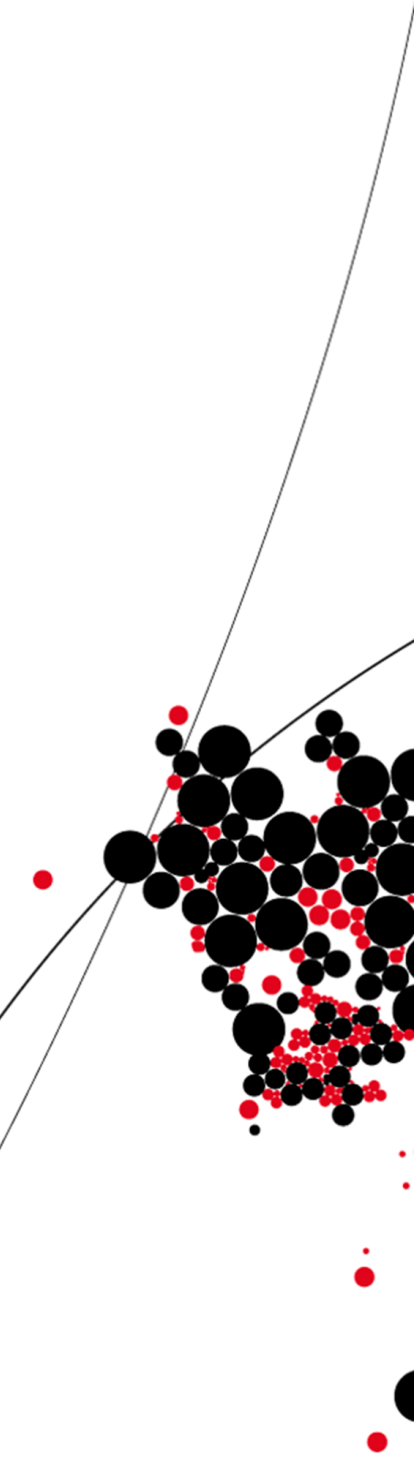
UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering, Mathematics & Computer Science

Human perception of an adaptive agent's fear simulated based on TDRL Theory of emotions

Laduona Dai

Master of Science Thesis
September 2019



Thesis Committee:
Prof.dr. K.P. Truong
Prof.dr. J. Broekens
Prof.dr.ing. G. Englebienne

Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Acknowledgements

First, I would like to thank my thesis committee for their time and input. Second, I would like to thank my supervisors Khiet Truong and Joost Broekens for guiding me through this thesis project. This thesis could never have been completed without their advice and encouragement. I really enjoy the time working with them and have learned so much from them. They have also cultivated in me an interest in the field of affective computing, especially about computational modelling of emotions. Finally, I would like to express my gratitude to my family who provides constant support throughout my life.

Laduona Dai
Schiedam, September 2019

Abstract

Social agents and robots might become widely used in our daily life. To better collaborate with humans, these systems need to be designed to learn from humans and the environment to function autonomously. In recent years, Reinforcement Learning(RL) has been used in different areas with successful results in task learning. An important feature for human-robot cooperation is lacking, robot-to-human transparency. The learning process in RL could be hard for humans to understand and therefore not able to give proper feedback to the robot about its behaviour, which is important for autonomous learning. One way to overcome this problem is to add emotions to the robot, as emotions are used by humans and many animals to express their internal states.

Temporal Difference Reinforcement Learning (TDRL) Theory of Emotion proposes a structure for agents to express appropriate emotions during the learning process. Simulations have been done to test simulate emotions in several scenarios, but there is no further experiment to test how plausible these simulated emotions are when perceived by humans.

This thesis aims to find out the plausibility of simulated fear perceived by humans. 6 different fear calculation methods based on TDRL emotion theory were compared with a baseline, 237 human participants were recruited to evaluate different fear calculation methods in terms of the plausibility of fear intensity and fear location. Results suggest the fear calculation method with ϵ-greedy fear policy($\epsilon = 0.1$) and long-horizon provides a plausible fear estimation, and humans could understand simulated fear based on TDRL Theory of emotions when properly expressed.

Contents

Chapter 1

Introduction

1.1 Introduction

The collaborations between humans and robots are becoming more important when robots are more and more present in society and our daily life. Just like the collaboration between humans, successful teamwork between humans and robots cannot be achieved without an agreed goal and mutual understanding about team members' intention [1]. However, it is impossible for the robot designer to pre-program for all different scenarios and user habits. Also, most users are not experts on machine learning or programming. Therefore, the robot's system needs to be designed to learn from the user and the environment.

Reinforcement Learning(RL), a machine learning method inspired by instrumental conditioning, provides an efficient algorithm for robots to learn new skills by trial-and-error [2]. By repeatedly interacting with the environment, an RL agent(the actor that decides what action to take) could use the adaptive state-action pairs to achieve the optimal state transition policy that maximizes the rewards [3]. In recent years, the autonomous learning feature of RL has been used in different areas, like trajectory optimization [4] and intelligent control [5]. But RL lacks a crucial feature for good teamwork, robot-to-human transparency. The learning process in RL could be hard for users to understand [6]. A user needs to understand why a robot behaves in a certain way [7]. This lack of transparency makes the user hard to give proper feedback to the robot, which is important for autonomous learning [8] [9].

One way to overcome the problem is to add emotions to the robot. Studies have shown that emotional expression by robots can help humans to better understand robots. Participants in [10] report higher pleasantness for an emotional robot, participants in [11] perceive emotional agents more convincing and participants in [12] report more effective communication. With emotions in RL agents, the robot's interactions with human will be more natural for human in the loop RL.

Temporal Difference Reinforcement Learning (TDRL) Theory of Emotion [13] proposes a structure for agents to express appropriate emotions to a human about the robot's internal state during the learning process. In this theory, emotion is defined as a valenced reaction to (mental) events that modify future action, grounded in bodily and homeostatic sensations [13]. In [6], it is argued that TDRL Theory of emotion provides sufficient structure to simulate emotions. However, experimental evidence that these emotions are understandable by, and plausible for human observers is lacking. This thesis aims to investigate the understandability and plausibility of the simulated fear based on TDRL Theory of emotions perceived by humans.

1.2 Report organization

The remainder of this report is organized as follows. In Chapter 2, basic concepts about reinforcement learning will be explained. Chapter 3 describes the related works about emotions in reinforcement learning agents and research questions. The implementation details about fear simulation will be presented in Chapter 4. Chapter 5 will explain the simulation environment and evaluation methods. Results and relevant discussions are presented in Chapter 6. Finally, Chapter 7 will finalize the work and give an outlook for possible future research and improvements on this topic.

Chapter 2

Reinforcement Learning concepts

Reinforcement Learning (RL) is a branch of machine learning methods, it is inspired by how animals learn to interact with their surroundings. In RL, a learning agent has no supervision, it learns the task by discovering on its own via trial-and-error interacting with the environment. During learning, the agent observes the state of the environment and the reward received for the corresponding action. After incorporating these observations in its decision making strategy, the agent chooses its next action based on the current new state and action-selection policy. Then, again, the agent observes the next reward for that action and the loop repeats. The goal of the agent is to maximize the cumulative rewards over time. See Figure 2.1.

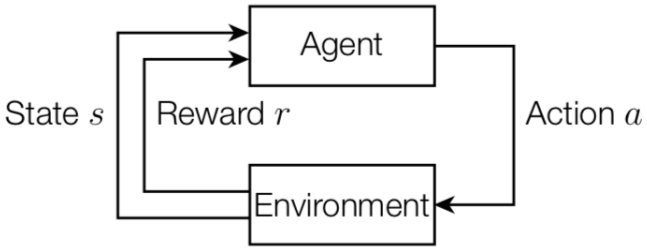


Figure 2.1: Basic RL model [2]

RL problems are usually defined in terms of Markov Decision Process (MDP), which provides a standard formalism for describing sequential decision making in an environment. In this chapter, relevant concepts for MDP will first be introduced, then different methods of RL will be described in the rest of this chapter. Most concepts in this chapter are referenced from [2].

2.1 Markov decision process

A Markov decision [2] process can be defined by $M =< S, A, P, R >$, where

- S: set of states $s \in S$
- A: set of actions $a \in A$
- $P(s_{t+1} \mid s_t, a_t)$: transition function, the probability of reaching state s_{t+1} while in state s_t choose action a_t

- $R(s_t, a_t, s_{t+1})$: reward function, the immediate reward of reaching state s_{t+1} while in state s_t choose action a_t

Return

A typical sequence for MDP looks like this: $(s_0, a_0, r_0, s_1, a_1, r_1 ... s_{n-1}, a_{n-1}, r_{n-1}, s_n, a_n, r_n ...)$

For an environment with a final time step, this sequence is finite. The goal of maximizing cumulative received reward, in the long run, could be defined as to maximize expected return. The return for time step t, can be denoted as G_t:

$$G_t = r_t + r_{t+1} + r_{t+2} + ... + r_T \tag{2.1}$$

For an environment can go on without limit, the above G_t could be infinite. Thus, a discount factor $\gamma \in [0, 1)$ is used to make sure the return is bounded and earlier rewards are preferred over later rewards in the optimization process. G_t is defined as:

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + ... = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \tag{2.2}$$

For a continuing task, the goal would then be to maximize the expected discounted return. Different γ value gives different weights to future rewards. A small γ value emphasizes immediate rewards and a large γ value emphasizes future rewards.

Policy

In MDP, a policy is defined as a mapping from states to the probability of choosing possible action in the corresponding state. The policy at time t, denoted as $\pi(a \mid s)$, represents the probability of choosing $a_t = a$ when $s_t = s$. In RL, the goal is to find the optimal policy that maximizes expected discounted return. Generally speaking, there are 2 approaches to find the optimal policy: value-function based method and policy based method. The former one tries to find the optimal policy by finding the optimal value function, the later one directly learns the optimal policy by policy parametrization [2]. In this thesis, the focus is on value-function based methods.

Value function

In MDP, value function estimates how good it is for an agent to be in a certain state(or state-action pair), and it is measured by the expected value of the cumulative discounted reward over the future following a certain policy start from that state(or in short expected return). The state value function of state s under policy π, denoted as $V_\pi(s)$, is given by:

$$V_\pi(s) = \mathbb{E}_\pi(G_t \mid s_t = s) = \mathbb{E}_\pi(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t = s) \tag{2.3}$$

Similarly, the action value function is defined as the expected return starting from s, taking the action a, and then following policy π. It is denoted as $Q_\pi(s, a)$ and given by:

$$Q_\pi(s, a) = \mathbb{E}_\pi(G_t \mid s_t = s, a_t = a) = \mathbb{E}_\pi(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t = s, a_t = a) \tag{2.4}$$

Bellman Equations

The Bellman equation for the state-value function can be derived by rewrite equation 2.3:

$$\begin{aligned} V_\pi(s) &= \mathbb{E}_\pi(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t = s) \\ &= \sum_a \pi(a \mid s) \sum_{s'} P(s' \mid s, a)[R(s, a, s') + \gamma V_\pi(s')] \end{aligned} \tag{2.5}$$

Similarly, the Bellman equation for the action-value function can be derived as:

$$Q_\pi(s,a) = \sum_{s'} P(s' \mid s,a)[R(s,a,s') + \gamma \sum_{a'} \pi(a' \mid s') Q_\pi(s',a')] \tag{2.6}$$

The Bellman equation expresses the value of a state s in terms of next state s', this means the value of s_t can be calculated if the value of s_{t+1} is known. This property makes it possible for iterative approaches to calculating the value for each state.

Optimal Policy and Optimal Value Functions

A policy π is defined to be better than or equal to policy π' if its expected return is greater than or equal to that of π' for all states, which implies $V_\pi(s) \geq V_{\pi'}(s)$ for $s \in S$. An optimal policy π_* is a policy that is better than or equal to all other policies. And the optimal value function, $V_*(s)$ is defined as:

$$V_*(s) = \max_\pi V_\pi(s), \forall s \in S \tag{2.7}$$

The optimal action value function, $Q_*(s,a)$ is defined as:

$$Q_*(s,a) = \max_\pi Q_\pi(s,a), \forall s \in S, a \in A \tag{2.8}$$

According to optimal policy, the following equation can be derived:

$$V_*(s) = \max_{a \in A(s)} Q_{\pi^*}(s,a) \tag{2.9}$$

By combining Equation 2.4 and 2.9, the Bellman optimality equation for $V_*(s)$ can be derived:

$$\begin{aligned} V_*(s) &= \mathbb{E}_\pi(G_t \mid s_t = s, a_t = a) \\ &= \mathbb{E}_\pi(r_t + \gamma G_{t+1} \mid s_t = s, a_t = a) \\ &= \mathbb{E}_\pi(r_t + \gamma V_*(s') \mid s_t = s, a_t = a) \\ &= \max_a \sum_{s'} P(s' \mid s,a)[R(s,a,s') + \gamma V_*(s')] \end{aligned} \tag{2.10}$$

And the Bellman optimality equation for Q_* is:

$$\begin{aligned} Q_*(s,a) &= \mathbb{E}_\pi(r_t + \gamma \max_{a'} Q_*(s',a') \mid s_t = s, a_t = a) \\ &= \sum_{s'} P(s' \mid s,a)[R(s,a,s') + \gamma Q_*(s',a')] \end{aligned} \tag{2.11}$$

2.2 Model-free vs model-based RL

There are many ways to classify RL algorithms, one that is often used in literature is based on whether the algorithm use environment model. Algorithms that use this model are called model-based RL, the rest are called model-free RL.

2.2.1 Model-free methods RL

Model-free RL algorithms do not need an environment model. These methods learn explicitly by trial-and-error from experience, in other words, they try to learn a value function from interacting with the environment and derived an optimal policy from that. Monte-Carlo learning and Temporal-Difference learning are two typical learning methods of this kind.

Monte Carlo learning method

Monte Carlo(MC) usually can only be applied to episodic tasks, where experience can be divided into episodes(each episode has a clear terminal state, reaching terminal state would reset the task). The learning of the MC method is in an episode-by-episode fashion, instead of a step-by-step(online) way. This means values estimates and policies will only be changed after the completion of an episode. MC method learns optimal policy and value function by iteratively taking two steps: policy evaluation and policy improvement.(see Figure 2.2). The policy is repeatedly improved with respect to the current value function, and the value function is repeatedly improved with respect to the current policy. As they keep change against each other, both policy and value function will approach to optimality [2].

In the policy evaluation step, a roll-out is generated for one episode according to current policy π, $(s_0, a_0, r_0, s_1, a_1, r_1 ..., s_T, a_T, r_T)$. The value function is estimated using this episode.

In the policy improvement step, the policy is improved by taking current value function greedily. For example, the greedy policy of an action-value function Q at state s is to choose an action with maximum action-value:

$$\pi(s) = arg\,\underset{a}{max}\, Q(s, a) \tag{2.12}$$

In this method, by taking these two steps alternatively, an arbitrary policy π_0 will converge to optimal policy π_*, and optimal value function Q_* can be found:

$$\pi_0 \to Q_{\pi_0} \to \pi_1 \to Q_{\pi_1} \to ... \to \pi_* \to Q_* \tag{2.13}$$

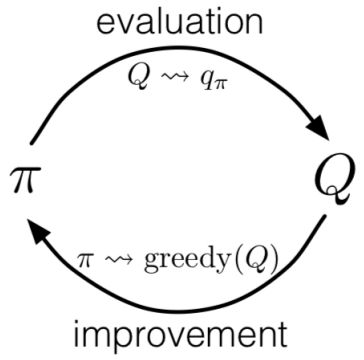


Figure 2.2: Monte Carlo Control [2]

Temporal Difference learning method

Temporal Difference(TD) learning is similar to the MC method, it can directly learn from raw experience without a model for the environment. But, TD learning can learn online after each step, unlike the MC method which only updates policy after each episode. Thus, TD learning can also be used for non-terminal tasks. The idea of TD learning is to update estimates based on other learned estimates, in other words, it updates guess from another guess. The simplest form of this method is the TD(0) algorithm(or one-step TD) proposed by Sutton in 1998, and the update rule for value function is:

$$V(s) = V(s) + \alpha(r + \gamma V(s') - V(s)) \tag{2.14}$$

where α is learning rate, γ is discount factor, s' is current state after taking an action at last state s, r is received reward arrived at state s'. The value of $r + \gamma V(s')$ is called one-step target, and

$(r + \gamma V(s') - V(s))$ is called one-step TD error. One of the famous TD learning algorithms is Q-Learning(see Algorithm 1). In Q-Learning, the action-value function Q directly approximate optimal action-value function.

