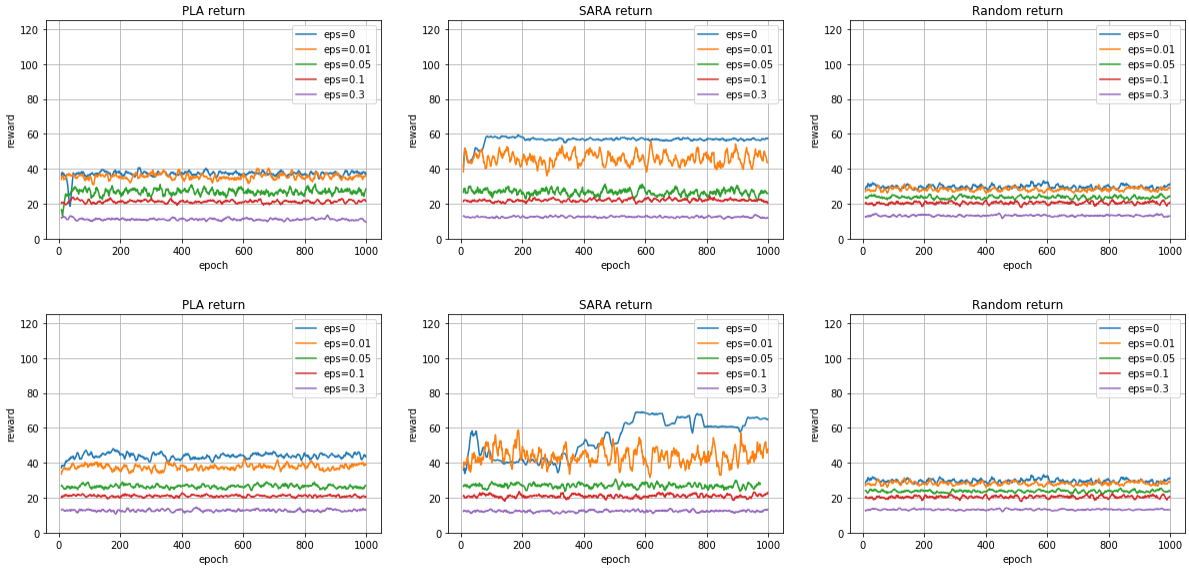


Figure 4.13: Attraction Model I: Reward of different level of randomness. First row: action noise; second row: parameter noise

The average returns of all runs are shown in Figure 4.13 and Figure 4.14. By looking at the third column of both figures, we can tell that as ϵ increases, the rewards that LAS can

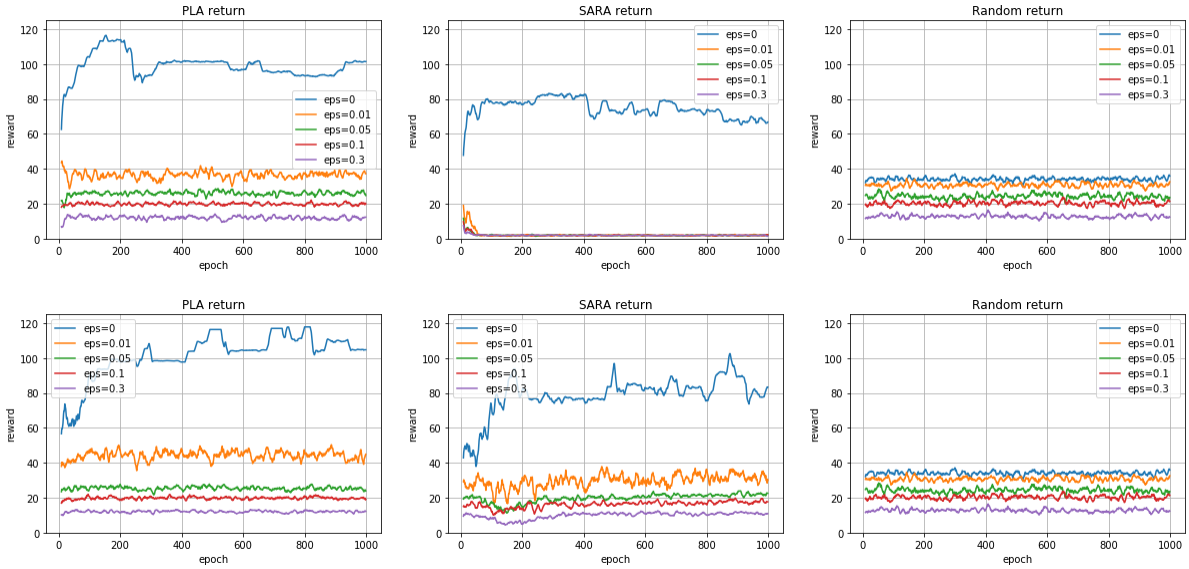


Figure 4.14: Attraction Model II: Reward of different level of randomness. First row: action noise; second row: parameter noise

get are generally reduced. This is expected as randomly selecting a destination will make visitors less likely to stay under or walk towards a node and be captured by IR sensors.

In Figure 4.13, when $\epsilon > 0$ and visitors are not deterministic anymore, rewards of both PLA and SARA decrease as ϵ increases. The magnitude of variation is greater than Random, which means learning algorithm performance is affected by ϵ.

In Figure 4.14, there is a drop in performance when $\epsilon > 0$. This is because when visitors arrive at a node, they decide whether to stay by polling the state of LEDs at each timestep. This decision has ϵ chance to cause the visitor to leave and move towards a random location (refer to Algorithm 1). It notably reduces the amount of time visitors are captured by IR sensors, thus diminishing rewards and increasing the difficulty of learning. According to the figure, learning with action noise in SARA is mostly influenced by this lack of reward. Larger exploration range by parameter noise helps offset this effect.

Overall, DDPG's learning ability is hindered by visitors' stochasticity and this effect is more severe as stochasticity increases. In reality, people's randomness may not be as high as 30%, but will still affect learning of algorithms.

4.5 Conclusion

In this chapter, we showed the performance of learning algorithms in a simulated environment with single and multiple visitors. In the single visitor environment, parameter noise exploration results in better rewards in PLA than action noise exploration, because of larger range of exploration. But it has a slight cost of learning speed in SARA. Similar results are observed in the multiple visitor environment and the effect is more prominent than in the single visitor case.

In both environments, it can be concluded that PLA and SARA have their own advantages when the visitor preference is different. Specifically, when the visitor preference is a subset of adaptive behaviour actions, the learning algorithm performs better. The advantage of different behaviour generating methods under different visitor attraction modes becomes more prominent in the multiple visitor environment than in the single visitor environment.

In addition, compared to a single visitor, multiple visitors yields richer rewards which is helpful to learning.

When visitors become stochastic, the ability of learning decreases as stochasticity increases and using parameter noise can offset the effect to some extent.

In a field experiment, the environment and visitors are much more complex. Each visitor has their own attraction model and it may change over time. As a result, a certain action might only interest some of the visitors, and visitors might get bored after certain a amount of time. A policy learnt from one group of visitors might not work on another. To accommodate unpredictable changes in the non-stationary environment, wider exploration is more desirable and therefore parameter noise is a better choice.

