Neurocomputational Simulations of Addiction using Reinforcement Learning

A dissertation submitted to The University of Manchester
for the degree of Master of Engineering
in the Faculty of Science and Engineering

2024

George Grainger

Department of Computer Science

Contents

Abstract viii

Declaration ix

Copyright x

Acknowledgements xi

1 Introduction 1

1.1 Motivation . . . 1
1.2 Aims and Objectives . . . 2
1.3 Evaluation strategy . . . 2
1.4 Report Structure . . . 3

2 Background 4

2.1 Reinforcement Learning (RL) . . . 4
2.1.1 Markov Decision Processes . . . 5
2.1.2 Temporal Difference Learning . . . 9
2.1.3 Q-Learning . . . 9
2.1.4 Model-Based and Model-Free Reinforcement Learning . . . 11
2.2 Reinforcment Learning in Psychology and Neuroscience . . . 12
2.2.1 Classical Conditioning . . . 12
2.2.2 Instrumental conditioning . . . 13
2.2.3 Dopamine as a Reward Signal . . . 14
2.2.4 Cognitive Maps, Habitual and Goal-Oriented Behaviour . . . 14
2.2.5 Neuroscience of addiction . . . 15

3 Design 16

3.1 Why Traditional RL Algorithms Fail to Capture Addiction . . . 16
3.2 Simulating Addiction Through Transient Dopamine Increase . . . 17
3.3 An Improved Approach to Modelling Addiction . . . 18
3.4 A Dual Process Model Capturing Cue Triggered Relapse . . . 20
3.4.1 State Classification . . . 22
3.4.2 Agent Value Evaluation . . . 24
3.4.3 Capabilities Relative to the Pre-Existing Designs . . . 25

4 Implementation 26

4.1 OpenAI Gym . . . 26
4.1.1 Core Environment Components . . . 26
4.1.2 Creating a custom environment . . . 26

4.1.3 Evaluation of OpenAI Gym 28
4.2 Creating agents 29

5 Experiments and Results **32**
5.1 Simulations 32
5.1.1 Acquisition of a Regular Response 33
5.1.2 Acquisition of a Pharmacologically Addictive Response 34
5.1.3 Cue Trigger Relapse 38
5.1.4 Behavioural addictions 39
5.1.5 Influence of Environmental and Genetic Factors 42
5.2 Evaluation of the design 43
5.2.1 Evaluation of Experiments Reproducing Pre-Existing Findings 43
5.2.2 Evaluation of Features Beyond the Pre-existing Designs 44
5.2.3 Fulfilment of Redish's (2008) Unified Framework for Addiction 44
5.2.4 Alternate Methods of Modelling Addiction 45

Bibliography **47**

Word Count: 13,809

List of Tables

2.1 Q-table of state-action values once q^* is converged upon 11
2.2 Pros and cons of model-free and model-based methods 12

5.1 Reward Dynamics in the Standard Testing Environment 32
5.2 Criteria for Addiction in Redish's (2008) unified model of addiction 45

List of Figures

2.1 Agent-environment interaction in a Markov decision process 5
2.2 A policy defines the probability of each action in a given state 6
2.3 Backup diagrams for the Bellman equations 7
2.4 Backup diagrams for the Bellman optimality equations 8
2.5 Q-learning illustration 9
2.6 Illustration of Q-learning converging upon q^* 10
2.7 Model free vs model based behaviours 11
2.8 Classical conditioning showing a natural stimulus becoming a conditioned stimulus . 12
2.9 Blocking is a failure to learn potential secondary conditional stimulus 13
2.10 Instrumental conditioning shows behaviour is contingent on the quality of its outcome 13

3.1 Influence of the pre-existing designs in this project's dual process model 16
3.2 The dual process model unifies state classification and value evaluation components . 21