Algorithm 1: Q-Learning

Initialize Q randomly
for *each episode* **do**
  Initialize s
  for *each step of episode* **do**
    Choose action a for s using policy derived from Q (e.g., ϵ-greedy)
    Take action a, observe reward r and next state s'
    $Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \underset{a}{max} Q(s',a) - Q(s,a))$
    $s \leftarrow s'$
  end
end

2.2.2 Model-based methods RL

Model-based RL methods can use information about environment dynamics for planning. Generally, model-based RL can be divided into 2 categories, if the transition functions and reward functions are known, then Bellman optimality equations can be solved iteratively. Otherwise, the model can be estimated online by collecting information about the environment. The former approach is also known as Dynamic Programming since it requires the knowledge of the model, which is rarely the case in practice. Nowadays most model-based methods choose to learn the model online. One of the examples is Dyna-Q algorithm.

Dyna-Q

Dyna-Q is a simple architecture that combines planning and learning at the same time. In this architecture, real experience is used in two ways: i)improve the model, ii) directly improve value function and policy. The first one is called model-learning, the second one is called direct reinforcement learning(direct RL) [2]. Figure 2.3 illustrates this architecture.

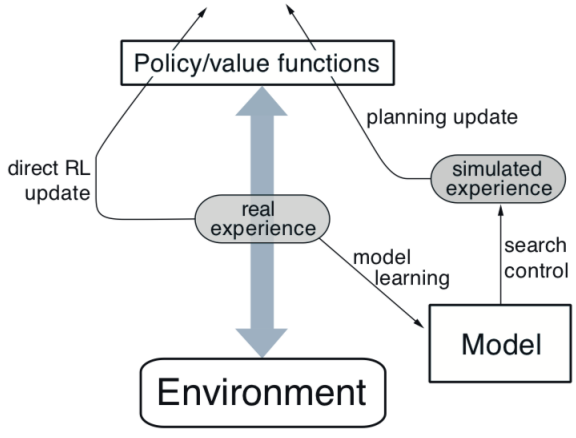


Figure 2.3: Dyna architecture. Real experience is used to improve value function & policy by: direct RL and model-learning. [2]

Algorithm 2 shows the pseudo-code for Dyna-Q. It first performs one-step tabular Q-Learning(Step1-6), then learning the model by observing the received reward and next state for a particular pair of

state and action, the transition function P and reward function R is thus updated according to this information(Step 7). Then, it performs random-sample one-step tabular Q-planning(Step 8-12), in the planning, simulated experiences are generated from the model and used to improve value function as if they are real experiences.

Algorithm 2: Dyna-Q

1 Initialize Q and $Model$
2 **Loop forever:**
3 $s \leftarrow$ current (non terminal) state
4 $a \leftarrow \epsilon$-greedy(s,Q)
5 Take action a, observe reward r and next state s'
6 $Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max_a Q(s',a) - Q(s,a))$
7 $Model \leftarrow r, s'$(assuming deterministic environment)
8 **Loop repeat n times:**
9 $s \leftarrow$ random previously observed state
10 $a \leftarrow$ random action previously taken in s
11 $r, s' \leftarrow Model$
12 $Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max_a Q(s',a) - Q(s,a))$

2.3 Exploration-Exploitation

In RL, most algorithms initialize with a random policy, so the agent could try different possibilities and explore the environment(exploration). After some learning, the policy will converge to a solution and the agent will stick to that solution(exploitation). But if the agent always chooses the actions that stick to this solution, it might miss a better solution. This is the dilemma of exploration and exploitation. In the following, 2 action-selection policies will be discussed for dealing with this dilemma.

ϵ-greedy policy

In this policy, an agent chooses the optimal action with a probability of $1 - \epsilon$ and randomly otherwise(See equation 2.15). It is widely used in RL algorithms because of its simplicity and intuitive nature. In practice, ϵ is usually set to a large value(for example, 0.9) initially to ensure the agent has enough exploration for the environment, then let the value decay to a small number(for example, 0.1) when approaching convergence to make sure there is still some exploration while most of the time the agent is following the found policy.

$$a = \begin{cases} \underset{a \in A}{arg\,max}\, Q(s,a) & with\ probability\ 1-\epsilon \\ Random\ a, a \in A & with\ probability\ \epsilon \end{cases} \tag{2.15}$$

Boltzman/Softmax policy

One disadvantage of ϵ-greedy policy is that it chooses all actions with the same probability when it chooses a random action. This means the worst action has equal chance to be chosen as other actions, and this can be problematic in some situations. The softmax policy tries to overcome this problem by giving actions with a higher value a higher probability of being selected. In the following equation 2.16, τ is a positive parameter. When $\tau \to 0$, softmax becomes greedy action selection, and when $\tau \to \infty$, softmax becomes random action selection. The drawback of this policy is that it is

difficult to find a proper τ for different environments.

$$\pi(a \mid s) = \frac{exp(Q(s,a)/\tau)}{\sum_{b \in A} exp(Q(s,b)/\tau)} \tag{2.16}$$

Chapter 3

Related work & research questions

3.1 Emotion in nature and psychology

In psychology, there are mainly three emotion theories: categorical, dimensional and appraisal theories.

Categorical emotion theory suggests there exists a set of discrete emotions shared among different cultures and societies [14]. According to evolution theory, these emotions were selected through evolution to ensure species' survival and improve the adaptive fit in primitive environments [15].

The Circumplex model, the most prominent dimensional model of emotion, was developed by James Russell [16] [17]. This theory suggests that there is a two-dimensional emotion space, arousal(emotion intensity) and valence(positive or negative). In this model, emotional states could be represented in a two-dimensional space with different levels of arousal and valence.

Appraisal theory was proposed by Arnold [18] and developed by Lazarus [19], it considers emotions as appraisal processes triggered by stimuli evaluated according to personal relevance. Compared to the previous two theories, modern appraisal theory define emotions as processes rather than states [20].

The role of emotion can be categorized into two groups: intrapsychic and interpersonal. The former one refers to the roles within an individual: emotions help ensure individual survival, adjust behaviour by goal management, and contribute to learning and information acquisition [21] [22] [23]. The interpersonal refers to the roles in social interaction, where emotions are used to communicate internal state and mediate group behaviour to maintain social structure [21].

3.2 Computational models of emotions in RL agents

The idea of implementing an RL agent or robot with emotions is not new. According to the review paper by Moerland et al. [24], many studies have been done in the field of the computational modelling of emotion in RL agents. The benefits of using emotions can be roughly divided into three categories: learning efficiency, emotion dynamics and human-robot interaction(HRI).

Learning efficiency

According to [24], the majority of the researches about emotion in RL are related to learning efficiency. For example, in one of the core papers in this field by Gadanho and Hallam [25], emotions(joy, sad, fear and angry) are derived from homeostasis/internal drive. In their emotion model, each

emotion intensity is based on a set of virtual robot's internal feelings, and feelings are derived from the robot's sensations. The set of feelings are (Hunger, Pain, Restlessness, Temperature, Eating, Smell, Warmth, Proximity). For instance, hunger rises when lacking resources, pain rises when bumping with obstacles, restlessness increases if the robot does not move and the temperature rises with high motor usage. After feelings are generated, each emotion is calculated through linear weighted dependencies from feelings, e.g. joy is derived from eating or smell food, sad from high hunger, fear from pain and anger from high restlessness. In their experiments, they compared the performance of emotional and non-emotional robots in a virtual world with obstacles and energy sources for a surviving task of maintaining adequate energy levels. The robot used reinforcement learning techniques(e.g. Q-learning) to learn the environment, and the emotion model was also integrated into the reinforcement learning framework for robot control. Results from their experiments suggest that emotional robots could achieve higher average reward and more likely to avoid collisions.

Another example is by the work of Marinier and Laird [26], in their model, emotions are elicitated based on the appraisal theory by Scherer [27]. Appraisal theories hypothesize that an emotional reaction to a stimulus is the result of an evaluation of that stimulus along a number of dimensions, most of which relate it to current goals. In Marinier and Laird's work, they used a subset of the appraisal dimensions described by Scherer. In their model of emotions, emotional feedback helps drive reinforcement learning by controlling RL attributes with different appraisal dimensions.

Apart from the above 2 studies, there is also much other research on learning efficiency, for example, emotional agent could learn faster(Ahn and Picard [28], Zhang and Liu [29]), emotional agent could avoid obstacles and collisions(Lee-Johnson et al. [30], Shi et al. [31]), emotions help agents improve the ability to switch goals(Cos et al. [32], Goerke [33]), and emotion helps in improving the exploration(Broekens et al. [34]).

Emotion dynamics

In this category, emotion signals are usually compared with known psychology theories. Jacobs et al. [35], propose a computational model of joy, distress, hope and fear as mappings between RL primitives(reward, value, update signal, etc...) and emotion labels. Joy/distress is derived from positive or negative TD for the current state, and hope/fear is derived from the learned value of the current state. In the experiments, an agent-based simulation was used for an RL-based agent in a virtual maze world. Results showed that their model meets the requirements from emotion elicitation literature [36], emotion development [37], emotion habituation and fear extinction [38].

Some research in this category worked on how emotion dynamics fit in social interaction. Tanaka et al. [39] focused on a model that emotions are affected by interacting with humans. In their emotion model, emotions are elicitated through internal state variables: fullness, pain, hunger, comfort, fatigue, and sleepiness. Different combinations of internal state values are associated with different emotions, and 7 emotions could be generated: happiness, sadness, anger, surprise, disgust, fear and neutral. In experiments, participants interact with the robot by hitting or padding, then the robot generates emotions and behaviours (gesture, voice and facial expression) according to the model. The results showed appropriate emotional responses(joy or fear) to people padding or hitting the robot. Moussa and Magnenat-Thalmann [40] included emotions, attachment and learning in a decision-making architecture for a virtual agent. Their framework was evaluated through simulation evaluations by interacting with users under different scenarios. The results showed correspondence with psychology theories and the virtual agent showed an appropriate emotional response to different user behaviours.

Human robot interaction

In this category, the focus is to show how emotions could be beneficial for interacting with humans, and this is usually done by participants filling in questionnaires after the experiments. Ficocelli et al. [12] proposed an emotion-based assistive behaviours model for a real robot that focuses on natural HRI scenarios. In their emotion model, emotions are derived from human's affective states and assistive tasks while interacting with humans. The robot would try to persuade participants to engage in an activity utilizing the observed participant's emotional state and the robot's emotion module. Results of comparing a robot with a working emotional module and a robot without emotional module showed emotional robot is more efficient at obtaining compliance from participants.

El-Nasr et al. [11] proposed a computational model of emotions that uses a fuzzy-logic representation to map events and observations to emotional states. Reinforcement learning methods were also incorporated in the agent to learn about the environment so that the agent could adapt its response to make its behaviour more believable. To evaluate their model, a virtual pet was implemented and participants were asked to perform tasks in different scenarios with this pet. After experiments, participants were asked to fill in a questionnaire to assess the virtual pet's behaviour in different aspects. Results showed that the adaptive component of the emotion model made the virtual pet's behaviour more convincing.

In the work of Kim and Kwon [10], an emotion model was implemented based on cognitive appraisal theory. A real robot that interacts with humans for a given service task utilized this model to appraise task-related situations and generate corresponding emotions. The interaction task is a game consisting of some questions. Two experiments were carried out to evaluate the interaction: the first was used to understand the overall affective evaluation of the interaction with the robot and the second one was used to understand the suitability of the emotion model. The evaluation of the experiments showed that an emotional robot during the task gives participants more positive feelings.

Summary from previous studies

So far, most of the computational models of emotions in RL agents and robots are based on cognitive appraisal theory and they assume emotions arise form a cognitive reasoning process instead of a learning process based on exploration and positive/negative feedback. This means that in most of the models, a cognitive reasoning module is needed. TDRL Theory of emotion, on the other hand, assumes emotions that are simulated and expressed by the agent are grounded in the learning process of that agent. So no cognitive reasoning module is needed to generate emotions.

In this thesis, the computational model of emotions is based on the work of [41], it is one implementation of the TDRL Theory and results of it showed emotions occur at appropriate states in 3 simulation scenarios. Since the generated emotions have only been tested in simulations and no experiment has been done to check the plausibility of the emotions perceived by humans, my contributions in this field are: 1)Verify the plausibility of the simulated fear emotion, 2) Test different fear calculation methods for fear intensity and fear locations.

3.3 TDRL Theory of Emotions

The essence of the TDRL Theory of Emotion is that all emotions are manifestations of temporal difference errors [13]. The definition of emotion in here is a valenced experience in reaction to (mental) events providing feedback to modify future action tendencies, grounded in bodily and homeostatic sensations and evolutionary developed serviceable habits [13].

Joy (Distress) is the manifestation of a positive (negative) temporal difference error [13].
In Q-learning, Joy and Distress are defined as [6]:

$$if(TD > 0) \Rightarrow Joy = TD \tag{3.1}$$

$$if(TD < 0) \Rightarrow Distress = TD \tag{3.2}$$

With TD defined as:

$$TD = r + \gamma \max_{a'} Q(s', a') - Q(s, a)_{old} \tag{3.3}$$

Hope (Fear) is the anticipation of a positive (negative) temporal difference error [13].
In [41], Hope and fear are simulated using Monte Carlo Tree Search procedure(UCT). Results from 3 test scenarios suggest hope and fear could emerge from anticipation in a model-based RL.

3.4 Research questions

TDRL theory of emotion could provide sufficient structure to simulate emotions [6], but there is no experiment so far to test how humans feel about these simulated emotions. Therefore, in this thesis, a TDRL-based emotion model is used for a virtual agent to generate fear in the autonomous learning process. Further, an approach to evaluate the generated fear is proposed. Concerning this issue, two main research questions are addressed:

- How plausible is the simulated fear perceived by humans?
- What is a proper fear calculation method for fear intensity and fear locations?

Chapter 4

Fear simulation based on TDRL Theory of Emotions

4.1 Overview of the fear simulation architecture

In this section, the details of the architecture for fear simulation in RL agents will be described. The architecture has three major components: forward planning, fear calculation, and agent learning. Algorithm 3 shows abstract steps of this architecture. As the algorithm shows, before each agent action, the agent uses forward planning based on learned knowledge to imagine different futures(step 3). From these imaged futures, the agent calculates fear for future states according to its fear policy. Then, the agent takes a step using the action-selection policy π. After observing the immediate reward r and next state s', the agent uses this observation for learning by updating the model and Q values. This loop then goes up until terminated. One important notice for this architecture is that the emotion fear does not affect the action-selection policy. In the following sections, these components will be described in more detail.

Algorithm 3: Fear simulation based on TDRL Theory of Emotions

1 Initialize R, P, Q
2 **while** *s is not terminal* **do**
3 Fear $\leftarrow$ MCTS-T+(s, d, N) // section 4.2&4.3
4 $a \leftarrow \pi(s, a)$ // action-selection policy
5 Take action a, observe r, s'
6 R, P $\leftarrow$ Update model(s, a, r, s') // section 4.4
7 Q $\leftarrow$ Q-Update(l, d) // section 4.5
8 $s \leftarrow s'$
9 **end**

4.2 Planning with MCTS-T+ (modified to stochastic model)

In order to estimate a proper future state, forward planning should be used for the current state. A commonly used algorithm for this purpose is Monte Carlo Tree Search (MCTS) [42]. Different variations of it are often used in board games, and the most famous one is Upper Confidence Bounds for Trees (UCT) [43] [44]. But one disadvantage of MCTS is that it is not good at dealing with asymmetric tree structure or loops in the tree, which is not a problem in board games. In navigation tasks, this

can occur quite often, like a narrow tunnel between walls. In [45], a MCTS-T+ algorithm is proposed, where an uncertainty attribute is added to deal with the previously mentioned problems. Testing results on a set of OpenAI Gym and Atari games show this algorithm always perform better than or at least equivalent to standard MCTS. However, it is only for the non-stochastic environment, which limits the usage of it. In this thesis, a modified version of it is used to fit the stochastic environment. Different from standard MCTS procedure of 4 steps(select, expand, roll-out and backup), in here MCTS-T+ is only used for building a tree structure and environment dynamics is learned online, thus the roll-out step is not used for a future state value estimation. The general approach is shown in Algorithm 4. The computational budget in this algorithm is decided by the maximum depth of the tree d and the number of search iterations N.