Chapter 5

Field Experiment

In this chapter, we deployed the proposed reinforcement learning approach in the Aegis installation and conducted a field study. We describe the experimental and data processing procedures. Finally we give a quantitative analysis of the results.

5.1 Adaptive Behaviour Modes

Four adaptive behaviour modes were tested in the field study. The first three modes are PB, PLA and SARA described in Section 3.1.2, 3.2.1 and 4.2.1. Like SARA, the fourth mode, *Agent Community Raw Act (ACRA)*, is designed to act in raw action space, i.e. directly control actuators, rather than acting in parameterized action space. SARA directly controls all actuators of the LAS using a single agent, while ACRA controls the raw actuators in a decentralized way, where a distributed multi-agent learning system replaces the single large learner in SARA. Compared with SARA, ACRA enables reducing the dimension of the action and observation spaces for each agent in ACRA, and by sharing observations among agents ACRA is also hypothesized to exhibit cooperating and competing behaviour as in other multi-agent systems[60]. In the experiment, SARA and ACRA are implemented by Lingheng Meng. Therefore, in this thesis, we focus on our analysis on the PLA and PB results.

5.2 Implementation Choice

As stated Chapter 4 Section 4.5, we use parameter noise in PLA as the exploration strategy. The choice of hyperparameters are identical to those in the simulation (see Section 4.2.5). Selecting hyper-parameters with real experiments is challenging, since we do not have a "validation set" that allows us to do multiple runs to choose hyper-parameters. Therefore, all hyper-parameters used are empirically chosen.

5.3 Experimental Procedure

Our experiment was conducted for two weeks from September 14 to October 3, 2018, at the ROM. We were permitted by the ROM to collect data from 1 p.m. to 4 p.m. every day on weekdays. At the same time, we conducted in-person surveys on September 18, 20, and 27. During the entire experiment period, visitors were free to visit and interact with the installation without any interference from researchers.

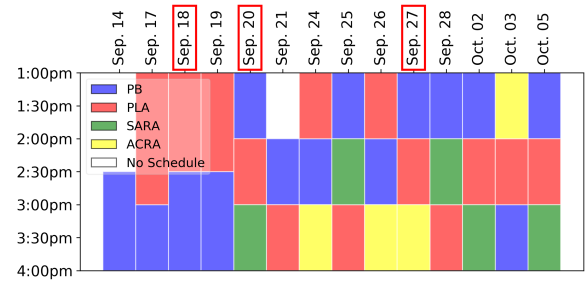
For each day of the experiment, the following procedure was followed:

1. Randomly schedule the different agent conditions into 1 or 1.5 hour time slots as shown in Figure 5.1. PB and PLA were scheduled on each day, while only one of SARA and ACRA were scheduled per day.

2. Automatically run scheduled behaviour at each time slot, and save interaction data and learned models and videos at the end of each behaviour.

During days where no visitor surveys were collected, researchers were not present in the environment. During the three survey days, researchers were present, but did not provide any additional instructions to visitors. Researchers observed which visitors interacted with the LAS, passively or actively, within a specific behaviour period. When visitors were finished with their visit, researchers unobtrusively approached visitors who had interacted with the system, and asked them if they were willing to participate in a survey. If a visitor agreed to do the survey, they were guided to a table located around a corner and were provided with a tablet with a questionnaire (see Section 5.4). The researchers also recorded which mode the visitor had interacted with. We only recruited visitors who had interacted with only one behaviour mode.

The overall experiment schedule is shown in Figure 5.1, where red, blue, green, yellow and white areas correspond to PB, PLA, SARA, ACRA and no schedule respectively. A summary of the experiment schedule and collected data is shown in Table 5.1.



Video was not available on Sep. 14.

Figure 5.1: Experiment Schedule

Table 5.1: Summary of Experiment Schedule and Data

Behaviour	Days	Hours	Survey Participants
PB	14	15.5	14
PLA	13	15	15
SARA	4	4	4
ACRA	4	4	3

Video was not available on Sep. 14. The Data for SARA on Sep. 20 was corrupted.

5.4 Data Collection

The data collected comes in four types: sensor readings, learning agent logs, human survey data and video data from the two web-cams.

Every raw sensor reading is logged. In addition to the raw sensor data, each agent also logs its own learning algorithm data collected during the course of learning.

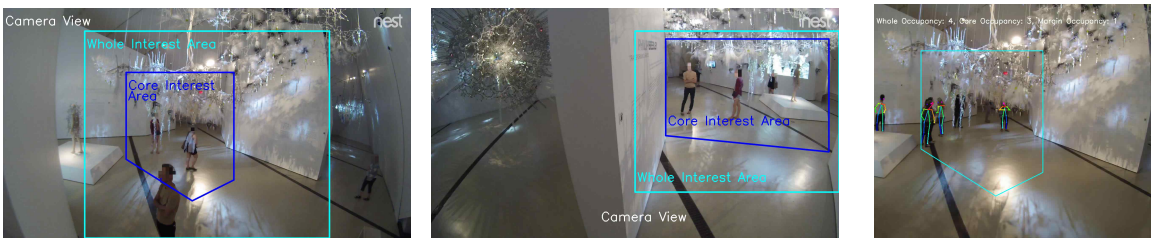
For human survey data, 14, 15, 4, and 3 participants completed surveys in the PB, PLA, SARA and ACRA modes respectively, as summarized in Table 5.1. The questionnaire used in our experiment is a standardized measurement tool for HRI: the Godspeed questionnaire [61]. In addition to the 24 Godspeed questions, we asked participants about their interests and background, and their general feedback and comments. The Questionnaire consists of four types of questions:

1. Participants' interests and background (multiple-select multiple choice);
2. Participants prior knowledge about interactive architecture and machine learning, including "How familiar are you with interactive architecture?" and "How familiar are you with machine learning algorithms?";
3. 24 Godspeed questions namely Godspeed I: Anthropomorphism, Godspeed II: Animacy, Godspeed III: Likeability, Godspeed IV: Perceived Intelligence and Godspeed V: Perceived Safety [61];
4. Participants' general feedback, i.e., "Any additional comments regarding your experience?" and "Any overall feedback?".

Video data is collected to calibrate sensory readings and validate occupancy estimates, which will be discussed in detail in Section 5.5. Video data is available for all the experiments except for Sep. 14.

5.5 Data Analysis

The camera view includes regions outside of the LAS itself. To only focus on areas directly related to the LAS, we define three parts of the whole camera view (as shown in Figure 5.2a and Figure 5.2b for Camera1 and Camera2 respectively). In Figure 5.2, each camera view is divided into Camera View, Whole Interest Area and Core Interest Area. For both IR Data Calibration and Occupancy Estimation, we only consider the Whole Interest Area. Any visitors outside this interest area will be ignored for the purposes of occupancy estimation. The Core Interest Area approximates the space directly underneath the LAS.