4.1 Class Hierarchy for the custom OpenAI Gym environment 27
4.2 Class Hierarchy for the agent classes 29

5.1 Envrionment Dynamics for the Simulations. 32
5.2 Results Showing the Adoption of an Optimal Policy in Response to Natural Reward . 33
5.3 Results Showing the Adoption of a Suboptimal Policy in Response to Addictive Reward 34
5.4 Results Showing A Reduction In Perceived Reward After Prolonged Drug Taking . . 35
5.5 Environment for Testing Relationship Between Drug Taking and Impulsivity 35
5.6 Results for Relationship Between Drug Taking and Impulsivity 36
5.7 Testing Environment for Blocking Simulation 37
5.8 Results of Blocking Simulation 37
5.9 The dual process model shows relapse whereas pre-existing models cannot 39
5.10 In the dual process model the number of internal states increases during extinction . . 40
5.11 Dual process model can show relapse without lowering the rewards after extinction . 41
5.12 Environment dynamics for the gambling experiment 41
5.13 The duration of a winning streak increases the likelihood of gambling addiction . . . 42
5.14 In the pre-existing models, win streak duration did not influence gambling addiction . 42
5.15 Influence of Genetic and Environmental Factors In Uptake of Drug Addiction 43

List of Algorithms

1 Q-Learning (off-policy TD control) for estimating $\pi \approx \pi^*$ 10
2 Environment initialisation 27
3 Agent Training Loop 28
4 TD UPDATE implementation in the final design 29
5 GET ACTION implementation in the final design 30
6 State identification implementation 30
7 Agent Train Method 31
8 Agent Implementation For Design 31

Acronyms and Abbreviations

AI	Artificial Intelligence	4
MDP	Markov Decision Process	5
ML	Machine Learning	4
RL	Reinforcement Learning	4
SUD	Substance Use Disorder	1
TD	Temporal Difference	9
MB	Model-based	11
MF	Model-free	11
MI	Mutual Information	22
RPE	Reward Prediction Error	13
GUI	Graphical User Interface	26

Abstract

NEUROCOMPUTATIONAL SIMULATIONS OF ADDICTION USING REINFORCEMENT LEARNING

George Grainger

A dissertation submitted to The University of Manchester
for the degree of Master of Engineering, 2024

Addiction is a widespread and growing problem that results in the suffering of millions of people. This project aimed to create reinforcement learning simulations that could simulate the neural basis of addiction and reproduce characteristics such as relapse and compulsion that cannot be seen in standard reinforcement learning algorithms. It is hoped that through the results of this work, insight can be gained into risk factors and the effectiveness of treatments for addiction and improve the quality of life for those who have an addiction, as well as helping to reduce broader issues across society.

To achieve this, the project created a dual-process model combining a model-free value evaluation function from Dezfouli's (2009) research into cocaine addiction, along with state classification components based on Pettine's (2023) research into human latent state generalization. This enabled the dual-process model to replicate findings from previous studies of addiction, showing the compulsive choice of addictive suboptimal addictive behaviour as well as increased impulsivity and lower perceived reward after addictive behaviours.

The dual-process model also moved beyond existing model-free designs, accurately predicting addictions beyond those directly stimulating dopamine stimulation. This allowed it to capture behavioural addictions in the form of gambling. Moreover, the introduction of a state classification component enabled relapse to be captured, even in the case that natural rewards are increased during a period of extinction.

Finally, the dual-process model assesses the relative impact of genetic and environmental factors in the uptake of addiction, with predictions that align with experimental results showing that increased natural rewards can decrease the chances of uptake and maintenance of an addiction.

Declaration

No portion of the work referred to in this dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made **only** in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see `http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=24420`), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see `http://www.library.manchester.ac.uk/about/regulations/`) and in The University's policy on presentation of Theses

Acknowledgements

I am truly grateful to my family and friends for their consistent support throughout my life. Your encouragement and care have been invaluable. I would particularly like to express my thanks to my parents and brothers, who have always been there for me. Their love and guidance have significantly shaped who I am today, and I am grateful for all they've done for me.

Lastly, I want to thank my project supervisor, Oliver Rhodes, for his guidance over the past year. His expertise and mentorship have been essential to the project's success and I would not have known where to start without his direction.