Algorithm 4: MCTS-T+(s, d, N) approach

```
Create root node with state s
while within computational budget do
    Select child node
    Expand current node
    Backup
end
return built tree
```

In Algorithm 4, three steps are applied per search iteration:

1): Select child node

2): Expand current node

3): Backup

These three steps will be explained in details in the following.

1.**Select child node**

Starting from root node, recursively select child nodes according to Algorithm 5.

Algorithm 5: Select child node

```
Current node is s
while s is fully-expand & non-terminal do
    s ← SELECTCHILD(s) // Algorithm 6
end
return s
```

In the above algorithm, a node is fully-expand if all possible child nodes according the built model is added to this node.

Algorithm 6: function SELECTCHILD(s)

1 Current node is s
2 **if** *node s has stochastic child nodes* **then**
3 **for** *all $a \in$stochastic action* **do**
4 **for** *all $s' \in$ stochastic child nodes with action a* **do**
5 $\sigma(s,a) \leftarrow (\sum_{s'}[n(s') * \sigma(s')])/(\sum_{s'} n(s'))$
6 **end**
7 **end**
8 **else**
9 pass
10 **end**
11 $act \leftarrow arg\,max_a \left(Q(s,a) + \sigma(s,a) * \sqrt{\frac{2 \cdot log\, n(s)}{n(s,a)}}\right)$
12 $s' \sim P(s' \mid s, act)$ // sample a child according to transition probability
13 **return** child s'

In Algorithm 6, a node having stochastic child nodes means there are multiple possibilities of next state s' when in state s taking action a. For example, in Figure 4.1, the left tree structure has no stochastic child node, but in the right tree structure, node s_0 has 2 stochastic child nodes and stochastic action is a_1 because the action a_1 can result 2 possible next state s_1 and s_2.

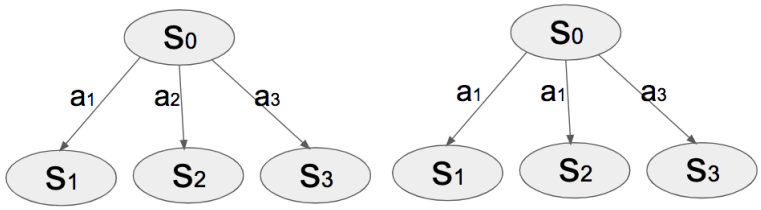


Figure 4.1: Tree structure without stochastic child nodes(left) and tree structure with stochastic child nodes(right).

In step 5, $\sigma \in [0,1]$ is the tree uncertainty attribute. $\sigma(s)$ represents the tree uncertainty below the sub-tree of s, $\sigma(s) = 1$ indicates a completely unexplored sub-tree below s, and $\sigma(s) = 0$ indicates a fully enumerated sub-tree. Similarly, $\sigma(s,a)$ represents the tree uncertainty below the sub-tree of s for branches with action a. And $n(s')$ represents the number of visits for node s'.

In step 11, an action is chosen according to equation similar to UCB1 [46], which balance exploration and exploitation. $n(s,a)$ represents the total number of visits for all the child nodes with action a under node s.

2.**Expand current node**

If current selected node is non-terminal, then it will be expanded with a child node. The child node is randomly chosen from all possible non-added child nodes. After this child node c is added, it will be initialized with attributes $\{n(c) = 0, \sigma(c) = 1\}$ if node c is non-terminal, otherwise $\{n(c) = 0, \sigma(c) = 0\}$, and this node c is chosen as current node.

3.**Backup**

Finally, recursively backup the uncertainty attribute and node visit information from the current

node until the root node s_0. The procedure is shown in Algorithm 7.

Algorithm 7: Backup

```
1  Current node is s
2  while s is not root node do
3      if s is leaf node then
4          n(s)+ = 1
5      else
6          n(s)+ = 1
7          UncertaintyUpdate(s)
8      end
9      s ← ParentNode(s)
10 end
```

In step 7, the UncertaintyUpdate for node s is by the equation:

$$\sigma(s) \leftarrow \frac{\sum\limits_{s' \in C} m(s') \cdot \sigma^*(s')}{\sum\limits_{s' \in C} m(s')} \tag{4.1}$$

where C is all possible child nodes according to model, and:

$$m(s') = \begin{cases} n(s') & ,if\ n(s') \geq 1 \\ 1 & ,otherwise \end{cases} \quad \sigma^*(s') = \begin{cases} \sigma(s') & ,if\ n(s') \geq 1 \\ 1 & ,otherwise \end{cases} \tag{4.2}$$

4.3 Fear calculation

In [47], the belief-desire theory of emotion(BDTE) is proposed to describe emotions based on two dimensions: belief and desire. In this theory, belief about a state s, $b(s) \in [0, 1]$ is defined as the perceived probability of state s happening, and desire about state s, $d(s) \in \mathbb{R}$, as the desirability for s. According to this theory, fear originates when $0 < b(s) < 1$ and $d(s) < 0$, since fear is about an uncertainly future event that is undesirable. In RL terms, fear can be modelled as anticipated negative temporal differences about a forward state [41]. In BDTE, fear intensity is defined as the product of belief and desire, $I(s) = b(s) \times d(s)$. Thus, in this thesis, fear is defined as the worst product of likelihood and temporal difference among all imaged future states. This will be explained in more detail below.

When the planning with MCTS-T+(s, d, N) is finished, a tree is built staring from root node s with depth d and N forward traces. For example, Figure 4.2 shows a built tree starting from root node s_0, with depth $d = 2$ and $N = 5$ trajectories. In the example tree, the 5 forward planning trajectories are:

$T_1 = \{s_0, s_1\}$

$T_2 = \{s_0, s_2\}$

$T_3 = \{s_0, s_1, s_3\}$

$T_4 = \{s_0, s_1, s_4\}$

$T_5 = \{s_0, s_2, s_1\}$

The different planning trajectories are like different imaged futures. For each imaged future, fear will be calculated for the end state, but the most feared state s' is the state that has highest negative

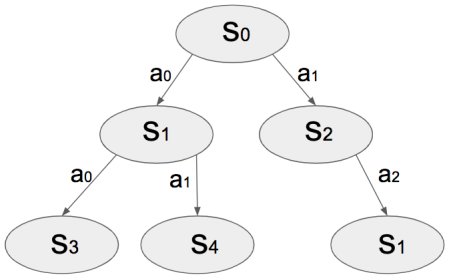


Figure 4.2: A MCTS-T+(s, d, N) tree with $s = s_0, d = 2, N = 5$. (Two s_1 nodes in the tree are treated as different nodes, they do not share visit count information)

TD error among all imaged futures. For each trajectory T, the fear in node s_0 is given by:

$$\begin{aligned} f(T) &= f(s_{end} \mid s_0) \\ &= b(s_{end} \mid s_0) \times d(s_{end} \mid s_0) \\ &= \prod_{t=0}^{end-1} [\pi(s_t, a_t) \cdot P(s_{t+1} \mid s_t, a_t)] \times \left(\left[\sum_{t=0}^{end-1} \gamma^t R(s_t, a_t, s_{t+1}) \right] + \left[\prod_{t=0}^{end} \gamma \right] V(s_{end}) - V(s_0) \right) \end{aligned} \tag{4.3}$$

where s_{end} is the last state in a trajectory, $\pi(s_t, a_t)$ is the emotion policy and $V(s) = \max_a Q(s, a)$. The uncertainty of a state depends on the emotion policy and environment dynamics $P(s_{t+1} \mid s_t, a_t)$.

In practice, there could be multiple paths towards a state s', for example, both T_1 and T_5 traces lead to state s_1. This means state s_1 is more likely to happen, so the fear for s_1 is the sum of trace T_1 and T_5, $F = f(T_1) + f(T_2)$. In general, the most feared state s' when in state s_0 and its intensity is defined as:

$$F(s_0) = \min_{s'} \left[\sum_{T \in Trajects\ end\ with\ s'} f(T) \right] \tag{4.4}$$

4.4 Update model

After each step, (s, a, r, s') is observed and will be used to update the transition function and reward function by normalized transition probability and average observed rewards, respectively. Assuming the observed sequence is like: $s_0, a_0, r_0, s_1, a_1, r_1, s_2, a_2, r_2, s_3$... Then:

$$P(s' \mid s, a) = \frac{\sum_{i=0} 1\{s_i = s, a_i = a, s_{i+1} = s'\}}{\sum_{i=0} 1\{s_i = s, a_i = a\}} \tag{4.5}$$

$$R(s, a, s') = \frac{\sum_{i=0} 1\{s_i = s, a_i = a, s_{i+1} = s'\} r_i}{\sum_{i=0} 1\{s_i = s, a_i = a, s_{i+1} = s'\}} \tag{4.6}$$

4.5 Q-Update(l,d)

After $P(s' \mid s, a)$ and $R(s, a, s')$ are updated for the observation of (s, a, r, s'), Q estimates will also be updated according to the Bellman equation:

$$Q(s, a) = \sum_{s'} P(s' \mid s, a)[R(s, a, s') + \gamma \max_{a'} Q(s', a')] \tag{4.7}$$

During learning, the agent always keeps a record of the l most recent steps including state, action and reward information. To propagate back TD errors fast over state-space, this Q update process is repeated for the l most recent step in reverse chronological order(e.g. the step just happened update first). And for each step in this record, this update process is again repeated for the steps that could lead to the step in the record until the depth of d.

Chapter 5

Simulations and evaluation methods

5.1 Introduction

In this chapter, the simulation environment and evaluation methods will be described. According to the research questions and TDRL Theory of emotions, the layout of the maze environment with a ghost is chosen based on the following reasons:

- The environment should not be too complex. To evaluate different fear calculation methods, human participants will be asked to watch the agent's learning from scratch in the environment. If the layout is too complex, it could take a long time for the agent to learn the task, and boring to watch.
- Since fear is an uncertain negative TD error from a future state, the stimuli in the environment for fear should be stochastic, so it can't be a still obstacle with negative rewards.
- The expression of the emotion should be intuitive and simple. Thus, the fear intensity of the agent is represented by the agent's color, instead of a separate diagram like a bar chart.

5.2 Environment

In this thesis, the learning agent and environment are implemented using Python and OpenAI Gym library [48]. OpenAI Gym is a toolkit for developing and comparing reinforcement learning algorithms, it contains a collection of benchmark environments designed for testing different RL algorithms. Since the environment used in this thesis has a different focus than normal benchmark environments in Gym, Gym library is only used for the environment visualization.

Situation	Reward
Make a step	-0.01
Hit wall	-0.1
Hit ghost	-10
Reach target	10

Table 5.1: Reward for agent in different situations

The environment used for this thesis is shown in Figure 5.1, it is a typical 2D discrete stochastic grid world maze. In this environment, the learning agent(gray block) starts at an initial position(central

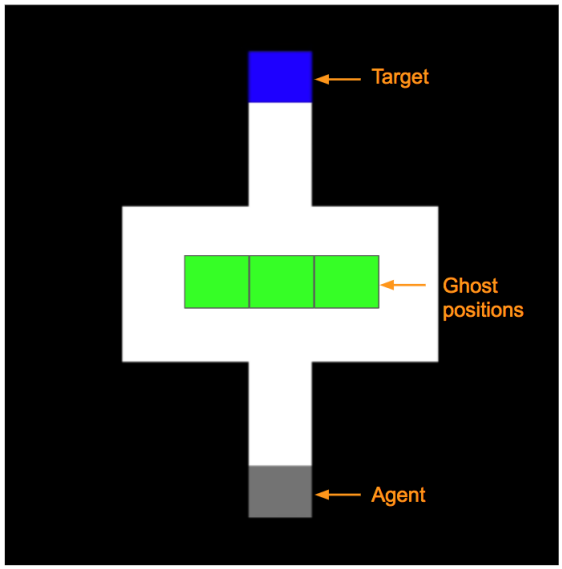


Figure 5.1: Agent's learning environment

lowest). It can move up, down, left or right one step each time. If it bumps into a wall(black surrounding), it will collide with the wall, receiving a small negative reward and stay at the original position. If it reaches the target(blue block, central highest), it will get a big positive reward and be sent back to the initial position(central lowest). On the way to the target, there is a ghost randomly appearing in one of the three green positions. If the agent hits the ghost, the agent will receive a big negative reward. The rewards for different situations are shown in Table 5.1. The agent's state is represented by its position in the grid world and whether the agent hits the ghost. The set of states for the agent is shown below:

$S = \{13, 22, 31, 38, 39, 40, 41, 42, 47, 48, 49, 50, 51, 56, 57, 58, 59, 60, 67, 76, 85, 48D, 49D, 50D\}$

Each numeric state represents a position in this grid world without hitting the ghost(13 is the state of reaching Target, 85 is the initial start state). 48D, 49D and 50D represent three states when the agent hits the ghost in state 48, 49 and 50, respectively.

The set of actions for the agent is: $A = \{move\ up, move\ down, move\ left, move\ right\}$

Some examples are shown in the following:

- Agent move up from initial position not hit anything: $s = 85, a = move\ up \rightarrow s' = 76, r = -0.01$
- Agent move left from initial position and hit wall: $s = 85, a = move\ left \rightarrow s' = 85, r = -0.1$
- Agent move up and hit the ghost: $s = 58, a = move\ up \rightarrow s' = 49D, r = -10$
- Agent move up and not hit the ghost: $s = 58, a = move\ up \rightarrow s' = 49, r = -0.01$
- Agent move up and reach the target: $s = 22, a = move\ up \rightarrow s' = 13, r = 10$

Initially, the agent has no prior knowledge about the environment(it doesn't know the layout, existence of the target, etc...), and it only has local information(e.g. it knows move left will get a reward of -0.1, but it does not know the reward of move up 5 times). By interacting with the environment, the agent will learn more about it and builds a more accurate model for it.

In this thesis, the chosen action-selection policy for the learning agent is ϵ-greedy. In total, the agent learns for 500 steps, the ϵ value decay linearly for the first 300 steps from 0.5 to 0.1, and maintained at 0.1 for the last 200 steps. With these ϵ values, the agent is able to first explore the

environment in all directions, but still maintain some randomness after the agent converges to a solution. From the behaviour aspect, one would observe the agent first explores the environment like randomly, then after the agent learns the ghost positions are harmful, it would try to avoid those position. In the end, the agent would follow the path to the target by going left or right in the center to avoid hitting the ghost.

5.3 Simulation settings

Hardware configuration

The experiment device is a laptop with 2.3 GHz Intel Core i5 CPU, 16 GB RAM, macOS 10.13 system. The used libraries are: Python 3.5 and OpenAI Gym.

Experimental variation of calculation methods

In this thesis, the emotion of fear doesn't affect the learning process(action-selection policy is not affected by fear in Algorithm 3). Therefore, to control variables, only one version of the learning agent is trained in the environment. Then, different fear calculation methods are varied to calculate fear for the same version of agent movement behaviour.

6 different fear calculation methods(3 fear policies x 2 horizons) are compared with a baseline for the experiment, see Table 5.2. Note that fear policy is different from action-selection policy. The fear policy is the π in MCTS-T+ used for assigning a probability for a future state in a forward planning tree, it does not affect how the tree is built. Horizon is the d in MCTS-T+, it affects the depth of the built tree, and it represents how far the agent can imagine for the future. Short-horizon is for MCTS-T+ with $d = 2$, and long-horizon is for MCTS-T+ with $d = 10$.

Version 1 uses ϵ-greedy fear policy with $\epsilon = 0.1$(choose the best action with probability of 0.9 for future states in a built forward planning tree) and MCTS-T+ with $d = 2$.

Version 2 uses softmax fear policy with $\tau = 5.0$(choose the best action with a high probability for future states in a built forward planning tree) and MCTS-T+ with $d = 2$.