(a) Camera1 (b) Camera2 (c) Sample of Estimated Occupancy from Camera1

Figure 5.2: Interest Area Used to Estimate Occupancy

5.5.1 IR Data Calibration

To enable comparison between different behaviour modes, the sensor data must be preprocessed to ensure consistency between conditions. Since visitors can physically interact with the system, it is possible that a visitor changes the direction of the IR sensor thus changing its field of view and subsequent readings. To calibrate the IR data, two calibration steps are taken: 1) IR sensors, whose value is relatively constant and effectively not responding to occupants (e.g., due to obstructions or height being greater than the sensor range), are removed, 2) the baseline reading for each sensor is shifted to zero. Note

that the calibration is only done for analysis, during the learning uncalibrated readings are used. To identify blocked IR and baseline shifts, we visually checked the videos recorded by the two web-cams, and selected a time period when there is no visitor within the whole interest area. Then, we find the IR data corresponding to the no-visitor time. Using the no-visitor time, we determine the thresholds for noise removal and blocked IR detection for the IR data. We use these thresholds to calibrate the raw IR data.

5.5.2 Occupancy Estimation

We also use the camera data to generate a second estimate of occupancy. We estimate the number of people occupying the space during a one minute interval, using OpenPose[1] [62] based on the videos recorded by one of the web-cams[2]. When estimating occupancy, we only considered the Whole Interest Area.

5.5.3 Non-visitor Period Examination

We also used the camera data to determine whether there are significant periods when no visitors are present. To identify the time periods with no visitors in Whole Interest Area, we manually labelled the time periods when no person is under either camera in the Whole Interest Area. If a person's body is partially visible in Whole Interest Area, we consider it as a person being in the area. The total amount of non-visitor time throughout the experiment is 1 hour, only 2.5% of total experiment time. Therefore, we use the whole time period for analysis without removing any non-visitor intervals.

5.6 Evaluation Metrics

For quantitative comparison, instead of using accumulative reward commonly seen in RL analysis, we consider average estimated engagement and average active interaction as two evaluation metrics, considering the complexity of the environment which is non-episodic and non-stationary. In the natural setting of LAS, the number of visitors varies at different time periods and is highly irregular, which makes the evaluation based on periodic accumulated reward unfeasible since this is a non-episodic environment. Besides, manually

[1] https://github.com/CMU-Perceptual-Computing-Lab/openpose

[2] Videos from Camera2 are highly affected by the changing light of the projector as shown in Figure 5.2b, so for occupancy estimation we only used videos from Camera1.

setting the maximum length of an evaluation period and comparing accumulated reward in that period is also unfair, because the total number of visitors in space is time-varying and maximum available rewards in episodes are also time-related. Therefore, we choose to regard the whole experiment as a continuous learning and evaluate in terms of average estimated engagement and average active interaction.

Both estimated engagement level and active interaction count are based on IR readings, but they emphasize different aspects of engagement. Specifically, estimated engagement level does not differentiate between passive and active interaction (illustrated in Figure 3.1), while active interaction count focuses on measuring active interaction.

5.6.1 Estimated Engagement Level

The observation vectors used for training PLA, SARA and ACRA are different (see Section 3.2), so the estimated engagement, i.e., reward used for training each behaviour, cannot be directly compared. Therefore, we use raw IR readings recorded during each behaviour and Equation 5.1 to calculate an estimated engagement for comparison among behaviours. Specifically, given M IR readings received within 1 minute (typically sampled at 10Hz) $\{\boldsymbol{ir}^{(1)}, \boldsymbol{ir}^{(2)}, ..., \boldsymbol{ir}^{(M)}\}$ where each IR reading $\boldsymbol{ir}^{(i)}$ is a vector of 24 IR values, the estimated engagement level e is defined by Equation 5.1:

$$e = \frac{1}{M}\frac{1}{24}\sum_{m=1}^{M}\sum_{i=1}^{24} ir_i^{(m)} \tag{5.1}$$

where $ir_i^{(m)}$ is the ith IR sensor in the mth IR reading. The estimated engagement is in the range [0,1], where the maximum 1 corresponds to a maximally engaging state, where all IR sensors are receiving maximum readings during the entire 1 minute window, while the minimum 0 corresponds to fully non-engaging state, where all IR sensors are receiving minimum readings for the duration of the one-minute window.

5.6.2 Active Interaction Count Analysis

In addition to the estimate of engagement, we separately estimate the level of active interaction. To capture active interactions, we count the number of IR readings having value >= 0.25, which corresponds to a proximity of 35cm or less from an IR sensor, within 1 minute. Despite the behavioral difference among IR sensors caused in manufacturing

and installation[3], we assume all sensors behave in the same way for simplicity. Formally, given M IR readings received within 1 minute (typically sampled at frequency $F = 10Hz$) $\left\{\boldsymbol{ir}^{(1)}, \boldsymbol{ir}^{(2)}, ..., \boldsymbol{ir}^{(M)}\right\}$ where each IR reading $\boldsymbol{ir}^{(i)}$ is a vector of 24 IR values, the number of active interactions N_{active} is defined by Equation 5.2:

$$N_{active} = \frac{1}{F} \sum_{m=1}^{M} \sum_{i=1}^{24} \mathbb{1}\left\{ir_i^{(m)} \geq 0.25\right\} \tag{5.2}$$

where $ir_i^{(m)}$ is the ith IR sensor in the mth IR reading, and $\mathbb{1}\{\cdot\}$ is a indicator function. Therefore, N_{active} is the total detected active interactions within 1 minute.

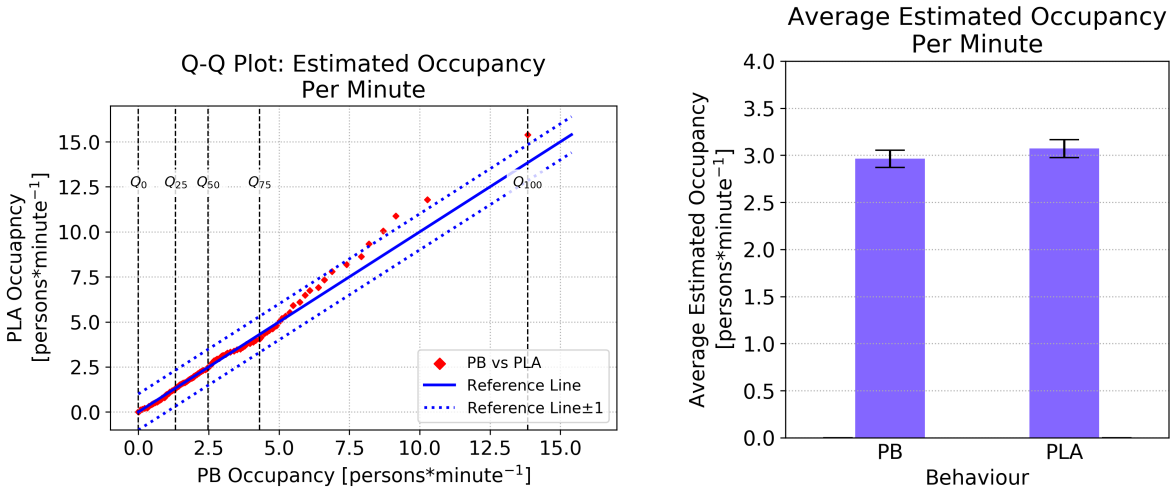
5.7 Results

5.7.1 Quantitative Comparison Between PB and PLA

In this section, we quantitatively compare the performance of the two behaviour modes based on sensory data collected during the interaction between visitors and the LAS. We use two ways to quantitatively compare the four behaviours' performance: 1) comparing the estimated engagement level, as described in Section 5.6.1, and 2) comparing the number of active interactions, as introduced in Section 5.6.2.