1 Introduction

This project creates simulations using reinforcement learning techniques to highlight the neural basis of addiction, investigating the influence of model-free and model-based mechanisms in human learning[26]. It examines the role of dopamine in motivation and decision-making, predicting how addictive choices can co-opt the natural reward-seeking process, by direct and indirect stimulation of the dopamine system, to result in the adoption of potentially maladaptive and destructive actions[61, 78]

1.1 Motivation

Addiction is defined as a compulsive and persistent behaviour despite adverse social, occupational and health consequences[3, 35, 54]. Cessation of problem behaviours is difficult and characterised by high rates of relapse even when there are strong intentions otherwise[8, 73].

Addiction is, therefore, responsible for significant costs on both personal and societal levels. In families with a substance use disorder (SUD), a child is at a 3x risk of physical or sexual abuse and has a 50% increased chance of being arrested in adolescence[42]. It also shares comorbidities with physical and psychiatric disease[7]: one in eight of the 11.2 million injecting drug users were living with HIV¡ and nearly half had Hepatitis C in 2020[77, 85]. Globally, it has a large economic impact and not-insignificant contributions to pressing issues such as climate change[15, 17, 85]. In 2021, 246,800 ha were used for cocaine cultivation, often through illegal deforestation, which is equivalent to 345,600 football pitches[85]. This will become a growing issue with more young people using non-prescription drugs compared with previous generations: in 2020, 5.6% of the world population aged 15-64 had used a drug in the past year, 26% more than in 2010[85].

Compounding this are the behavioural addictions, such as pathological gambling, gaming and social media addiction, which are becoming increasingly problematic[34, 54, 63]. Even though they lack many of the pharmacological effects of SUDs they are characterised by neurobiological parallels and similar symptomatology[23, 29, 54]. This has led to dramatic changes in the field of addiction in recent years. Still, despite considerable neurocomputational investigations into understanding the basis of substance addictions, behavioural addictions remain relatively neglected[23, 34].

The treatment of addiction could improve the quality of life of afflicted individuals and their loved ones while helping reduce issues across wider society. Therefore, developing effective therapeutic interventions is a high-priority goal for neuroscience[15, 78]. However, with stigma and misconception toward addictive illness being commonplace, and those with addictions being vulnerable, it makes clinical research challenging[10]. Consequently, computational neuroscience could be particularly valuable, with reinforcement learning enabling the development of accurate testing environments and agents[78]. Harmful patterns and risk factors could be more easily identified and research made increasingly accessible. This could enable new developments of treatments while limiting the need for in-person testing, reducing both demand and harm for participants for in-vivo studies.

1.2 Aims and Objectives

The project aims to extend upon previous research to produce a novel reinforcement learning agent that captures both pharmacological and behavioural addictions. It hopes to predict some of the critical features of addiction that cannot be characterised by typical reinforcement learning algorithms, such as the interference between addictive substances and dopamine-mediated reward-prediction errors to result in sub-optimal actions being selected[78]. Furthermore, it's hoped to simulate the long-term neuroadaptations in the biological circuitry that result in relapse[15, 74], by moving beyond a purely "habit-based" model-free agent, to a dual-learning system that can predict more complex "goal-oriented" choices[26, 43]. Pursuing this will involve both research and design objectives:

Research objectives:

1. Build a strong understanding of reinforcement learning
2. Learn the basic psychological theory of decision-making and addiction
3. Review state-of-the-art research to enable the creation of a new design that is an evolution of previous models

Design objectives:

1. Develop an agent that compulsively and persistently chooses sub-optimal addictive behaviours
2. Enable cue based relapse through combining model-free and model-based approaches
3. Capture behavioural addictions beyond those that directly stimulate the dopamine system
4. Explore genetic and environmental factors, verifying the predictions against experimental results

1.3 Evaluation strategy

Evaluation of the model will be achieved through a verification framework consisting of comparisons with existing psychological understanding and results from previous studies[14, 61, 63].