Version 3 uses ϵ-greedy fear policy with $\epsilon = 1$(assign equal probability for choosing different actions for future states in a built forward planning tree) and MCTS-T+ with $d = 2$.

Version 4 is the baseline, it does not use MCTS-T+ or any fear policy to calculate fear for a future state. It generates a random value uniformly in the range of $[0, 1]$ at each step for a random location in the grid world.

Version 5 uses ϵ-greedy fear policy with $\epsilon = 0.1$ and MCTS-T+ with $d = 10$.

Version 6 uses softmax fear policy with $\tau = 5.0$ and MCTS-T+ with $d = 10$.

Version 7 uses ϵ-greedy fear policy with $\epsilon = 1$ and MCTS-T+ with $d = 10$.

Version	Fear policy	Horizon
1	ϵ-greedy_0.1	2
2	softmax_5.0	2
3	ϵ-greedy_1.0	2
4(baseline)	Ranom	-
5	ϵ-greedy_0.1	10
6	softmax_5.0	10
7	ϵ-greedy_1.0	10

Table 5.2: Different fear calculation methods

To investigate the plausibility of the fear intensity and fear location separately, 2 different videos[1] have been made for each of the 7 fear calculation methods. The first video shows the fear intensity by the color of the agent(fear intensity is the normalized fear value by dividing the fear value with the max fear value throughout the entire process for each method), if the agent has no fear, it's color is gray, if agent has max fear, then it will show red color(color scale is shown in Figure 5.2, and Figure 5.3 shows 2 examples).

The second video focuses on the fear location, this time the color of the agent itself doesn't change according to its fear intensity, but the agent's most feared location in this environment will be marked by red(see Figure 5.4).

Apart from 2 videos, a fear plot is also made for each fear calculation method. This plot shows fear value and location in the environment when the agent is set back to the initial position after 500 steps of learning. By using this plot, an observer will be able to assess the plausibility of the fear policy and horizon at the same time. One of the example is shown in Figure 5.5, the plots for all fear calculation methods are shown in Appendix A.3.

Figure 5.2: Color scale for the agent's fear intensity

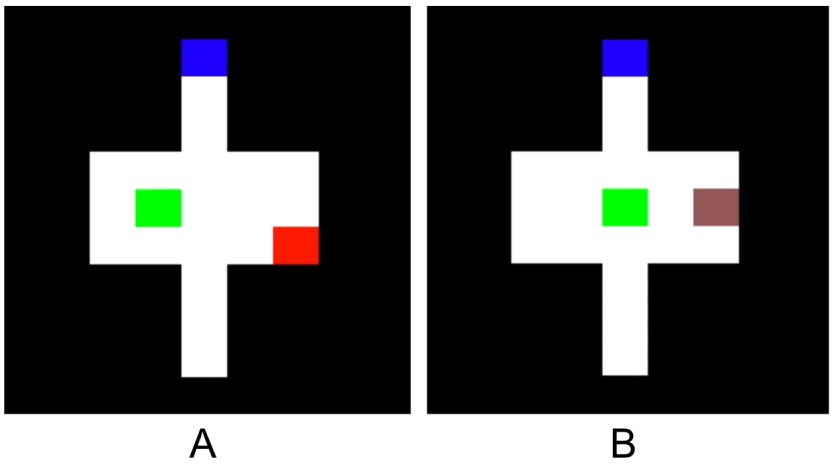


Figure 5.3: 2 examples of different agents' fear intensities. The agent in A(left) has more fear than the agent in B(right)

[1] All 14 videos are available at `https://www.youtube.com/playlist?list=PLfe14MM6YWfyK_edFQYJxnI2ARePC3psm`. 'a' is for fear intensity, 'b' is for fear location.

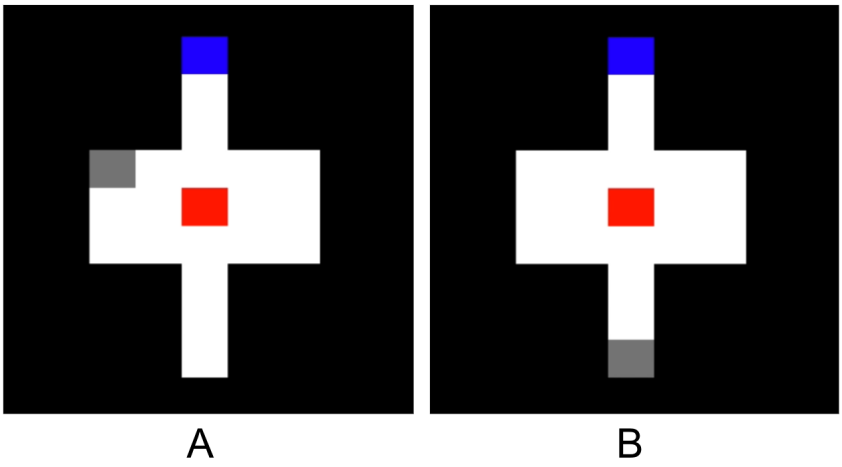


Figure 5.4: 2 examples of the agent's most feared location. The agent's color does not change according to its fear intensity(always in gray), the ghost is not shown for clarity. The most feared location is marked in red.

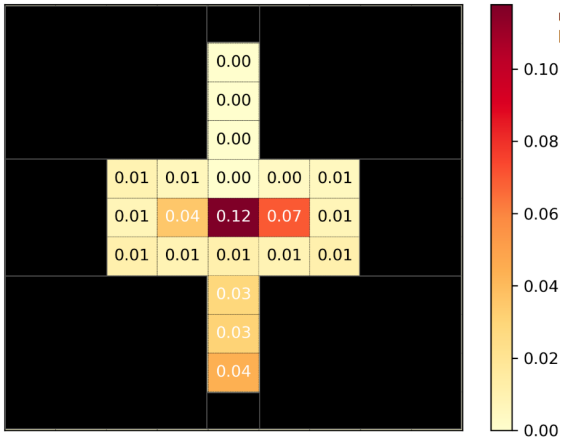


Figure 5.5: Fear plot for version 5(ϵ-greedy fear policy with $\epsilon = 0.1$ and long-horizon).

5.4 Evaluation method

Evaluation protocol and dependent measures

To evaluate how different fear calculation methods affect human's perceived fear plausibility, participants watched simulations for different fear calculation methods using pre-recorded videos. A between-subject setup is used, each participant only saw one of the seven conditions. All participants were asked to watch 3 videos about the agent's learning process and a fear plot from an agent with the same fear calculation method. The overall flow is shown in Figure 5.6. Video 1 was shown to participants twice to ask different questions.

Each video and plot was followed by a set of mandatory questions for participants to answer. The length for each video is about 2 minutes, and the total time for an average person to finish this survey is about 12 minutes.

In the survey introduction, the background information was explained to participants, for example, the gray block is the agent, the blue block is the target with positive reward and the agent receive small punishment when hitting the wall. But participants were not told the existence and function

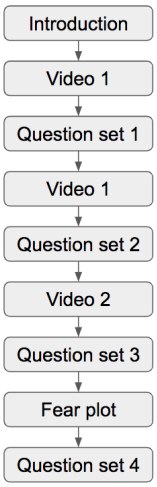


Figure 5.6: Study flow chart

of the ghost, and they were only informed the color of the agent represents one particular emotion intensity. Video 1 shows the agent's learning in this environment for 500 steps, emotion intensity as the color scale in Figure 5.2, and the ghost is randomly appearing in one of the three green positions. After this video, in Question set 1, participants were asked to answer the following 3 questions:

1. What do you observe in this agent's behaviour?

2. Which emotion do you think the agent's color intensity represents?
A.Anger B.Disgust C.Fear D.Happiness E.Sadness F.Surprise G.Other

3. Which of the following item do you think the popping green block represents?
A.Food B.Danger C.Other

The first question was set up to check how participants observe the agent's learning process in general. For the second question, the purpose is to check participants' first instinct about the agent's emotion type just by observing the emotion intensity of the agent accompany with its moving behaviours. The third question is used to check the participants' understanding of the ghost's function just by observing the agent's behaviours and emotion intensity.

After participants answered the above questions, they were told the emotion felt by the agent is fear, and the agent will get punishment for hitting the green block. Then they were instructed to watch Video 1 again, and answer the following questions in Question set 2:

1. How plausible do you think the fear intensity is?
Rate your answer on a scale 0-10. (0 = fear intensity makes no sense at all, 10 = fear intensity makes perfect sense)

2. Rating explanations:

When previous sections were finished, participants were told to watch Video 2 and given the information that this time the most feared location will be marked in red, the popping green block and the intensity of fear will not be shown for clarity. Then, they were asked to answer questions in Question set 3:

1. How plausible do you think the fear location is?
Rate your answer on a scale 0-10. (0 = fear location makes no sense at all, 10 = fear location makes perfect sense)

2. Rating explanations:

Finally, the fear plots and the following questions in Question set 4 were shown to participants:
Now assume the agent is done learning and replaced at the starting location of the maze (bottom middle). We show you a graphical representation of what the agent fears.
Do you think this distribution of fear over the locations makes sense? In other words, do you think it is logical for an agent in this environment to fear the locations that are most red more than the locations that are beige?
Pay attention to the relation between the locations, not so much to the actual values of the fear.

1. How plausible do you think the fear plot is?
Rate your answer on a scale 0-10. (0 = fear plot makes no sense at all, 10 = fear plot makes perfect sense)

2. Rating explanations:

Selection of participants

Participants in the evaluation were recruited from Amazon Mechanical Turk(MTurk). The reward for participation is 1$, and each participant was only allowed to take part in the survey once. To ensure the reliability of the results, the selection criteria on MTurk for participants were set as following: (Completed assignments $\geq$ 500, Approve rate for assignments $\geq$ 97%, Location: U.S.). No other special demographic requirements were chosen to reduce the bias from users background knowledge, for example, a user with HRI background might have a different focus than a normal user.

Chapter 6

Results and discussion

The surveys were collected and analyzed for 237 participants in total, each fear calculation method version was taken by about 33 different participants, the details are shown in Table 6.1. In general, most participants are in the age range of 25-44, with college background and even distribution among genders.(see Figure 6.1 and 6.2). The layout and questions in the survey were structured to evaluate different fear calculation methods from various aspects. In this section, the survey results will be presented with some participants comments about their observations.

Version	# participants
1	38
2	34
3	35
4	33
5	28
6	36
7	33

Table 6.1: Number of participants for each version of fear calculation method

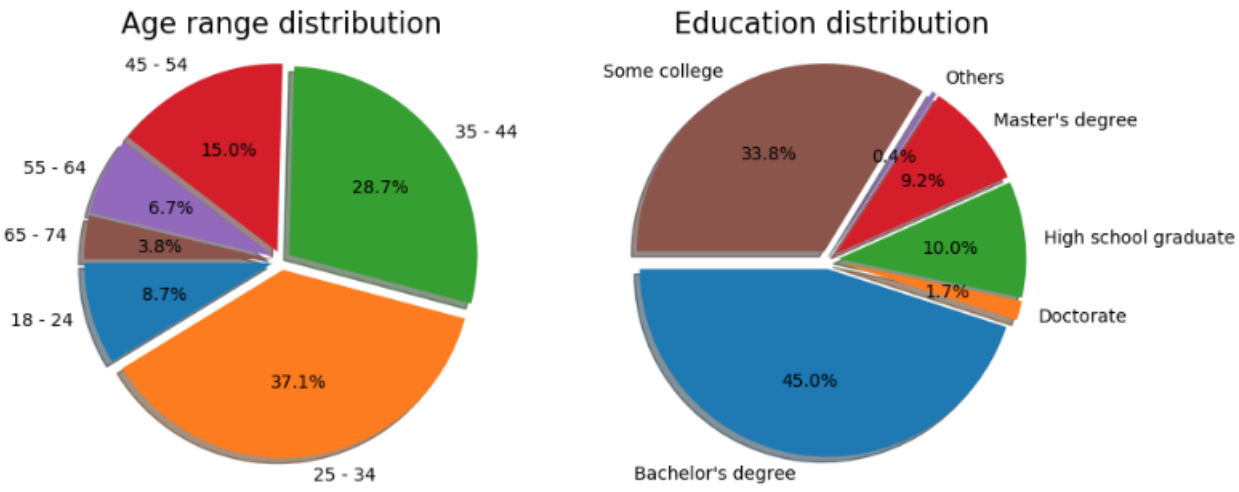


Figure 6.1: Age and education distribution for participants

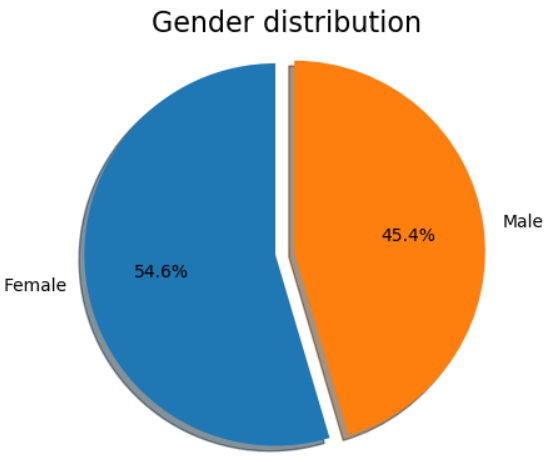


Figure 6.2: Gender distribution for participants

6.1 General observation

The first question in Question set 1 is about how participants observe the agent's learning process in general. Most participants observed the agent's exploration behaviour in the initial phase, after some struggle, the agent learned to avoid the center positions with popping green block and took path to the target with a detour. Participants wrote for example:

"It seems to be learning. Initially, it would generally go in one direction, but it would have a higher intensity. Later, it changed and started going in the opposite direction (right). Then it would switch between going left and right in direction to be more successful in avoiding the green box obstacle."

"It took a little bit for an agent to learn how to get to the target but it experienced high emotions every time it get bumped by a green block until the end."

"In the first part of the video, there seems to be a lot of learning behavior as the agent realizes there must be a moving obstruction in the middle of the map. There was some confusion as it wondered why it was able to move or not move into spaces it had before (due to the moving green square). However, it very quickly learned the "correct" path after some moving around and experimenting. It discovered the path of the green object and eventually came up with a way to avoid it"

"The agent tries to reach its goal by all means. When the green blocks arises in front of it, it finds a way to go round and reach its goal."

For the second question about emotion guess in Question set 1, the overall result for all participants is shown in Figure 6.3. According to this result, it seems the most recognized emotion is anger based on perceived behaviour in general. This is also the case for each of the 7 fear calculation methods(see all plots in Appendix A.1). For the 6 non-baseline calculation methods, a multinomial logistic regression analysis is performed to check whether there is a significant effect of fear policy or horizon affecting people's perceived emotion. Figure 6.4 shows the model fitting information, the p-value(0.538) is larger than 0.05. This means the model does not fit the data significantly better than the null model. Figure 6.5 shows the results for likelihood ratio tests, the p-values for both fear policy(0.292) and horizon(0.872) are larger than 0.05. This means neither fear policy nor horizon has a significant overall association with people's perceived emotion.

Among the participants who think the emotion is anger, their descriptions about the agent's behaviour in the previous question usually are about the agent tries to reach the target but keeps

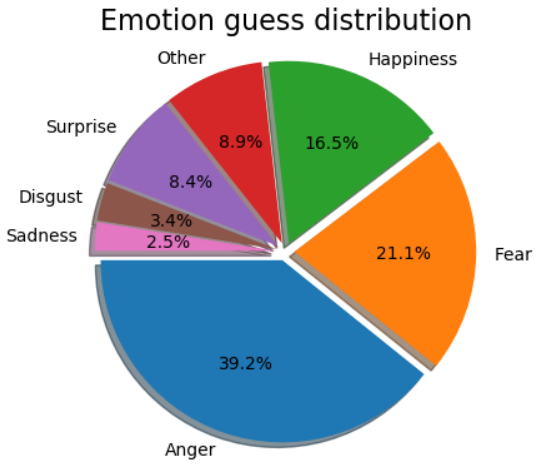


Figure 6.3: Emotion type guess distribution

Model Fitting Information

Model	Model Fitting Criteria	Likelihood Ratio Tests		
	-2 Log Likelihood	Chi-Square	df	Sig.
Intercept Only	117.947			
Final	101.169	16.777	18	0.538

Figure 6.4: Model fitting information for multinomial logistic regression analysis on emotion guess

Likelihood Ratio Tests

Effect	Model Fitting Criteria	Likelihood Ratio Tests		
	-2 Log Likelihood of Reduced Model	Chi-Square	df	Sig.
Intercept	101.169[a]	0.000	0	
Fear_policy	115.305	14.136	12	0.292
Horizon	103.634	2.464	6	0.872

The chi-square statistic is the difference in -2 log-likelihoods between the final model and a reduced model. The reduced model is formed by omitting an effect from the final model. The null hypothesis is that all parameters of that effect are 0.

a. This reduced model is equivalent to the final model because omitting the effect does not increase the degrees of freedom.