Our experiment is run in a natural setting, i.e., a publicly accessible museum, so it is possible that there are different occupancy levels in the space due to factors not related to the behaviour mode. To check whether there are different occupancy levels between conditions (which might be caused either by some behaviours being more attractive to visitors, or factors not related to system behaviours), we analyze the overall occupancy level for PB and PLA, as described in Section 5.5.2. Figure 5.3 shows a comparison of the estimated occupancy between PB and PLA, where (a) shows that, in only about 5% of data, PLA has approximately 1 more visitor than PB, and (b) shows that the average occupancy between PB and PLA is very similar. A Mann-Whitney U test indicates that there is no significant difference between PB and PLA in terms of occupancy level, $U = 239030.5$, $p = 0.92$ (two-sided).

[3]The heights of sensors in the Aegis installation are not identical, thus there are differences in the distance of detected activities.



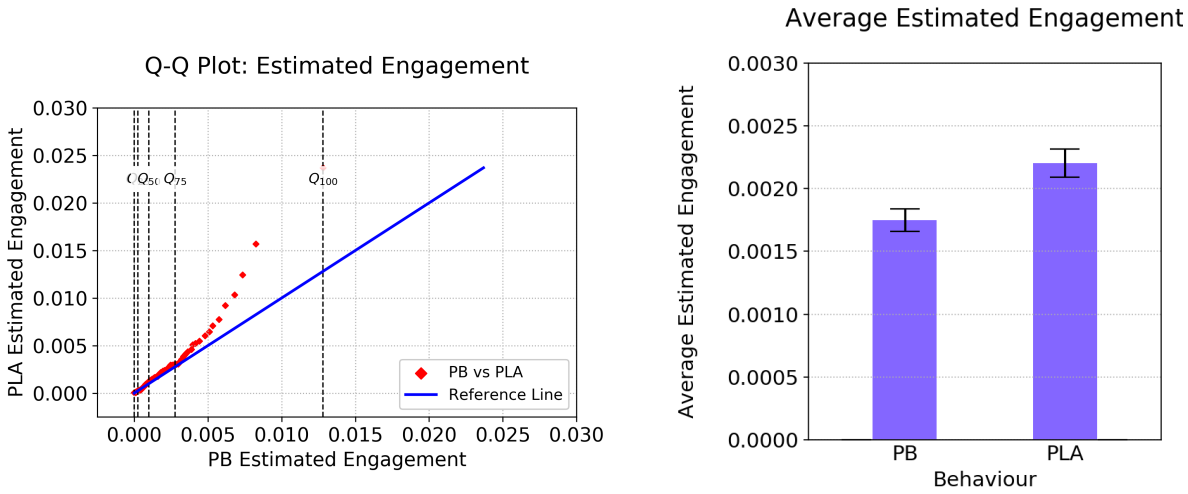
(a) Comparison of Estimated Occupancy Distributions.

(b) Average Estimated Occupancy

Figure 5.3: Estimated Occupancy Comparison. (a) is a Q-Q (100-quantiles-100-quantiles) plot of estimated per-minute occupancy, using the method introduced in Section 5.5.2, where the coordinate (x, y) of the q-th point from bottom-left to up-right corresponds to the estimated occupancy of (PB, PLA) for the q-th percentile, i.e. $Q_q, q = 0, 1, ..., 100$, and the reference line indicates a perfect match of distribution between PB and PLA. For example, the point $(4.3, 4)$ for PB vs PLA at the Q_{75} means that 75% of observations for PB and PLA are less than 4.3 and 4, respectively. (b) shows the average estimated per-minute occupancy and its standard error for PB and PLA.

Estimated Engagement Level Comparison

Figure 5.4 compares the estimated engagement (defined in Section 5.6.1) between PB and PLA. From Figure 5.4a, we can observe that for the first 75% of data there is no noticeable difference between PB and PLA, while for the last 25% of data PLA has larger estimated engagement than PB. Figure 5.4b shows the average estimated engagement and average estimated engagement, in which PLA achieved higher rewards than PB.



(a) Estimated Engagement

(b) Average Estimated Engagement

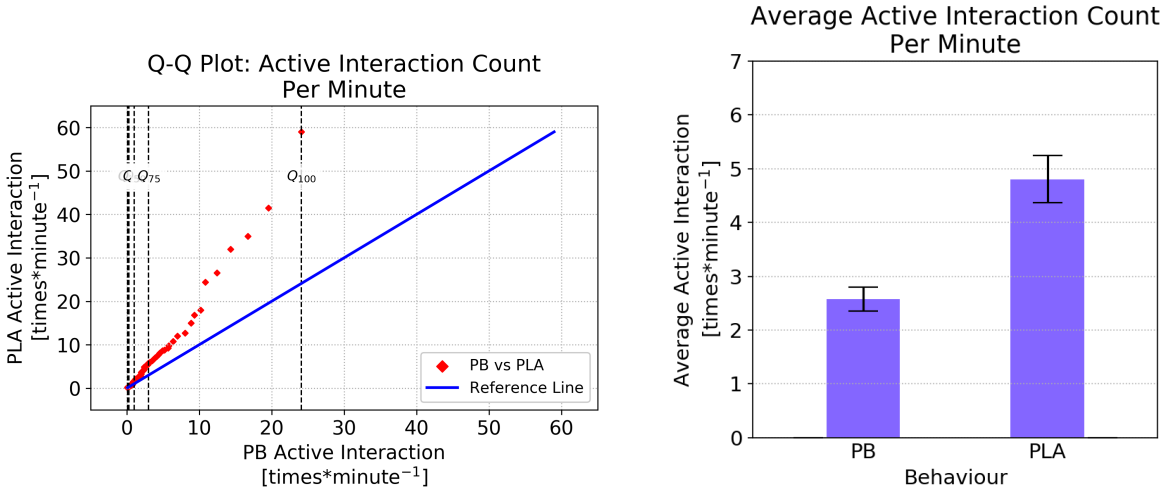
Figure 5.4: Estimated Engagement Comparison (a) is a Q-Q (100-quantiles-100-quantiles) plot between PB and PLA based on average estimated engagement, and the reference line represents a perfect match of distributions between PB and PLA. (b) compares the average estimated engagement, where blue bars with standard errors show the average estimated engagement and its corresponding standard error.

Active Interaction Comparison

Figure 5.5 compares the active interaction count based on Eq. 5.2. From Figure 5.5a, we can see that for about 50% of observations, PLA achieves higher active interaction than PB. Figure 5.5b compares PB against PLA in terms of average active interaction count. As shown in these results, PLA almost doubles the PB average active interaction count.