Initially, testing will occur on foundational elements of the model, such as the acquisition of beneficial behaviours. This will ensure additional complexity builds upon valid assumptions and notable findings are truly outcomes of the implementation rather than extraneous factors.

Extending this, more involved tests based on predictions of psychological effects such as blocking[14, 78] and environmental change[54] will take place. Following this, attention will be paid to the outcomes of these results and the ten key vulnerabilities outlined in Redish's 2008 unified framework for addiction[62]. This presents a range of model-free and model-based features that indicate a person's susceptibility to addiction, the transition to addiction, and relapse. This will help justify whether the differences between the predictions from previous models and this project's model are a limitation or strength of the design.

Finally, the model will be considered relative to its achievements and complexity, evaluating its success based on whether additional features are computationally worthwhile.

1.4 Report Structure

Chapter 1 summarises why this research is important. It provides the aims and goals the project hopes to achieve, alongside an overview of how the project will be evaluated.

Chapter 2 outlines background material, placing this project in the context of previous research. It details both computational and psychological reinforcement learning understanding, the dopamine system and how they each relate to addiction.

Chapter 3 is dedicated to the design of the model. The project presents multiple designs of increasing complexity, highlighting their respective benefits and limitations. Through this, it's hoped that different aspects of the project's final design are more understandable. This content will build on the background, with only reinforcement learning and psychological concepts specific to each design being explained as they're introduced.

Chapter 4 details the implementation of these designs, providing insight into the agents and environments used in the project. It discusses the use of OpenAI gym, and how object-oriented design makes the project more extensible for future work.

Chapter 5 defines the experiments for verifying the model against previous models and established psychological theory. This chapter tests factors ranging from genetic susceptibility to cue-triggered relapse, highlighting the advantages of this project's model over the others in the design chapter.

Chapter 6 concludes the report with a personal reflection on the project's successes and failures alongside suggestions for future work.

2 Background

2.1 Reinforcement Learning (RL)

Reinforcement learning (RL) is a concept that bridges psychology and computer science, originating from studies of animal learning[30]. Experiments by B.F. Skinner on Operant Conditioning showed that animals, including humans, associate behaviours with their positive or negative consequences, repeating behaviours resulting in rewarding outcomes and avoiding those penalised[75, 76]. This built upon Thorndike's earlier work on the Law of Effect, proposing that behaviours followed by satisfying outcomes will be more likely to be replicated, while those that aren't will be reduced[79, 80, 81]. This laid the foundations for some of the earliest work in artificial intelligence (AI) and led to the revival of RL as a branch of machine learning (ML) by Watkins in the 1980s[78, 88].

Principally, computational RL implements many of these psychological concepts, with an emphasis on a decision-making agent learning the actions that maximise a cumulative reward function, through direct interaction with an environment[78]. In contrast to supervised learning methods, the agent learns to evaluate actions based on training information, rather than being instructed by labelled examples. RL can work in the presence of uncertainty and incomplete information; using trial and error, agents discover which actions result in the best reward to be given as feedback from the environment. Notably, these actions may also alter the agent's next state and influence subsequent rewards[78]. As a result, the agent must learn the policy that not only chooses the best short-term option, but one that navigates the environment to maximise the overall reward, accounting for delayed gratification.
Key elements of a RL agent are:

1. A *policy* defining an agent's behaviour at a given time. It's a mapping from states to the probability of selecting each possible action and is psychologically analogous to stimulus-response rules[24].

2. A *reward signal* defining the goal of the RL problem. At each time step a numeric *reward* is sent from the environment and used to evaluate good and bad events. It's immediate, analogous to pleasure and pain in humans, and has the primary role of changing the policy.

3. A *value function* defining what's optimal long term. It's an expectation of future rewards based on the current state, accounting for the desirability of states that could be subsequently moved into.

4. An optional *model* mimicking the environment. This provides inferences on what the environment may do next, enabling the agent to plan.

Reinforcement learning is an effective and increasingly popular computational approach to understanding and automating goal-oriented learning and decision-making[78]. It's applied in a range of areas, from robotics[38] and intelligent game-playing systems[50] to neuroscience. This project aims to demonstrate the latter of these through its application of RL to modelling addiction.