Figure 6.5: Likelihood ratio tests for multinomial logistic regression analysis on emotion guess

blocked by the green square and the agent seems frustrated in the initial phase. Participants wrote for example:

"The agent continually getting blocked by a green square, but also sometimes making it around the green square. The agent is showing intense emotion."

"It gets angry when it hits an obstacle. It learns as it goes."

"It seemed to be confused and frustrated at first when it reached the center part with the moving green block. It eventually appeared to learn where the walls were and the range of the green block's movement though."

"At first, the agent struggled to figure out how to get to the blue block, but after it figured it out the first time, it seemed to find it faster the next few times. It is definitely experiencing intense emotions frequently, maybe anger? or fear?"

"The agent is very persistent and always tries to overcome the obstacle and teach the target. However, it seems very irritated and frustrated by the process and turns red frequently out of frustration."

"The agent is angry when he is bumping into the obstacle in the middle. He is always attempting to get toward the blue area at the top of the puzzle."

"I feel likke the block is trying to move up but keeps getting caught up"

For the participants who think the emotion is fear, their answers for the previous question usually focus on describing the agent tries to avoid the center part and the agent's emotion gets more intense when close to the green square. Participants wrote for example:

"It turns red when it gets close to the green block but it learned a path around the green block"

"The agent becomes red when it gets close to the green block."

"The agent appears to show high levels of fear when it is near the green block."

"The agent is trying to go around the green box. The level of intensity increases and decreases with where the green box is"

"It would change color when it came into contact with the green square."

For the third question about guessing the ghost's function in Question set 1, the overall result for all participants is shown in Figure 6.6, majority of the participants believe the green block is not beneficial for the agent. This is also the case for each of the 7 fear calculation methods(see all plots in Appendix A.2). For the 6 non-baseline calculation methods, a multinomial logistic regression analysis is performed to check whether there is a significant effect of fear policy or horizon affecting people's perception about the ghost's function. Figure 6.7 shows the model fitting information, the p-value(0.814) is larger than 0.05. This means the model does not fit the data significantly better than the null model. Figure 6.8 shows the results for likelihood ratio tests, the p-values for both fear policy(0.712) and horizon(0.698) are larger than 0.05. This means neither fear policy nor horizon has a significant overall association with people's perception about the ghost's function.

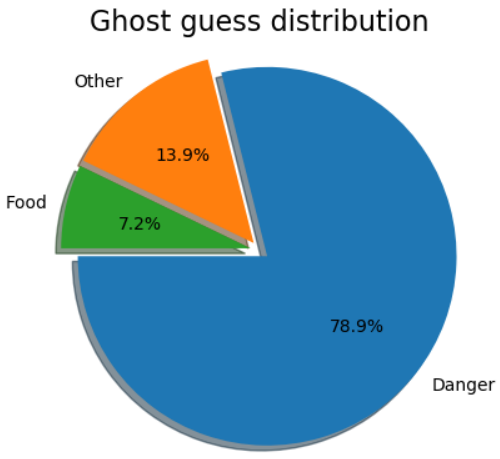


Figure 6.6: Ghost's function guess distribution

Model Fitting Information

Model	Model Fitting Criteria	Likelihood Ratio Tests		
	-2 Log Likelihood	Chi-Square	df	Sig.
Intercept Only	42.873			
Final	39.912	2.961	6	0.814

Figure 6.7: Model fitting information for multinomial logistic regression analysis on the ghost's function guess

Likelihood Ratio Tests

Effect	Model Fitting Criteria	Likelihood Ratio Tests		
	-2 Log Likelihood of Reduced Model	Chi-Square	df	Sig.
Intercept	39.912[a]	0.000	0	
Fear_policy	42.040	2.128	4	0.712
Horizon	40.630	0.719	2	0.698

The chi-square statistic is the difference in -2 log-likelihoods between the final model and a reduced model. The reduced model is formed by omitting an effect from the final model. The null hypothesis is that all parameters of that effect are 0.

a. This reduced model is equivalent to the final model because omitting the effect does not increase the degrees of freedom.

Figure 6.8: Likelihood ratio tests for multinomial logistic regression analysis on the ghost's function guess

6.2 Fear intensity plausibility

For the fear intensity plausibility rating in Question set 2, the standard deviation with 95% confidence for each version of the method is calculated. And it is calculated using the following formula:

$$\bar{x} \pm Z * \frac{\sigma}{\sqrt{n}} \tag{6.1}$$

where $\bar{x}$ is the mean, Z is 1.96 for 95% confidence, σ is the standard deviation and n is the size of the sample.

Table 6.2 shows the means and confidence intervals of the ratings, the lowest rating is the baseline V4. Figure 6.9 shows these ratings graphically. Table 6.3 shows the T-test results of null hypotheses between the 6 non-baseline calculation methods and the baseline. The p-values of V3(ϵ-greedy fear policy with $\epsilon = 1$ and short-horizon), V5(ϵ-greedy fear policy with $\epsilon = 0.1$ and long-horizon), and V6(softmax fear policy with $\tau = 5.0$ and long-horizon) are smaller than 0.05, thus those 3 null hypotheses of equal ratings with the baseline are rejected, there are significant differences between those calculation methods and the baseline. Figure 6.10 shows 2-way ANOVA analysis about the non-baseline ratings for how fear policy and horizon affect the ratings. For the following 3 hypotheses:

- H_1: Fear policy will have no significant effect on fear intensity plausibility ratings
- H_2: Horizon will have no significant effect on fear intensity plausibility ratings
- H_3: Fear policy and horizon will have no significant effect on fear intensity plausibility ratings

All their p-values are larger than 0.05($p_1 = 0.691$, $p_2 = 0.400$, $p_3 = 0.202$), thus all 3 hypothesises are not rejected. Fear policy, horizon or the combination of those two have no significant effect on fear intensity plausibility ratings.

In participants explanations for low ratings(0-3), the reasons are usually the same, except for the V4 baseline. In the V4 baseline, participants gave low ratings because they think the fear emotion seems random, for example:

"The fear intensity seemed very random and did not coincide with the movements of the agent."

"For me to think that the color intensity was fear, I'd need to see it change color each time it got near the green square. It only does so randomly. Maybe the green square is more scary at certain times, but I don't see any subtle changes either showing a slight fear."

In other calculation methods, participants' low rating explanations usually describe the agent shows fear before or after hitting the ghost without avoiding it and fear should decrease to zero over time. Participants wrote for example:

"It doesnt make a lot of sense that its fear, because it only turns red before it does something wrong"

"They should not be fearful as the game progresses. They should be getting less fearful, and eventually to almost no fear at all."

"It seemed to have a higher intensity whenever it came upon an obstacle that wasn't the green block, like when it ran into a wall. I would equate this to like you're in a dark fun house and can't see anything but can only feel your way around."

"It does not really avoid the green block"

"As time went on, the agent learned to avoid the block and reach the target. If the fear as its emotion made sense, as the agent learned to avoid punishment and receive more reward, its level of fear should decrease for each reward received but during the video, this did not happen. The fear experienced kept on being repeated each time the agent received a reward. This is not logical."

Version	Mean	Confidence Interval
1	6.63	5.81-7.45
2	5.88	5.05-6.72
3	6.91	6.08-7.75
4 (baseline)	5.52	4.56-6.47
5	7.0	6.10-7.90
6	6.97	6.07-7.87
7	6.39	5.45-7.34

Table 6.2: Ratings for fear intensity plausibility

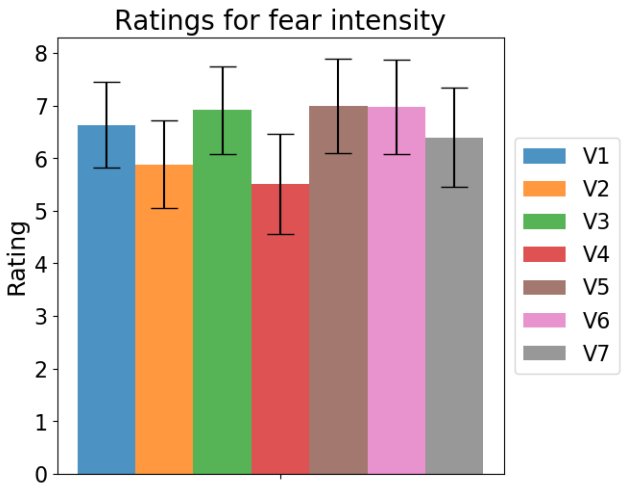


Figure 6.9: Ratings for fear intensity plausibility

	T-statistic	2 tailed p-value
V1 vs V4	1.71761	0.09057
V2 vs V4	0.56037	0.57720
V3 vs V4	2.13407	0.03665
V5 vs V4	2.19028	0.03247
V6 vs V4	2.14997	0.03521
V7 vs V4	1.26438	0.21068

Table 6.3: Fear intensity plausibility ratings T-test results

Tests of Between-Subjects Effects

Dependent Variable: Ratings

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	31.721[a]	5	6.344	0.913	0.474
Intercept	8891.888	1	8891.888	1279.534	2.34E-88
Fear_policy	5.138	2	2.569	0.370	0.691
Horizon	4.940	1	4.940	0.711	0.400
Fear_policy * Horizon	22.396	2	11.198	1.611	0.202
Error	1375.965	198	6.949		
Total	10368.000	204			
Corrected Total	1407.686	203			

a. R Squared = .023 (Adjusted R Squared = -.002)

Figure 6.10: Fear intensity plausibility ratings 2-way ANOVA results about fear policy and horizon

6.3 Fear location plausibility

Participants' responses about the plausibility of fear locations are shown in Table 6.4 and Figure 6.11. For the fear location plausibility ratings, again, the baseline V4 has the lowest rating score. The V5(ϵ-greedy fear policy with $\epsilon = 0.1$ and long-horizon) is considered as the most plausible one with a mean score of 7.04. Though the difference between it and the second highest rating is not significant(p-value = 0.3 for T-test). Table 6.5 shows the T-test results of null hypotheses between the 6 non-baseline calculation methods and the baseline. The only p-value larger than 0.05 is V6(softmax fear policy with $\tau = 5.0$ and long-horizon), thus the null hypothesis of equal ratings between V6 and the baseline is not rejected, and there are significant differences between the rest non-baseline calculation methods and the baseline. Figure 6.12 shows 2-way ANOVA analysis about the non-baseline ratings for how fear policy and horizon affect the ratings. For the following 3 hypotheses:

- H_1: Fear policy will have no significant effect on fear location plausibility ratings
- H_2: Horizon will have no significant effect on fear location plausibility ratings
- H_3: Fear policy and horizon will have no significant effect on fear location plausibility ratings

All their p-values are larger than 0.05($p_1 = 0.305$, $p_2 = 0.930$, $p_3 = 0.286$), thus all 3 hypotheses are not rejected. Fear policy, horizon or the combination of those two have no significant effect on fear location plausibility ratings.

In the non-baseline fear calculation methods, most low ratings(0-3) or medium ratings(4-7) are given for about the same reason, the agent sometimes doesn't show fear locations in the center(this is observed in all non-baseline methods), especially the fear locations in the start hallway or target hallway makes no sense. Participants wrote for example:

"Some of the time the fear location made sense, because it was around the center area, and other times it made no sense at all, like when it was near the corner edges nowhere near the center."

"The corners I don't believe make a lot of sense, however, the middle I totally get."

"It feels fear as it navigates the area and that is justified. however, it also feels fear as it gets close to the reward and that is odd."

"Because it did not make sense that the agent experienced fear at the very beginning and at the very end where the target was because there was no moving green block and it must have learned and figured that out. All the middle section is reasonable."

"Some of the fear locations make sense, where as others don't. If the agent learns where the walls are, then it doesn't make sense for it to fear a location it has already learned is an acceptable move. Especially the entrance to the target "hallway" should not be feared. It also doesn't make sense it would decide to move into a space it was afraid of."

"I feel like it makes sense while the green block is there, but not otherwise. Maybe the blue block represents death, and the block is reincarnated whenever it reaches it? That would explain why he is afraid of the hallway to the blue block, even though it is clearly safer there than in the more open square plot."

"Why would it feel fear when it is closing in on the target or when it is starting out?"

"It doesn't make sense that there would be fear in the beginning hallway section because there is no block or anything in the way or anything threatening to the agent here. It does make sense for it to feel fear toward the middle because that is where the block is."

"I don't understand why the regions at the start and end would produce fear. Also, it seemed fairly arbitrary at times, as the outer border areas also caused 'fear'."

Version	Mean	Confidence Interval
1	6.08	5.21-6.95
2	6.06	5.09-7.02
3	6.23	5.15-7.31
4 (baseline)	4.39	3.39-5.40
5	7.04	6.00-8.08
6	5.5	4.56-6.44
7	5.94	5.02-6.86

Table 6.4: Ratings for fear location plausibility

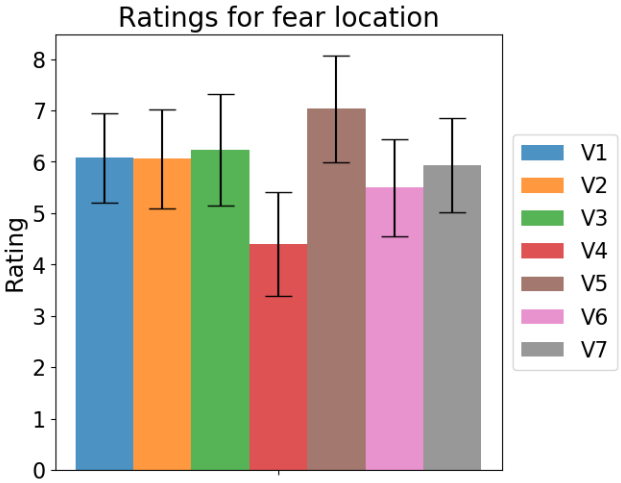


Figure 6.11: Ratings for fear location plausibility

	T-statistic	2 tailed p-value
V1 vs V4	2.44449	0.01719
V2 vs V4	2.30421	0.02443
V3 vs V4	2.39450	0.01950
V5 vs V4	3.51490	0.00086
V6 vs V4	1.54652	0.12675
V7 vs V4	2.18610	0.03250

Table 6.5: Fear location plausibility ratings T-test results

Tests of Between-Subjects Effects

Dependent Variable: Ratings

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	38.967[a]	5	7.793	0.907	0.477
Intercept	7621.211	1	7621.211	887.302	4.43E-75
Fear_policy	20.516	2	10.258	1.194	0.305
Horizon	0.066	1	0.066	0.008	0.930
Fear_policy * Horizon	21.627	2	10.814	1.259	0.286
Error	1700.660	198	8.589		
Total	9350.000	204			
Corrected Total	1739.627	203			

a. R Squared = .022 (Adjusted R Squared = -.002)

Figure 6.12: Fear location plausibility ratings 2-way ANOVA results about fear policy and horizon

6.4 Fear plot plausibility

Table 6.6 and Figure 6.13 show participants ratings for different fear plots. It is apparent that fear calculation method V5(ϵ-greedy fear policy with $\epsilon = 0.1$ and long-horizon) is regarded as the most plausible one with a mean score of 8, while the rest methods have rating scores around 4. Table 6.7 shows the T-test results of null hypotheses between the 6 non-baseline calculation methods and the baseline. The only p-value smaller than 0.05 is V5, thus the null hypothesis of equal ratings between V5 and the baseline is rejected, and there are no significant differences between the rest non-baseline methods and the baseline. Figure 6.14 shows 2-way ANOVA analysis about the non-baseline ratings for how fear policy and horizon affect the ratings. For the following 3 hypotheses:

- H_1: Fear policy will have no significant effect on fear plot plausibility ratings
- H_2: Horizon will have no significant effect on fear plot plausibility ratings
- H_3: Fear policy and horizon will have no significant effect on fear plot plausibility ratings

All their p-values are smaller than 0.05($p_1 = 8.62E-06$, $p_2 = 2.29E-05$, $p_3 = 1.94E-04$), thus all 3 hypotheses are rejected. The combination of fear policy and horizon has a significant effect on fear plot plausibility ratings. And this can be explained by the effect of ϵ-greedy fear policy($\epsilon = 0.1$) with a long-horizon.