Daily Average Estimated Engagement and Active Interaction

To analyse how performance evolved over the 3 week experiment, we plot the daily average engagement and active interaction over the whole experiment. Figure 5.6 shows daily average metrics of PB and PLA. From the regression lines in Figure 5.6a, we can see that at the first couple of days PB outperforms PLA, while after Sep. 25 PLA overtakes PB for the rest of time. However, daily average active interaction shown in Figure 5.6b shows different pattern that PLA receives more active interaction than PB at the very

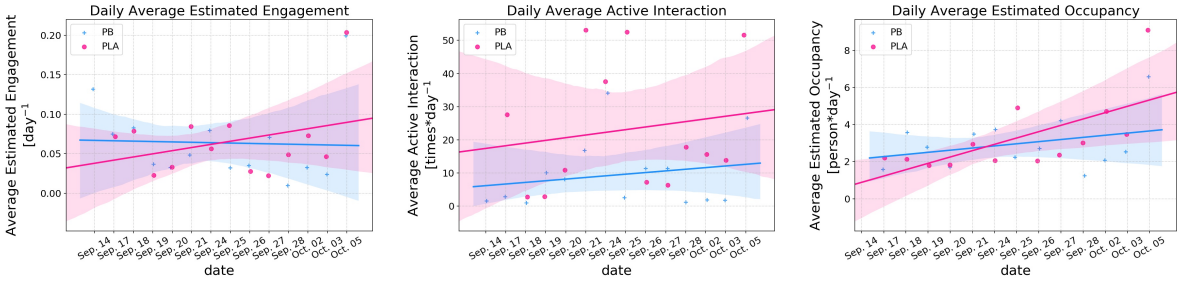


(a) Comparison of Average Active Interaction Distributions

(b) Average Active Interaction Count

Figure 5.5: Active Interaction Count Comparison. (a) is the Q-Q (100-quantiles-100-quantiles) plot on active interaction count per minute obtained using Eq. 5.2. The reference line indicates a perfect match of distribution between PB and PLA. (b) compares the average active interaction count per minute between PB and PLA.

beginning and keeps expanding the gap between PLA and PB. Even though it seems PLA is evolving, we are not clear if this is caused by variation of number of visitors or by continuous adapting of PLA, because the daily average estimated occupancy in Figure 5.6c also continuous increase over the whole experiment and the increase of estimated occupancy could be caused by more engaging behaviour of PLA or independent from the interactive action of the LAS.



(a) Estimated Engagement (b) Active Interaction (c) Estimated Occupancy

Figure 5.6: Trajectory of Daily Average Metrics. (a) Daily Average Estimated Engagement, (b) Daily Average Active Interaction and (c) Daily Average Estimated Occupancy, where each data point is the corresponding average on each day, and the lines are the linear regression of these data and the translucent bands around the regression line are the 95% confidence interval for the regression estimate.

5.7.2 Human Survey Results

In this section, we analyze the visitor responses to the survey for PB and PLA, and omit analysis of SARA and ACRA due to the very limited number of participants (see Table 5.1). We first examine if there are any differences in the population characteristics between the participants who engaged with the system in PB or PLA behaviour modes. Then, we compare the PB and PLA responses for each Godspeed category. Finally, we compare PB and PLA for each Godspeed question individually, comparing the number of participants providing a rating of five.

We first analyze whether there are population differences between conditions. In Section 5.5.2, we confirmed that there were no significant differences between PB and PLA in terms of estimated occupancy. To test for differences in participant background and interest, we performed a χ^2-test on participants' background and interests based on the first two questions in our questionnaire (see Section 5.4), and found no statistically significant differences between the two groups.

Cronbach's α test was conducted on each category of Goodspeed for both PB and PLA to examine the reliability of participants' responses, results are shown in Table 5.2. As the results shown, although α on Anthropomorphism and Perceived Safety is questionable or unacceptable, α on others are all acceptable, especially for Likeability $\alpha \geq 0.85$.

Table 5.2: Cronbach's α on Goodspeed for PB and PLA

	Anthropomorphism	**Animacy**	**Likeability**	**Perceived Intelligence**	**Perceived Safety**
PB	0.74	0.77	0.85	0.89	0.52
PLA	0.64	0.80	0.93	0.85	0.27

A commonly accepted rule[63]: $0.9 \leq \alpha$: Excellent; $0.8 \leq \alpha < 0.9$: Good; $0.7 \leq \alpha < 0.8$: Acceptable; $0.6 \leq \alpha < 0.7$: Questionable; $0.5 \leq \alpha < 0.6$: Poor; $\alpha < 0.5$: Unacceptable.

Figure 5.7 shows the Box-plot and Violin-plot of the calculated average grade over each Godspeed category for PB and PLA. Within the five Godspeed categories, only *Likeability* has a relatively large gap between the medians of PB and PLA. In addition, Likeability has a relatively small variance, whereas other categories have large variance. A t-test on the average grade shows that PLA is rated higher than PB for Likeability with 95% confidence, whereas for other categories there is no significant difference between PB and PLA. A normality test was conducted for the Likeability category from PB and PLA respectively. Shapiro-Wilk Test [64] indicates PB ($p = 0.12$) is normally distributed and PLA ($p = 0.0008$) is not. Therefore, for clarity a histogram for PB and PLA on Likeability

is shown in Figure 5.8, from which we can see that more participants from PLA rated grade 5 than that from PB.

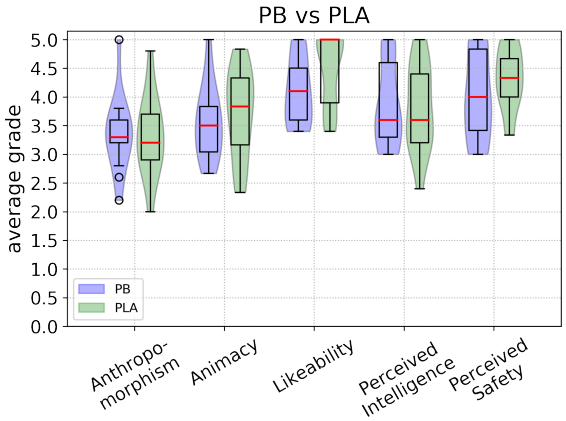


Figure 5.7: Boxplot and Violinplot of Average Grade of each Godspeed Category over Participants within PB or PLA.

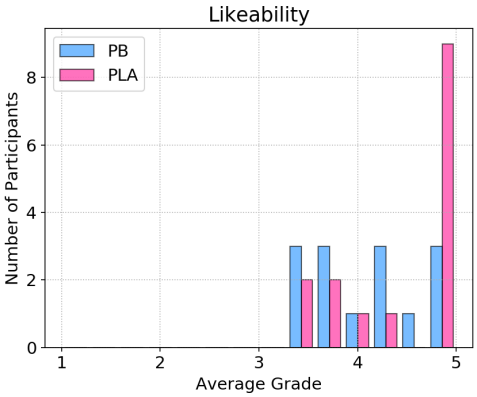


Figure 5.8: Histogram of Participants Average Grade over Questions in Likeability. The grade range [1,5] is uniformly divided into 15 binds

Figure 5.9 shows the proportion of participants who rated each question = 5. The reason for choosing "5" for comparing participants' responses is that participants for both PB and PLA were very generous, and gave grades higher than or equal to 3 for most questions (e.g. over 92% of all responses to questions in Likeability were 3 or higher). From this figure, we can see that there are significant differences between PB and PLA in questions related to *Godspeed III-Likeability*, whereas for most of the questions in the other four Godspeed categories, there are no statistically significant differences. Specifically, for questions *Inert-Interactive*, *Unfriendly-Friendly*, *Unkind-Kind*, *Unpleasant-Pleasant* and *Awful-Nice* PLA is better than PB with a high confidence >90%, while for the other questions there are no statistically significant differences.