Figure 6.15 shows the percentage of low rating(0-3), medium rating(4-7) and high rating(8-10) for fear plots with each method. In fear calculation method V5(ϵ-greedy fear policy with $\epsilon = 0.1$ and long-horizon), participants mostly think that the plot makes a lot of sense because the 3 most feared positions are the ghost positions. Participants wrote for example:

"It very plausible because the fear plot intensifies as the agent gets to the location of the previous punishment and danger. "

"yes, based on the agent's past experience, it realizes that's where the trouble lies, in those fear zones."

For other non-baseline plots, the low ratings are because participants believe the most feared locations should be where the ghost appeared, instead of the beginning hallway. Participants wrote for example:

"There is only fear reaction at the beginning, before being blocked, hence the fear is irrational, and nonsensical."

"The fear is only located at the beginning, not in the the main section."

"It doesn't make sense for the agent to fear the safest spot in the plot, at the start position."

"It makes no logical sense for the agent to fear the starting position which has always been safe, and not fear any of the rest of the board, which it knows could have danger or punishment."

As for the participants gave high ratings in other non-baseline plots, they either think the agent feared to start over or think fear come from the unknown in the beginning. Participants wrote for example:

"I think it's logical because the agent keeps ending up at the same point, even after accomplishing the goal of reaching the blue tile. The agent is confused and doesn't know how to progress."

"It fears having to start over at the beginning."

"This is the point they are starting out and have no knowledge, so yes this makes sense"

"The unknown can create a lot of fear so when you begin a task you can have a lot of fear."

"It makes sense that they fear the most at the beginning because of the fear of the unknown"

"Right before the agent started is the most feared spot, it is plausible because it may not know what to expect."

The fear plot for the baseline V4 is a special one(see Appendix Figure A.13) because it is plotted by generating random numbers in all positions. Therefore, some participants regard is as completely random and makes no sense at all, others think it makes sense because many fear locations are corners or on the way to the ghost. Participants wrote for example:

"If it is learning that the green block in the middle is punishment, it would fear those squares the most."

"it doesn't seem plausible because the red areas seem random"

"There is a lot of fear felt right at the target area, when I don't see any reason for that. There is not a lot of fear in the middle area where the green block hangs out, and I feel there should be."

"It makes a lot of sense this way, the point of reentry and the corners cause more fear."

"Corners, starting out, and possibly entering a long unknown area are points where someone would feel the most intense fear. So by that logic, I definitely think it is plausible. "

"In those corners that have a higher intensity of fear, the path leading up to the green block pathway and the path to the finish make sense as to why those would be higher in fear. The agent feared going up to the green block, feared going around, and then feared going up the finish as it most likely had a "something is following me" feeling."

Version	Mean	Confidence Interval
1	3.63	2.68-4.58
2	3.03	1.92-4.14
3	3.54	2.56-4.53
4 (baseline)	4.03	3.12-4.94
5	8.0	7.20-8.80
6	3.89	2.87-4.91
7	3.82	2.82-4.82

Table 6.6: Ratings for fear plot plausibility

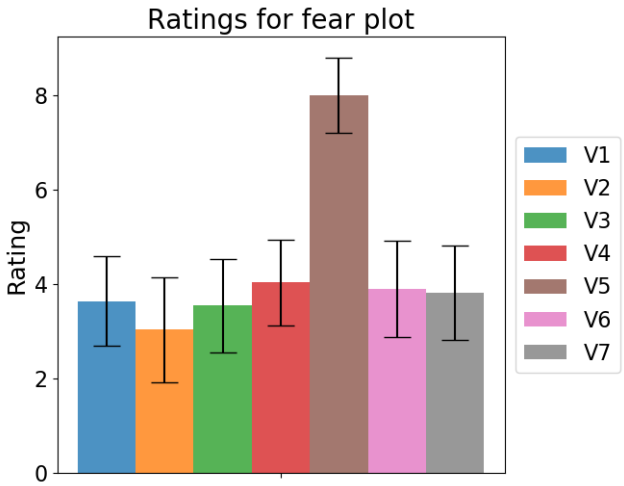


Figure 6.13: Ratings for fear plot plausibility

	T-statistic	2 tailed p-value
V1 vs V4	-0.58566	0.56002
V2 vs V4	-1.34360	0.18391
V3 vs V4	-0.70095	0.48580
V5 vs V4	6.32051	3.8E-08
V6 vs V4	-0.19969	0.84233
V7 vs V4	-0.30232	0.76340

Table 6.7: Fear plot plausibility ratings T-test results

Tests of Between-Subjects Effects

Dependent Variable: Ratings

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	486.581[a]	5	97.316	10.771	3.57E-09
Intercept	3769.781	1	3769.781	417.234	1.23E-50
Fear_policy	223.637	2	111.818	12.376	8.62E-06
Horizon	170.053	1	170.053	18.821	2.29E-05
Fear_policy * Horizon	161.283	2	80.642	8.925	1.94E-04
Error	1788.963	198	9.035		
Total	5859.000	204			
Corrected Total	2275.544	203			

a. R Squared = .214 (Adjusted R Squared = .194)

Figure 6.14: Fear plot plausibility ratings 2-way ANOVA results about fear policy and horizon

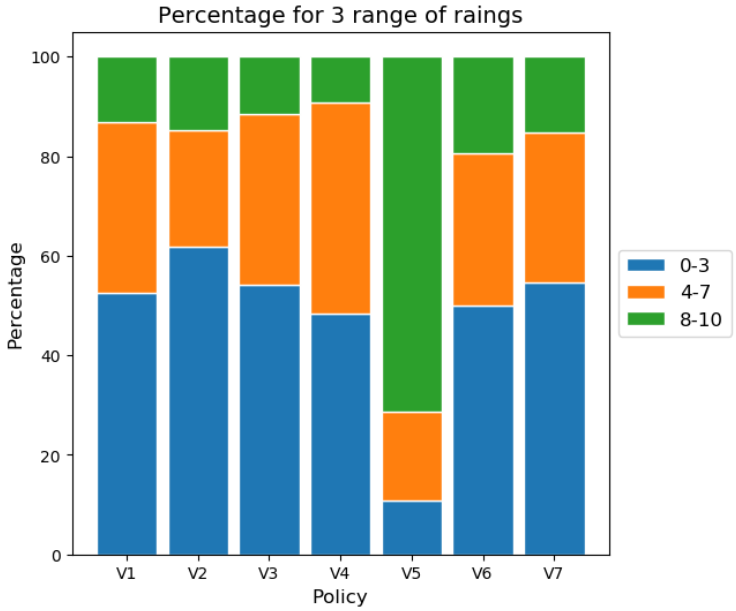


Figure 6.15: Percentage of 3 rating ranges for 7 fear plots

6.5 Discussion

6.5.1 General observation & fear intensity

Most participants were able to recognize the agent's learning behaviour, exploring the environment in the initial phase and avoid the ghost after some contacts with it. However, based on the overall results, it seems the majority of the participants interpreted the agent's movement behaviour and change in emotion intensity as anger. The reason for this is they observed the agent trying to reach the target but blocked by the ghost and emotion intensity increased at the same time. One possible cause for this anger interpretation might come from the difference in perceived risk probability for the agent and participants. For participants, they could always see there is a green square popping in one of the three locations in the center, so the existence and locations of the ghost are relatively certain for them. But for the agent, it only knows there is a small probability of a penalty when in a certain state and takes a certain action, therefore, the risk for the agent is uncertain and uncontrollable. According to the research of Smith and Ellsworth [49], both anger and fear are negative in valence, but these two emotions differ in terms of the certainty and control dimensions. Anger has more individual control(how individual influence the situation) and certainty, but fear is more about situational control(how the environment influences the situation) and uncertainly. Therefore, the agent's emotion, perceived by the participants as certain and self-controllable, is mostly regarded as anger.

After participants were informed about the punishment for hitting the ghost and the emotion was fear, fear calculation method V4(baseline) was regarded as the least plausible one for fear intensity. The ratings for V3(ϵ-greedy fear policy with $\epsilon = 1$ and short-horizon), V5(ϵ-greedy fear policy with $\epsilon = 0.1$ and long-horizon), and V6(softmax fear policy with $\tau = 5$ and long-horizon) have significant differences compared with the baseline. Among the low ratings(0-3) explanations, there are two common ones almost appear in all calculation methods: the agent shows fear before or after hitting the ghost without avoiding it and the ghost should show fear intensity decrease to zero over time. In TDRL Theory, fear is defined as a forward temporal difference, and distress is defined as immediate temporal difference. That's why the agent doesn't show fear the moment it hits the ghost. Sometimes the agent shows fear for the ghost, but still goes into that direction, the reasons for this can be twofold. First, in some calculation methods, the fear showed by the agent could be the fear for the wall it just hit instead of the ghost. Second, the action-selection policy is ϵ-greedy with ϵ decay from 0.5 to 0.1 for the initial 300 steps and keeps at 0.1 for the last 200 steps, so there is always some randomness in choosing an action. In the experiments, the agent gradually builds the model for the environment by interacting with it. The forward planning(or future imaginations) is based on the environment model. If in this model there is no knowledge about the consequence of taking an action in a certain state, then the agent won't be able to predict whether it should be afraid for that situation or not. This is why in some cases the agent seems to not have fear after just hitting a ghost. Humans have more information than the agent by just looking at the environment layout. For example, for a sequence during the exploration phase shown in Figure 6.16, the agent moved from position A to B and hit the ghost, then it moved down to position C but felt no fear for position B. This is because the agent had never taken the action "up" in position C, so it didn't know the consequence of moving up will end up in position B based on its knowledge. Therefore, it showed no fear for position B even just hit the ghost in there. During experiments, the random parameter for the fear policy was kept at a constant, 0.1 or 1 for ϵ-greedy and 5 for softmax. This random parameter decides the probability of randomly choosing an action in a forward planning tree. So when the agent approached converge in the later phase, there is always some fear due to this randomness. This explains why participants didn't see fear decreased to zero in the videos.

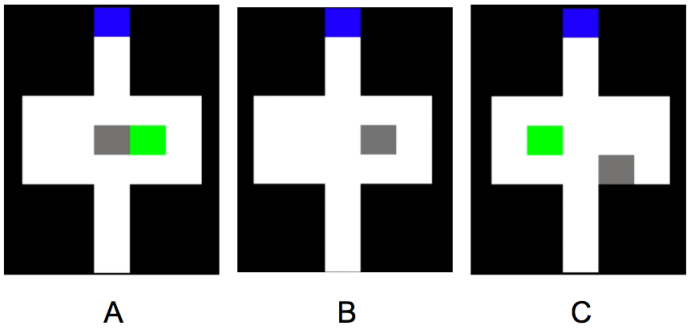


Figure 6.16: The agent doesn't have fear after hitting the ghost. In this sequence, the agent in A position move right hit the ghost in B position, and then move down to C position.

6.5.2 Fear location & fear plot

Both fear location video and fear plot check the plausibility of fear locations for different fear calculation methods. The difference is that the former one shows the agent's most feared location during the entire video, while the later one shows all fear locations and values when the agent is situated in start position after 500 steps of learning. In theory, the rating results for these two should be similar. However, there is a significant difference between these two rating results. For the fear location ratings, the highest is V5(ϵ-greedy fear policy with $\epsilon = 0.1$ and long-horizon) with a mean score of 7.04, other non-baseline methods are around 6. For the fear plot ratings, the highest is V5 with a mean score of 8, and the rest are around 4.

This difference in ratings might be explained by how fear is expressed in videos and plots. In fear plots, participants had an overview of all feared locations and corresponding fear values at the same time. In fear location videos, there were no overviews and it's more difficult for participants to analyze in real-time. For the fear plots, the effects of fear policy and horizon are obvious with an overview, most participants were able to recognize the most feared locations should be the ghost's positions and gave low ratings for non-V5 calculations methods(See participants' comments in Section 6.4 and Figure 6.15). The comparison of Figure A.10(V1, ϵ-greedy fear policy with $\epsilon = 0.1$ and short-horizon) and A.14(V5, ϵ-greedy fear policy with $\epsilon = 0.1$ and long-horizon) shows the effect of horizon. Both fear plots use the same fear policy, the only difference is the former one can only imaging 2 steps away(short-horizon) but the later one can imaging 10 steps away(long-horizon). For the V1 fear plot, the most feared location is its position, since it can only imagine a short future and most futures are about staying in that place after hitting the wall. For the V5 fear plot, the agent's most feared positions are the center positions because it could image that far away. The comparison of Figure A.14(V5, ϵ-greedy fear policy with $\epsilon = 0.1$ and long-horizon), A.15(V6, softmax fear policy with $\tau = 5.0$ and long-horizon) and A.16(V7, ϵ-greedy fear policy with $\epsilon = 1$ and long-horizon) shows the effect of different fear policies. These three fear plots use the same horizon but different in the fear policy. In both Figure A.15 and A.16, the most feared position is the agent's own position because of implausible fear policy.

In fear location videos, the effects of fear policy and horizon were more difficult to notice. For example, Figure 6.17(V1, ϵ-greedy fear policy with $\epsilon = 0.1$ and short-horizon) and 6.18(V5, ϵ-greedy fear policy with $\epsilon = 0.1$ and long-horizon) show fear locations of two different horizon agents after the same amount of learning when in exact positions. The differences are obvious in the form of static images, but in videos, those 3 frames were shown in just one second.

The small environment used for the agent's learning also diminishes the effect of different horizons in fear location videos to some extent. For the previous examples of Figure 6.17 and 6.18, the agent

will move up a step and take the right side path to the target after in position C. During the right path until the target hallway, those 2 calculation methods will have same fear locations because the distance between the center position and the agent is not larger than the distance of short-horizon.

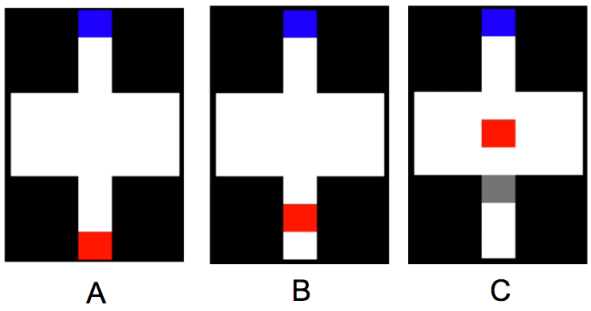


Figure 6.17: A sequence of fear V1(ϵ-greedy fear policy with $\epsilon = 0.1$ and short-horizon) shows the most feared location(marked in red) after some learning when in position A, B and C. In both positions A and B, the agent most feared location is its own location.

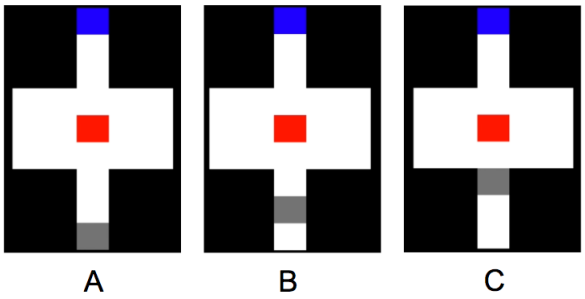


Figure 6.18: A sequence of fear V5(ϵ-greedy fear policy with $\epsilon = 0.1$ and long-horizon) shows the most feared location(marked in red) after some learning when in position A, B and C.

Although V5(ϵ-greedy fear policy with $\epsilon = 0.1$ and long-horizon) calculation method has the highest rating for both fear location and fear plot, there are some comments about its implausibility. For the fear location videos, the low rating comments are mainly about showing fear locations in the target hallway made them confusing. Figure 6.19 shows an example for this, in the presented sequence, the most feared locations for the agent on the way to target is shown for position A, B, and C. When the agent is in position A, according to its knowledge, there are many outcomes that could result in the center ghost position(e.g. move left, move down and move right). After the agent moves up in the target hallway, now the most feared location is its own place. This is because during the learning process it learned the best action in that position is to move up, and second-best actions are move left and move right. The worst action is to move down and according to the transition model the probability of choosing move down is pretty small(e.g. 1%), thus the fear for the ghost in the center is not as big as the penalty for hitting the wall. For the fear plot ratings, 3 participants gave low ratings because they believe there should be fear in more places based on previous videos.

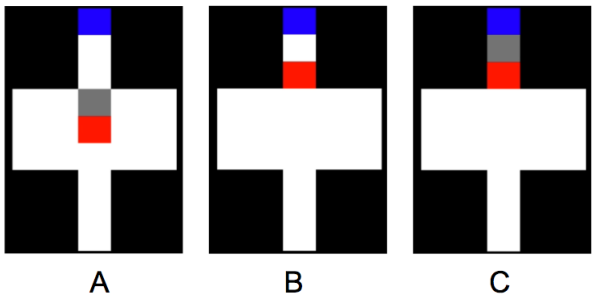


Figure 6.19: A sequence of fear V5(ϵ-greedy fear policy with $\epsilon = 0.1$ and long-horizon) shows the most feared location(marked in red) after some learning when in position A, B and C. Agent move up from A to B, then to C.

Chapter 7

Conclusions and Future work

7.1 Conclusions

Emotions can help humans better understand learning robots. The TDRL Theory of emotions proposed a computational model of emotions grounded in RL process. Preliminary results have shown hope and fear could be efficiently estimated in 3 simulation scenarios. But there is no study about the plausibility of the simulated emotions perceived by humans. In this thesis, 6 fear calculation methods(3 fear policies x 2 horizons) based on TDRL Theory of emotions are compared with a baseline to check: a) the plausibility of fear simulation perceived by humans b) the effect of different fear calculation methods for fear intensity and fear location. The forward planning for fear simulation was based on a modified version of MCTS-T+, which efficiently solves the problem of an asymmetric tree with loops.

In total, 237 participants were recruited from Amazon Mechanical Turk. On average, each of the 7 fear calculation methods had about 33 participants. According to participants' general observations for different methods of simulations without knowing the exact type of emotion, they believed the most probable emotion is anger. After analyzed participants' comments, the possible reason about why they thought the emotion is anger instead of fear is discussed in detail in Chapter 6.5.1. In short, it is possible that showing participants the presence of the ghost(green square) in the video affects their understanding of the agent's situations.

After participants were informed the emotion type is fear. The rating results for fear intensity plausibility suggest V3(ϵ-greedy fear policy with $\epsilon = 1$ and short-horizon), V5(ϵ-greedy fear policy with $\epsilon = 0.1$ and long-horizon), and V6(softmax fear policy with $\tau = 5.0$ and long-horizon) fear calculation methods are better than the baseline with a significant difference. From participants rating explanations that associated with low ratings, it seems the current fear expression from the simulation is not human intuitive enough and fear policy needs some more improvement.

The rating plausibility results from fear location videos and fear plots suggest V5 calculation method(ϵ-greedy fear policy with $\epsilon = 0.1$ and long-horizon) gives the most plausible fear among the rest. And the 2-ANOVA results of the fear plot suggest the combination of fear policy and horizon has a significant effect on the ratings. However, participants' comments about this method on fear location and fear plot also suggest that the expression of the most feared object can be improved by setting a threshold on fear values or other mapping functions.

In summary, fear calculation method with ϵ-greedy fear policy($\epsilon = 0.1$) and long-horizon provides a plausible fear estimation. And humans could understand simulated fear based on TDRL Theory of emotions when properly expressed.

7.2 Future work

This thesis provides interesting results for how humans perceive the simulated fear based on TDRL Theory of emotions. The followings could be possible improvements for a more plausible fear simulation or future research directions.

- In the first part of the results, most people think the emotion expressed by the agent is anger instead of fear. The possible cause could be the situation perceived by the agent and by the participant is different. Therefore, to properly investigate the plausibility of the simulated emotion, the experiment should be designed to make sure participants have the same perception about the environment as the agent.
- In the experiments, fear intensity expression is by first normalize the TD error for each calculation method and then expressed by color using a gray to red scale. From participants comments about the ratings, the current expression method is not natural. Maybe some other mapping function from TD error to emotional intensity and visual cue can be used for improvements.
- For the current implementation of fear simulation, fear doesn't affect the learning process. But, some participants believe the agent should always avoid the feared object. So one of the future research direction could be to test the agent with emotion affect action-selection policy.
- For all fear location videos, the most feared location is marked by red in the environment for all steps, even the fear value for that location is pretty small. This leads to some confusions for participants, especially when the agent shows fear locations in places they believe should be safe. One improvement for current fear location estimation could be to only show fear locations if the fear for that object exceeds a certain threshold.

Bibliography

[1] B. Hayes, B. Scassellati, and C. Stein. Challenges in shared-environment human-robot collaboration. *Collaborative Manipulation Workshop at HRI 2013*, 2013.

[2] R. S. Sutton and A. G. Barto. Reinforcement learning: An introduction. *Cambridge: MIT Press.*, 1998.

[3] T. Kuremoto, T. Tsurusaki, K. Kobayashi, S. Mabu, and M. Obayashi. An improved reinforcement learning system using affective factors. *Robotics, 2(3), 149-164*, 2013.

[4] T. Kollar and N. Roy. Trajectory optimization using reinforcement learning for map exploration. *Int. J. Robot. Res.*, 27:175–196, 2008.

[5] M. Obayashi, N. Nakahara, T. Kuremoto, and K. Kobayashi. A robust reinforcement learning using concept of slide mode control. *Artif. Life Robot*, 13:526–530, 2009.

[6] J. Broekens and M. Chetouani. Towards transparent robot learning through tdrl-based emotional expressions. *IEEE Transactions on Affective Computing*, 2019.

[7] T. Kim and P. Hinds. Who should i blame? effects of autonomy and transparency on attributions in human-robot interactions. *15th IEEE International Symposium on Robot and Human Interactive Communication*, pages 80–85, 2006.

[8] W. B. Knox, P. Stone, and C. Breazeal. Training a robot via human feedback: A case study. *In G. Herrmann, M. Pearson, A. Lenz, P. Bremner, A. Spiers, U. Leonards (Eds.), Social Robotics (Vol. 8239, pp. 460-470): Springer International Publishing.*, 2013.

[9] A. L. Thomaz and C. Breazeal. Reinforcement learning with human teachers: evidence of feedback and guidance with implications for learning performance. *Proceedings of the 21st national conference on Artificial intelligence - Volume 1. Boston, Massachusetts: AAAI Press.*, 2006.

[10] H. R. Kim and D. S. Kwon. Computational model of emotion generation for human-robot interaction based on the cognitive appraisal theory. *Journal of Intelligent Robotic Systems, 60(2), 263-283*, 2010.

[11] M. S. El-Nasr, J. Yen, and T. R. Ioerger. Flame-fuzzy logic adaptive model of emotions. *Autonomous Agents and Multi-agent Systems, 3(3), 219257*, 2000.

[12] M. Ficocelli, J. Terao, and G. Nejat. Promoting interactions between humans and robots using robotic emotional behavior. *IEEE Transactions on Cybernetics, 46(12), 29112923.*, 2016.

[13] J. Broekens. A temporal difference reinforcement learning theory of emotion: A unified view on emotion, cognition and adaptive behavior. *Emotion Review, submitted.*, 2018.

[14] P. Ekman. An argument for basic emotions. *Cognition and emotion*, 6(3-4):169–200, 1992.

[15] D. Pacella, M. Ponticorvo, O. Gigliotta, and O. Miglino. Basic emotions and adaptation. a computational and evolutionary model. *PLoS ONE 12(11)*, 2017.

[16] D. C. Rubin and J.M. Talerico. A comparison of dimensional models of emotion. *Memory.*, 17 (8):802–808, 2009.

[17] J. Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39 (6)(1161-1178), 1980.

[18] M. B. Arnold. Emotion and personality. *New York, NY: Columbia University Press*, 1960.

[19] R. S. Lazarus. Psychological stress and the coping process. *New York, NY: McGraw-Hill.*, 1966.

[20] A. Moors, P. C. Ellsworth, K. R. Scherer, and N. H. Frijda. Appraisal theories of emotion: State of the art and future development. *Emotion Review*, 5:119–124, 2013.

[21] E. Hudlicka. Computational analytical framework for affective modeling: Towards guidelines for designing. *In Psychology and Mental Health: Concepts, Methodologies, Tools, and Applications*, pages 1–64, 2016.

[22] N. H. Frijda. The emotions. *Cambridge: Cambridge University Press*, 1986.

[23] K. Oatley and P. N. Johnson Laird. Towards a cognitive theory of the emotions. *Cognition and Emotion*, 1:29–50, 1987.

[24] T. Moerland, J. Broekens, and C. M. Jonker. Emotion in reinforcement learning agents and robots: A survey. *Machine Learning, vol. 107, no. 2, p. 443480*, 2018.

[25] S. C. Gadanho and J. Hallam. Robot learning driven by emotion. *Adaptive Behavior*, 9(1):42–64, 2001.

[26] R. Marinier and J. E. Laird. Emotion-driven reinforcement learning. *Cognitive science*, pages 115–120, 2008.

[27] K. R. Scherer. Appraisal theory. *Handbook of cognition and emotion*, pages 637–663, 1999.

[28] H. Ahn and R. W. Picard. Affective cognitive learning and decision making: The role of emotions. *In EMCSR 2006: The 18th European meeting on cybernetics and systems research*, 2006.

[29] H. Zhang and S. Liu. Design of autonomous navigation system based on affective cognitive learning and decision-making. *In 2009 IEEE international conference on robotics and biomimetics (ROBIO)*, pages 2491–2496, 2009.

[30] C. P. Lee-Johnson, D. Carnegie, and et al. Mobile robot navigation modulated by artificial emotions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 40(2):469–480, 2010.

[31] X. Shi, Z. Wang, and Q. Zhang. Artificial emotion model based on neuromodulators and q-learning. *In W. Deng (Ed.), Future Control and Automation: Proceedings of the 2nd International Conference on Future Control and Automation (ICFCA 2012)*, 1:293–299, 2012.

[32] I. Cos, L. Caamero, G. M. Hayes, and A. Gillies. Hedonic value: Enhancing adaptation for motivated agents. *Adaptive Behavior*, 21(6):465–483, 2013.

[33] N. Goerke. Emobot: A robot control architecture based on emotion-like internal values. *Rijeka: INTECH Open Access Publisher.*, 2006.

[34] J. Broekens, W. A. Kosters, and F. J. Verbeek. On affect and self-adaptation: Potential benefits of valence-controlled action-selection. *Bio-inspired modeling of cognitive tasks*, pages 357–366, 2007.

[35] E. Jacobs, J. Broekens, and C. M. Jonker. Emergent dynamics of joy, distress, hope and fear in reinforcement learning agents. *In Adaptive learning agents workshop at AAMAS2014*, 2014.

[36] A. Ortony, Gerald L. Clore, and A. Collins. The cognitive structure of emotions. *Cambridge University Press*, 1988.

[37] L Alan Sroufe. Emotional development: The organization of emotional life in the early years. *Cambridge University Press*, 1997.

[38] K. M. Myers and M. Davis. Mechanisms of fear extinction. *Mol Psychiatry*, 12(2):120–150, 2006.

[39] F. Tanaka, K. Noda, T. Sawada, and M. Fujita. Associated emotion and its expression in an entertainment robot qrio. *Entertainment computingICEC 2004*, pages 499–504, 2004.

[40] M. B. Moussa and N. Magnenat-Thalmann. Toward socially responsible agents: Integrating attachment and learning in emotional decision-making. *Computer Animation and Virtual Worlds*, 24(3-4):327–334, 2013.

[41] T. Moerland, J. Broekens, and C. M. Jonker. Fear and hope emerge from anticipation in model-based reinforcement learning. *In Proceedings of the international joint conference on artificial intelligence (IJCAI)*, pages 848–854, 2016.

[42] R. Coulom. Efficient selectivity and backup operators in monte-carlo tree search. *In International conference on computers and games,Springer.*, pages 72–83, 2006.

[43] L. Kocsis and C. Szepesvri. Bandit based monte-carlo planning. *In ECML,Springer.*, 6:289–293, 2006.

[44] T. Cazenave and N. Jouandeau. On the parallelization of uct. *In proceedings of the Computer Games Workshop*, pages 93–101, 2007.

[45] T. M. Moerland, J. Broekens, A. Plaat, and C. M. Jonker. Monte carlo tree search for asymmetric trees. *arXiv preprint arXiv:1805.09218*, 2018.

[46] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

[47] Rainer Reisenzein. Emotional experience in the computational belief-desire theory of emotion. *Emotion Review*, 1(3):214–222, 2009.

[48] OpenAI Gym. *https://gym.openai.com*, 2015.

[49] C. A. Smith and P. C. Ellsworth. Patterns of cognitive appraisal in emotion. *Journal of Personality and Social Psychology*, 48:813–838, 1985.

Appendix A

Appendix

A.1 Emotion guess for 7 different fear calculation methods

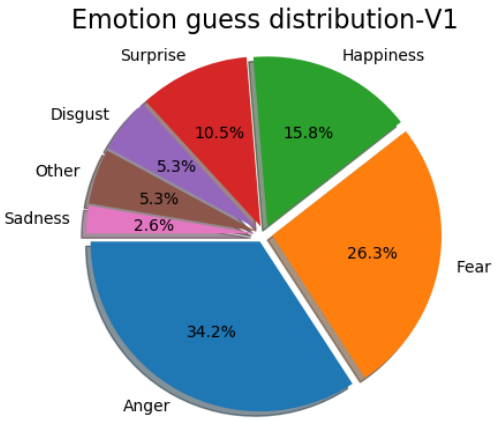


Figure A.1: Emotion type guess distribution for V1(ϵ-greedy fear policy with $\epsilon = 0.1$ and short-horizon)

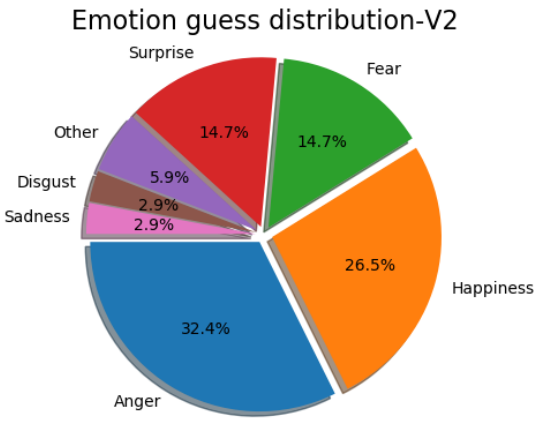


Figure A.2: Emotion type guess distribution for V2(softmax fear policy with $\tau = 5.0$ and short-horizon)

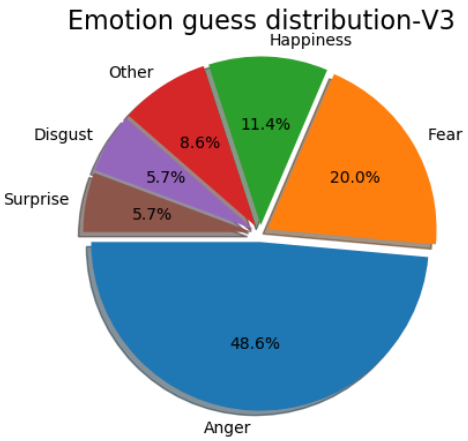


Figure A.3: Emotion type guess distribution for V3(ϵ-greedy fear policy with $\epsilon = 1$ and short-horizon)