In summary, PLA is rated higher than PB by the participants in terms of Likeability and interactivity, while there are no significant differences between PB and PLA in the other Godspeed categories.

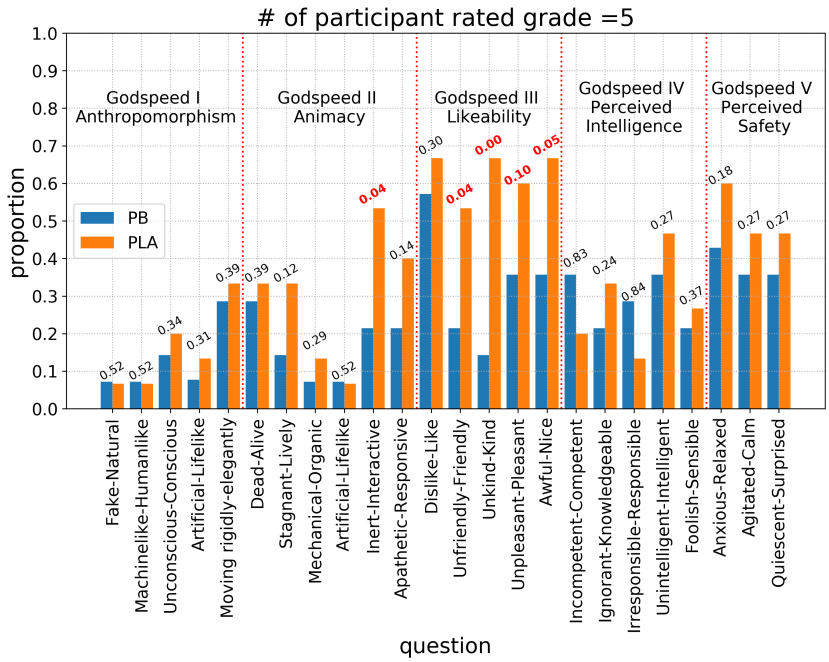


Figure 5.9: Proportion of Participants who Rated with Grade=5. The value above the bars for each question is the p value of z-test with alternative hypothesis $p_{PB} < p_{PLA}$, where p_{PB} and p_{PLA} are the proportion of participants who have a = 5 rating among all participants within PB and PLA respectively.

5.8 Conclusions

In this chapter, we evaluate algorithms for generating interactive behaviours through a field experiment in a natural setting. We analyze the interactions between human and LAS and collect human survey data. By using two evaluation metrics, we show that a learning agent acting on parameterized space, i.e. PLA, has higher average engagement level than pre-scripted behaviour. The human survey data shows that PLA is better rated than PB in terms of Likeability and interactivity.

Chapter 6

Conclusions and Future Work

6.1 Conclusions and Contributions

In this thesis, we proposed an approach to autonomously generate interactive behaviours for human-robot interaction. The work was developed for an interactive art installation developed by LASG/PBAI and exhibited at the ROM. An RL algorithm is used to learn in a parameterized action space which exploits human designers' knowledge. Simulations were performed to validate the learning algorithm in a simplified setting. Then, a field study was conducted in a natural setting, where no constraints were imposed on visitors and group interaction was accommodated.

In the simulation, we compared three behaviour generation modes, PLA, SARA and Random, in the single and multiple visitor environment. Two exploration methods, parameter noise and action noise, were also examined. The result shows that a small action space (PLA) could benefit from using parameter noise for exploration, but in a large action space (SARA), it slows down the learning speed. This effect becomes more noticeable when the number of visitors in the space increases. Meanwhile, when the visitor preference is a subset of adaptive behaviour, the learning algorithm has better performance. This effect is also more prominent in the multiple visitor environment. We also investigated the influence of the visitor's stochasticity on learning performance, and find that the learning algorithm becomes less effective as the stochasticity in simulated visitor behaviour increases.

In the field study, we developed and evaluated algorithms for generating interactive behaviours with the LAS. Specifically, we provided a way to estimate engagement during group interaction based on low level sensing, i.e. IR sensors. This might be helpful for designing other large-scale interactive systems.

PB and PLA were examined to evaluate how the use of human knowledge influences the social interactions between the LAS and human visitors. We showed that PLA has higher engagement level and active interaction count than PB. The human survey data showed that participants gave higher ratings in Likeability and interactivity for PLA. It is hypothesized that the PLA configuration outperforms PB because it benefits from both human expert input, such as parameterized action space and manual reward function, and learning. Our results illustrate how human expertise and autonomous learning may be combined by using a reinforcement learning agent learning from estimated engagement of group interactions.

6.2 Limitations

Even though we created several test scenarios, the simulation is still much simpler than real life. People are attracted to different stimuli in various forms, such as sound, visual effects and complex combinations of stimuli. The interests will also die out and change as participants become more and more familiar with the system. In the simulation, we assume the visitors have simple attraction models which remain unchanged during the training process. In real life, different individuals may have different interaction preferences. They do not necessarily walk towards the attractions immediately, nor necessarily figure out instantly how to interact with the system, i.e. waving at IR sensors. In the simulation, we assume visitors are capable of interacting with the LAS system efficiently by placing them in front of IR sensors.

The field study is conducted in a natural setting, where basic assumptions of a stationary environment in reinforcement learning are violated. The response time from human visitors is not constant and sets an upper bound for the interaction frequency, thus limiting the number of experiences that the learning agent can learn from. Therefore, although we exploit a RL framework in our work, the role it plays is different from that of standard testbeds such as OpenAI Gym[10]. In this work, RL is used to introduce adaptability, but there is no guarantee that the learning leads to optimal policy. In fact, we do not observe a fixed policy learnt by the agent, but instead find the policy to be constantly evolving. We hypothesize that the agent is adapting to the new environment.

As the field study is non-repeatable, we could not have runs with identical settings in order to choose optimal hyperparameters. The choice of the hyperparmeters is completely empirical even though they are validated in the simulation. Meanwhile, we used DDPG because of its wide applications. There are other advanced algorithms for the continuous

action space such as PPO[25] and TD3[65], and we did not have the chance to compare their performances.

Due to the varying lighting condition in the exhibition area, we could not use the webcam footage for the pose or the facial expression analysis. Compared to other social HRI work with rich sensing such as cameras and microphones, we were still able to estimate engagement with limited sensing and generate engaging behaviours accordingly. This might be helpful to other large scale interactive systems where having sophisticated measurement may be unfeasible.

6.3 Future Work

In this work, we proposed a learning agent that learns on one specific implementation of human designed interactive behaviours. Other human designed behaviours should be examined and compared to see how the change in human knowledge representation affects the learning performance. In addition, hierarchical RL with PB bootstrapping could be a promising extension, where we could design a pool of PBs and various levels of reward functions, and see how complicated action patterns could emerge.