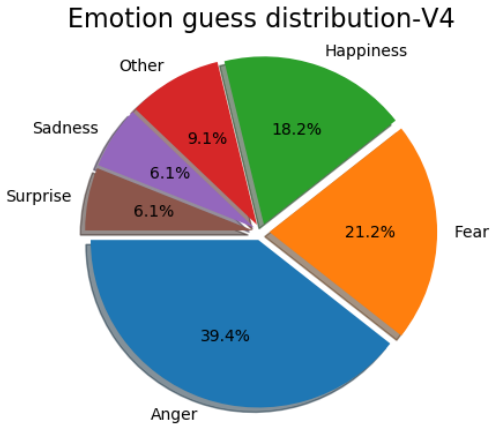


Figure A.4: Emotion type guess distribution for V4(baseline)

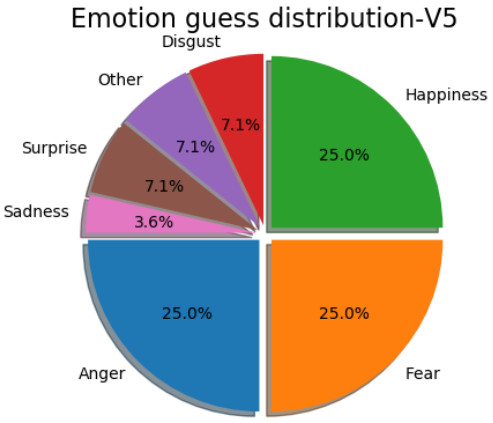


Figure A.5: Emotion type guess distribution for V5(ϵ-greedy fear policy with $\epsilon = 0.1$ and long-horizon)

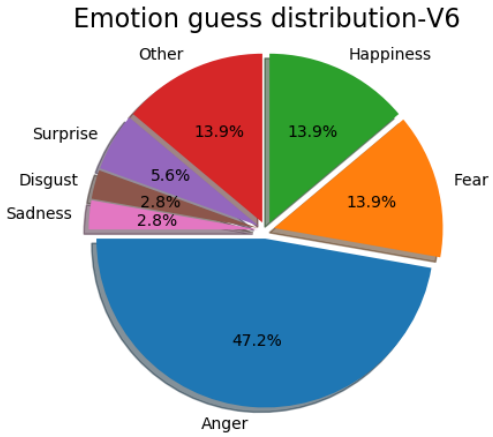


Figure A.6: Emotion type guess distribution for V6(softmax fear policy with $\tau = 5.0$ and long-horizon)

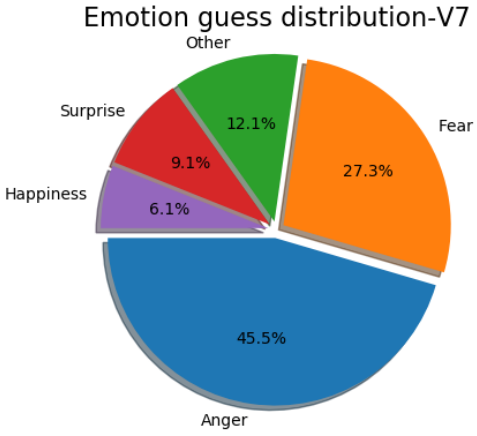


Figure A.7: Emotion type guess distribution for V7(ϵ-greedy fear policy with $\epsilon = 1$ and long-horizon)

A.2 Ghost guess for 7 different fear calculation methods

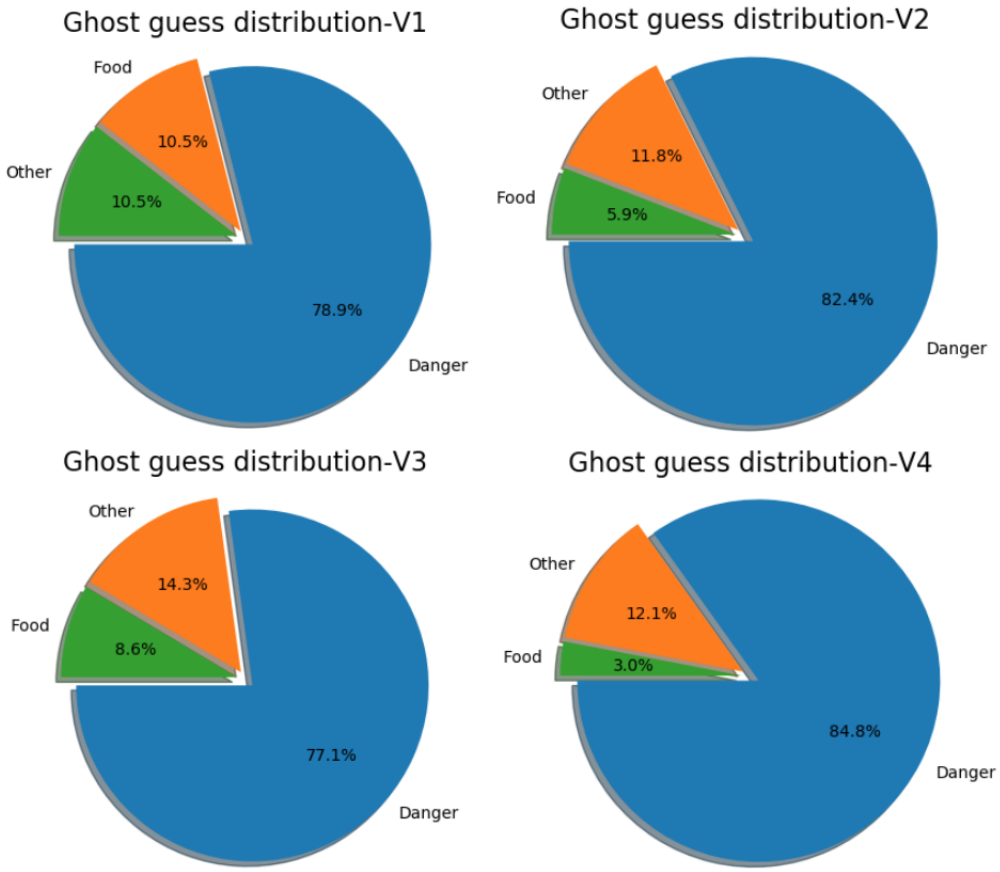


Figure A.8: Ghost guess distribution for V1, V2, V3 and V4

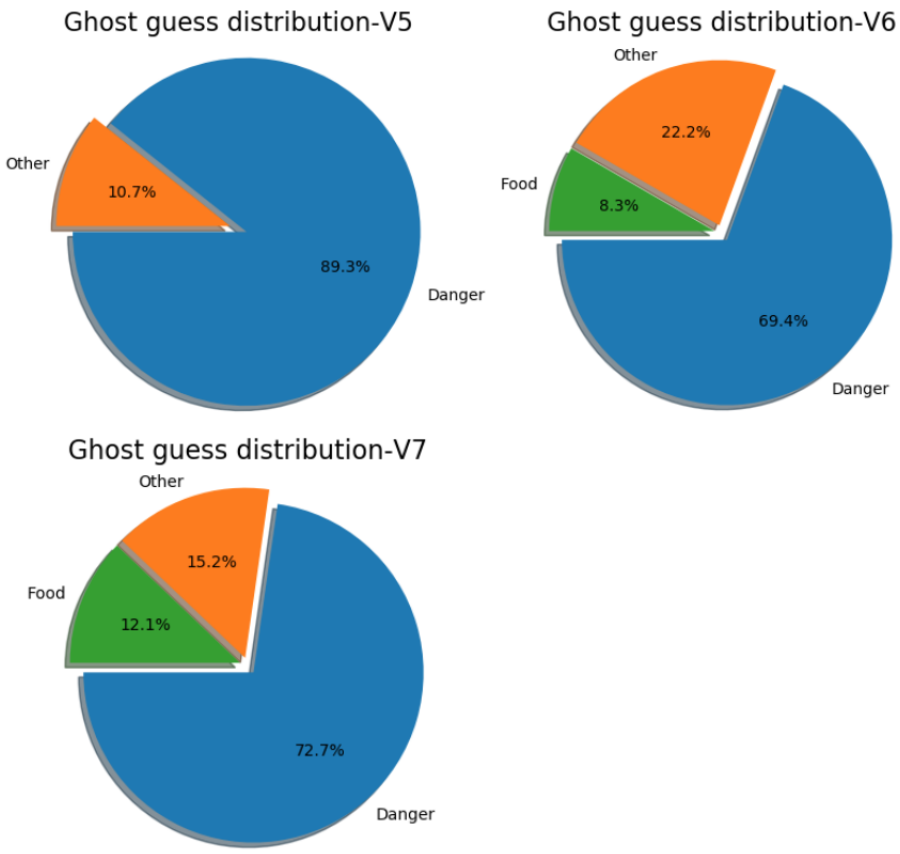


Figure A.9: Ghost guess distribution for V5, V6 and V7

A.3 Fear plot for 7 different fear calculation methods

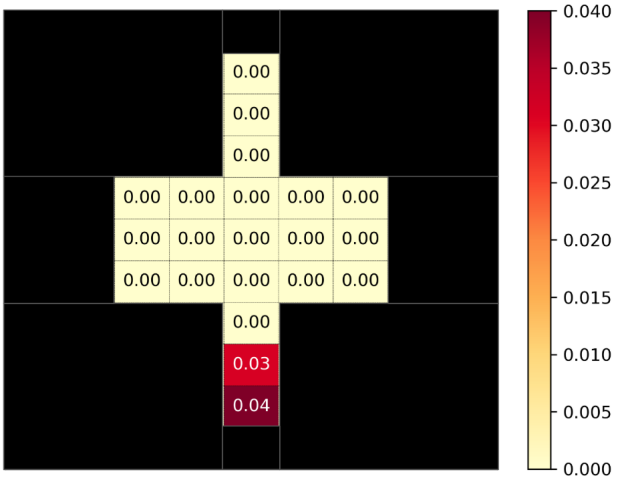


Figure A.10: Fear plot for version 1(ϵ-greedy fear policy with $\epsilon = 0.1$ and short-horizon)

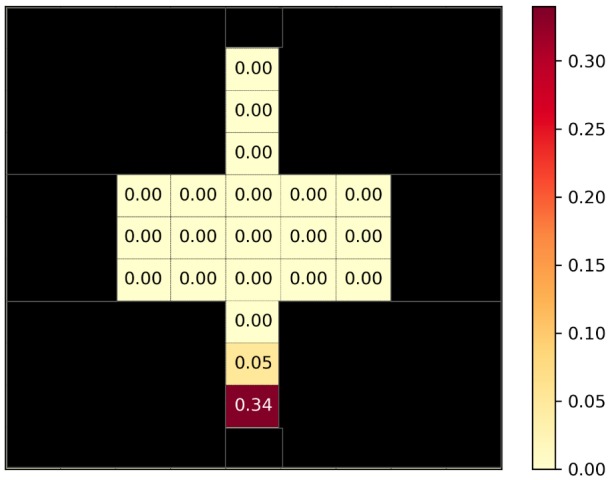


Figure A.11: Fear plot for version 2(softmax fear policy with $\tau = 5.0$ and short-horizon)

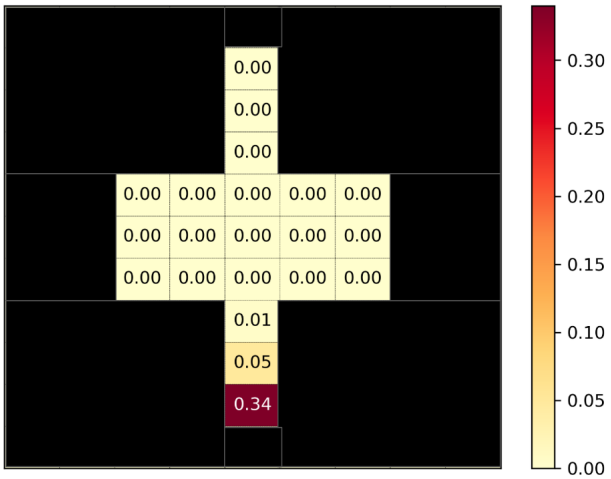


Figure A.12: Fear plot for version 3(ϵ-greedy fear policy with $\epsilon = 1$ and short-horizon)

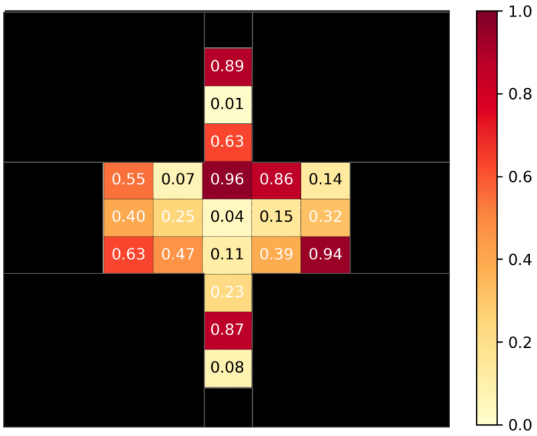


Figure A.13: Fear plot for version 4(baseline)

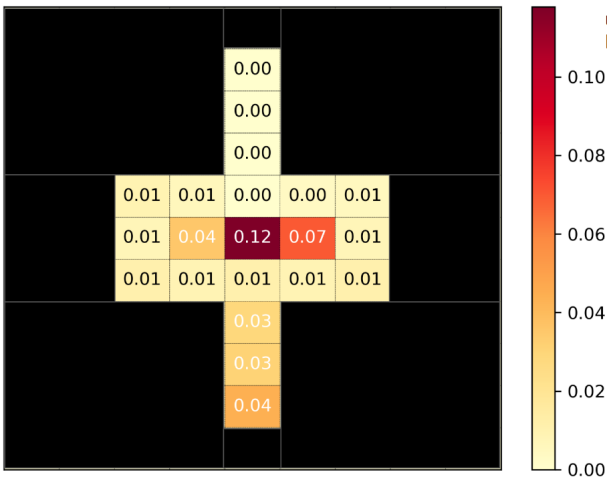


Figure A.14: Fear plot for version 5(ϵ-greedy fear policy with $\epsilon = 0.1$ and long-horizon)

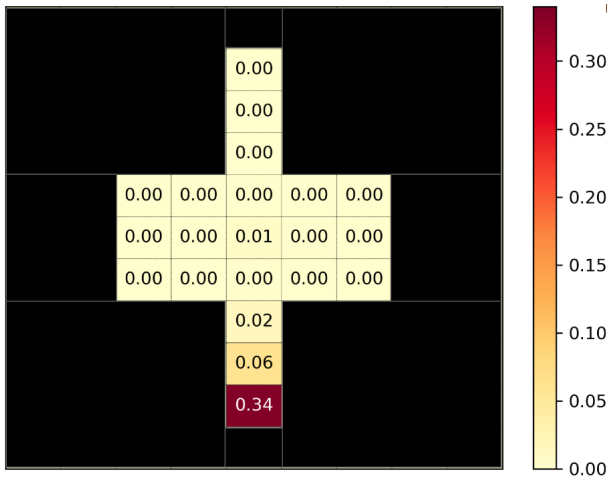


Figure A.15: Fear plot for version 6(softmax fear policy with $\tau = 5.0$ and long-horizon)

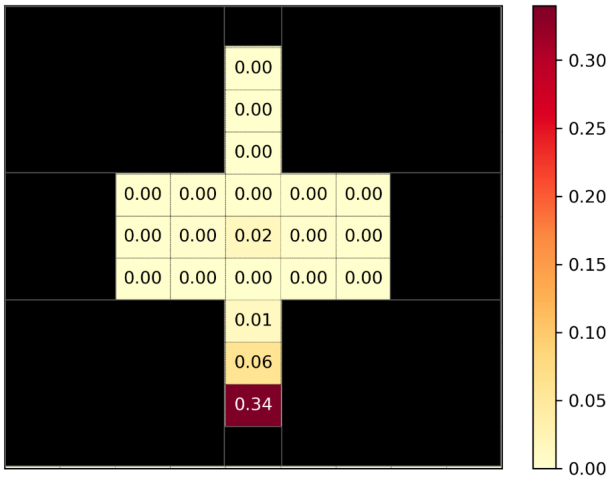


Figure A.16: Fear plot for version 7(ϵ-greedy fear policy with $\epsilon = 1$ and long-horizon)