In the field study, even though PLA received higher average engagement and perceived likeability than PB, we cannot be certain about the cause of this difference. Therefore, a baseline with random policy can be tested to see if there is a difference between this baseline and PLA to confirm that the learning agent is indeed learning from or adapting to its interaction. Other advanced continuous control RL algorithms such as PPO and TD3 are also worth investigating. To tackle the low pace of interaction in LAS and high sample requirement of RL, we can also investigate how to bootstrap the learning by transferring learnt models in the simulation to the physical LAS.

Like previous work by Chan et al.[28], we could also introduce intrinsic motivation and a learning algorithm driven both intrinsically and extrinsically for LAS. This would require more sophisticated sensing in the LAS.

References

[1] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[2] Nicola Pedrocchi, Federico Vicentini, Malosio Matteo, and Lorenzo Molinari Tosatti. Safe human-robot cooperation in an industrial environment. *International Journal of Advanced Robotic Systems*, 10(1):27, 2013.

[3] George Michalos, Sotiris Makris, Panagiota Tsarouchi, Toni Guasch, Dimitris Kontovrakis, and George Chryssolouris. Design considerations for safe human-robot collaborative workplaces. *Procedia CIrP*, 37:248–253, 2015.

[4] Allison M Okamura. Methods for haptic feedback in teleoperated robot-assisted surgery. *Industrial Robot: An International Journal*, 31(6):499–508, 2004.

[5] S Jezernik, R Schärer, G Colombo, and M Morari. Adaptive robotic rehabilitation of locomotion: a clinical study in spinally injured individuals. *Spinal cord*, 41(12):657, 2003.

[6] Peter R Culmer, Andrew E Jackson, Sophie Makower, Robert Richardson, J Alastair Cozens, Martin C Levesley, and Bipin B Bhakta. A control strategy for upper limb robotic rehabilitation with a dual robot system. *IEEE/ASME Transactions on Mechatronics*, 15(4):575–585, 2009.

[7] Kai Keng Ang, Karen Sui Geok Chua, Kok Soon Phua, Chuanchu Wang, Zheng Yang Chin, Christopher Wee Keong Kuah, Wilson Low, and Cuntai Guan. A randomized controlled trial of eeg-based motor imagery brain-computer interface robotic rehabilitation for stroke. *Clinical EEG and neuroscience*, 46(4):310–320, 2015.

[8] Michael A. Goodrich and Alan C. Schultz. HumanRobot Interaction: A Survey. *Foundations and Trends in HumanComputer Interaction*, 1(3):203–275, 2008.

[9] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

[10] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[11] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[12] Rémi Barraquand and James L Crowley. Learning polite behavior with situation models. In *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, pages 209–216. ACM, 2008.

[13] Andrea Lockerd Thomaz, Guy Hoffman, and Cynthia Breazeal. Real-time interactive reinforcement learning for robots. In *AAAI 2005 workshop on human comprehensible machine learning*, 2005.

[14] Andrea Lockerd Thomaz, Cynthia Breazeal, et al. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *Aaai*, volume 6, pages 1000–1005. Boston, MA, 2006.

[15] Ioannis Papaioannou, Christian Dondrup, Jekaterina Novikova, and Oliver Lemon. Hybrid chat and task dialogue for more engaging hri using reinforcement learning. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 593–598. IEEE, 2017.

[16] Ioannis Papaioannou and Oliver Lemon. Combining chat and task-based multimodal dialogue for more engaging hri: A scalable method using reinforcement learning. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 365–366. ACM, 2017.

[17] Goren Gordon, Samuel Spaulding, Jacqueline Kory Westlund, Jin Joo Lee, Luke Plummer, Marayna Martinez, Madhurima Das, and Cynthia Breazeal. Affective personalization of a social robot tutor for childrens second language skills. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[18] Klaus Weber, Hannes Ritschel, Ilhan Aslan, Florian Lingenfelser, and Elisabeth André. How to shape the humor of a robot-social behavior adaptation based on

reinforcement learning. In *Proceedings of the 2018 on International Conference on Multimodal Interaction*, pages 154–162. ACM, 2018.

[19] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[20] Volodymyr Mnih, Adri Puigdomnech Badia, Mehdi Mirza, Alex Graves, Tim Harley, Timothy P. Lillicrap, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML'16 Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, pages 1928–1937, 2016.

[21] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. 2014.

[22] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[23] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. 2015.

[24] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015.

[25] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[26] Kazumi Kumagai, Ikuo Mizuuchi, Lingheng Meng, Alexandru Blidaru, Philip Beesley, and Dana Kulić. Towards individualized affective human-machine interaction. In *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 678–685. IEEE, 2018.

[27] Noriaki Mitsunaga, Christian Smith, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. Adapting robot behavior for human–robot interaction. *IEEE Transactions on Robotics*, 24(4):911–916, 2008.

[28] Matthew TK Chan, Rob Gorbet, Philip Beesley, and Dana Kulič. Curiosity-based learning algorithm for distributed interactive sculptural systems. In *2015 IEEE/RSJ*

International Conference on Intelligent Robots and Systems (IROS), pages 3435–3441. IEEE, 2015.

[29] Matthew TK Chan, Rob Gorbet, Philip Beesley, and Dana Kulić. Interacting with curious agents: User experience with interactive sculptural systems. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 151–158. IEEE, 2016.

[30] Pierre-Yves Oudeyer. Intelligent adaptive curiosity: a source of self-development. 2004.

[31] Pierre-Yves Oudeyer, Frdric Kaplan, and Verena V Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation*, 11(2):265–286, 2007.

[32] Hannes Ritschel, Tobias Baur, and Elisabeth André. Adapting a robot's linguistic style based on socially-aware reinforcement learning. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 378–384. IEEE, 2017.

[33] Hannes Ritschel. Socially-aware reinforcement learning for personalized human-robot interaction. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1775–1777. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[34] Iolanda Leite, André Pereira, Ginevra Castellano, Samuel Mascarenhas, Carlos Martinho, and Ana Paiva. Modelling empathy in social robotic companions. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 135–147. Springer, 2011.

[35] Changchun Liu, Karla Conn, Nilanjan Sarkar, and Wendy Stone. Online affect detection and robot behavior adaptation for intervention of children with autism. *IEEE transactions on robotics*, 24(4):883–896, 2008.

[36] Halit Bener Suay and Sonia Chernova. Effect of human guidance and state space size on interactive reinforcement learning. In *2011 Ro-Man*, pages 1–6. IEEE, 2011.

[37] Ernest Edmonds, Greg Turner, and Linda Candy. Approaches to interactive art systems. In *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 113–117. ACM, 2004.

[38] Maria Karam, Carmen Branje, Gabe Nespoli, Norma Thompson, Frank A. Russo, and Deborah I. Fels. The emoti-chair: An interactive tactile music exhibit. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, pages 3069–3074, New York, NY, USA, 2010. ACM.

[39] Jeffrey E Boyd, Gerald Hushlak, and Christian J Jacob. Swarmart: interactive art from swarm intelligence. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 628–635. ACM, 2004.

[40] Bruce Wands. Variations: An interactive musical sculpture. In *Proceedings of the 5th Conference on Creativity &Amp; Cognition*, C&C '05, pages 306–309, New York, NY, USA, 2005. ACM.

[41] David St-Onge and Nicolas Reeves. Human interaction with flying cubic automata. In *Proceedings of 2010 IEEE/ACM Internations Conference on Human Robots Interaction.*, 2010.

[42] Philip Beesley and Christine Macy. *Kinetic architectures & geotextile installations.* Riverside Architectural Press Toronto, 2010.

[43] Hayley Isaacs. *Hylozoic Ground: Liminal Responsive Architecture: Philip Beesley.* Riverside Architectural Press, 2010.

[44] Philip Beesley, Matthew Chan, Rob Gorbet, Dana Kulic, and Mo Memarian. Evolving systems within immersive architectural environments: New research by the living architecture systems group. *Next Generation Building*, 2(1):31–56, 2015.

[45] Mark Scheeff, John Pinto, Kris Rahardja, Scott Snibbe, and Robert Tow. Experiences with sparky, a social robot. In *Socially intelligent agents*, pages 173–180. Springer, 2002.

[46] Wolfram Burgard, Armin B Cremers, Dieter Fox, Dirk Hähnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. Experiences with an interactive museum tour-guide robot. *Artificial intelligence*, 114(1-2):3–55, 1999.

[47] AW Moore and CG Atkeson. Memory-based function approximators for learning control. *Manuscript submitted for publication*, 1992.

[48] Sebastian Thrun, Maren Bennewitz, Wolfram Burgard, Armin B Cremers, Frank Dellaert, Dieter Fox, Dirk Hahnel, Charles Rosenberg, Nicholas Roy, Jamieson Schulte, et al. Minerva: A second-generation museum tour-guide robot. In *Proceedings 1999*

IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C), volume 3. IEEE, 1999.

[49] Mikael Svenstrup, Thomas Bak, Ouri Maler, Hans Jørgen Andersen, and Ole B Jensen. Pilot study of person robot interaction in a public transit space. In *International Conference on Research and Education in Robotics*, pages 96–106. Springer, 2008.

[50] Astrid Weiss, Nicole Mirnig, Ulrike Bruckenberger, Ewald Strasser, Manfred Tscheligi, Barbara Kühnlenz, Dirk Wollherr, and Bartlomiej Stanczyk. The interactive urban robot: user-centered development and final field trial of a direction requesting robot. *Paladyn, Journal of Behavioral Robotics*, 6(1), 2015.

[51] Masahiro Shiomi, Daisuke Sakamoto, Takayuki Kanda, Carlos Toshinori Ishi, Hiroshi Ishiguro, and Norihiro Hagita. A semi-autonomous communication robota field trial at a train station. In *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 303–310. IEEE, 2008.

[52] Yamato Iwamura, Masahiro Shiomi, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. Do elderly people prefer a conversational humanoid as a shopping assistant partner in supermarkets? In *Proceedings of the 6th international conference on Human-robot interaction*, pages 449–456. ACM, 2011.

[53] Takayuki Kanda, Takayuki Hirano, Daniel Eaton, and Hiroshi Ishiguro. Interactive robots as social partners and peer tutors for children: A field trial. *Human–Computer Interaction*, 19(1-2):61–84, 2004.

[54] Lingheng Meng, Daiwei Lin, Adam Francey, Rob Gorbet, Philip Beesley, and Dana Kulic. Learning to engage with interactive systems: A field study. *CoRR*, abs/1904.06764, 2019.

[55] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning*, pages 1329–1338, 2016.

[56] Arthur Juliani, Vincent-Pierre Berges, Esh Vckay, Yuan Gao, Hunter Henry, Marwan Mattar, and Danny Lange. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627*, 2018.

[57] Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. Openai baselines. https://github.com/openai/baselines, 2017.

[58] Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y. Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. *CoRR*, abs/1706.01905, 2017.

[59] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pages 278–287, 1999.

[60] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *CoRR*, abs/1706.02275, 2017.

[61] Christoph Bartneck, Dana Kulić, Elizabeth Croft, and Susana Zoghbi. Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. *International journal of social robotics*, 1(1):71–81, Jan 2009.

[62] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *CoRR*, abs/1611.08050, 2016.

[63] Robert F DeVellis. *Scale development: Theory and applications*, volume 26. Sage publications, 2016.

[64] Samuel Sanford Shapiro and Martin B Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611, 1965.

[65] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *ICML 2018: Thirty-fifth International Conference on Machine Learning*, volume 80, pages 1587–1596, 2018.

[66] Hado V Hasselt. Double q-learning. In *Advances in Neural Information Processing Systems*, pages 2613–2621, 2010.

APPENDICES

Appendix A

DDPG Algorithm Implementation

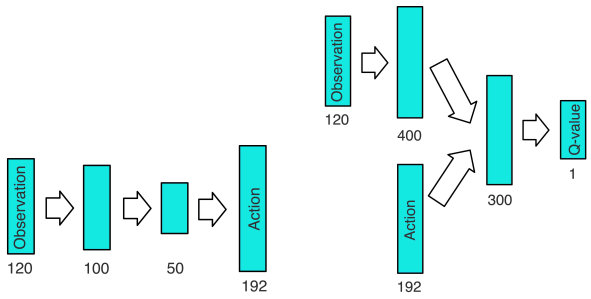
Initialize policy paramters θ, Q function parameters ϕ, empty replay buffer $\mathcal{D}$
Set target parameters equal to main parameters $\theta_{targ} \leftarrow \theta$, $\phi_{targ} \leftarrow \phi$
while *True* **do**
  Observe state s and select action $a = \pi_\theta(s) + \mathcal{N}$
  Execute a in the environment
  Observe next state, reward and done signal, s', r and d
  Store experience (s, a, r, s', d) in buffer $\mathcal{D}$
  if *Reach maximum steps per episode* **then**
    for *fixed number of updates* **do**
      Sample a batch $\mathcal{B}$ from $\mathcal{D}$ and compute target y:
      $y(r, s') = r + \gamma Q_{\theta_{targ}}(s', \pi_{\theta_{targ}}(s'))$
      Update the critic by one step of gradient descent:
      $\nabla_\phi \frac{1}{|B|} \sum_{(s,a,r,s',d) \in \mathcal{B}} \left(Q_\phi(s, a) - y\left(r, s', d\right)\right)^2$
      Update the policy:
      $\nabla_\theta \frac{1}{|B|} \sum_{s \in B} Q_\phi\left(s, \mu_\theta(s)\right)$
      Update target networks:
      $\theta_{target} \leftarrow \rho\theta_{target} + (1 - \rho)\theta$
      $\phi_{target} \leftarrow \rho\phi_{target} + (1 - \rho)\phi$
    end
  end
end

Algorithm 2: DDPG algorithm with action noise as the exploration method.

Appendix B

Implementation Details of SARA

The neural network structure for SARA's actor-critic agent is shown in Figure. B.1, where the number under each layer is the neural units in that layer. The neural network is fully-connected with layer-norm applied. All hidden layers use ReLu activation function and output layer uses tanh activation function. The exploration strategy for SARA is ϵ-greedy in which the *epsilon* parameter is reset to 0.5 and discounted to 0.05 with discount rate 0.9 everyday.



(a) Actor (b) Critic

Figure B.1: Actor-Critic of SARA