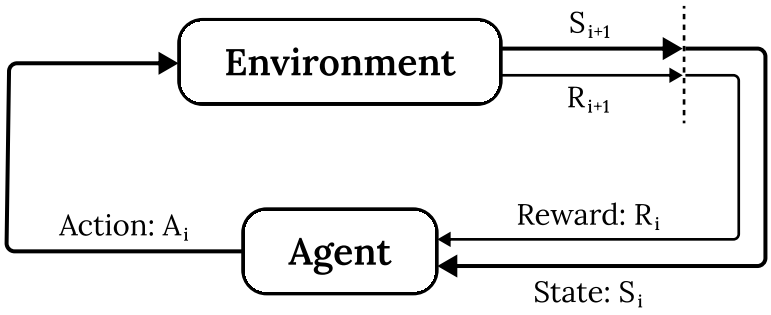

Figure 2.1: Agent-environment interaction in a Markov decision process

2.1.1 Markov Decision Processes

Markov decision processes (MDPs) are mathematical formalisations of sequential decision-making. They act as idealised forms of RL, where on each discrete time step t, an agent in state S_t chooses an action A_t based on its current policy, in turn producing a reward R_{t+1} and new state S_{t+1} (fig. 2.1). Formally a MDP can be defined as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$[78, 86] where:

- $\mathcal{S}$ is the set of all possible states
- $\mathcal{A}$ is the set of all possible actions
- $\mathcal{P}^{ar}_{ss'} \doteq \mathbb{P}\left[S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a\right]$ is the function describing the system's dynamics. It is the joint probability of a reward r and next state s', given action a is taken in state s.
- γ is the discount factor trading off later rewards for earlier ones, $\gamma \in [0, 1]$

The Markov property

The Markov property is commonly stated as "given the present, the future is independent of the past[5]." It's the concept that future states depend entirely on the current state since this sufficiently captures all the relevant historical information of the system[78, 86]. Formally, this is given by:

$$\mathbb{P}\left(S_{t+1} = s_{t+1} \mid S_t = s_t\right) = \mathbb{P}\left(S_{t+1} = s_{t+1} \mid S_t = s_t, S_{t-1} = s_{t-1}, \ \ldots \ , S_0 = s_0\right) \tag{2.1}$$

Despite not being viable in all situations, it's a simplifying assumption that makes modelling complex real-world processes manageable and is commonly used in neurocomputational simulations[78].

Calculating expected future reward

Formalising an agent's objective can be done by looking at its typical trajectory:

$$S_0,\ A_0,\ R_1,\ S_1,\ A_1,\ \ldots\ ,\ S_t,\ A_t\ R_{t+1},\ \ldots\ ,\ R_T,\ S_T \tag{2.2}$$

By eq. (2.2), it's shown at time step t, an agent's return from future rewards will be R_{t+1} , R_{t+2}, R_{t+3}, ..., R_T. Therefore, a simple measure of the expected return at a given timestep G_t can be defined as:

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \ldots + R_T \tag{2.3}$$

However, the definition in eq. (2.3) isn't always optimal. It's mathematically convenient to discount and prioritise immediate reward to a degree given by γ. This overcomes issues for reinforcement learning in the case where $T = \infty$ and better reflects human behaviour, which can prefer short to long-term rewards of greater magintude[25, 78, 86]. An improved definition for G_t can be given by:

$$G_t \doteq R_{t+1} + \gamma\, R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} = R_{t+1} + \gamma\, G_{t+1} \tag{2.4}$$

The agent will have a policy π mapping the probabilities of taking an action given a particular state:

$$\pi(a \mid s) = \mathbb{P}[A_t = a \mid S_t = s] \qquad \forall s \in \mathcal{S},\ \forall a \in \mathcal{A} \tag{2.5}$$

An example policy is illustrated in fig. 2.2, where it's shown in state s_0 all actions are equally likely to be chosen while in s_4 action a_0 has double the probability of actions a_1 and a_2.

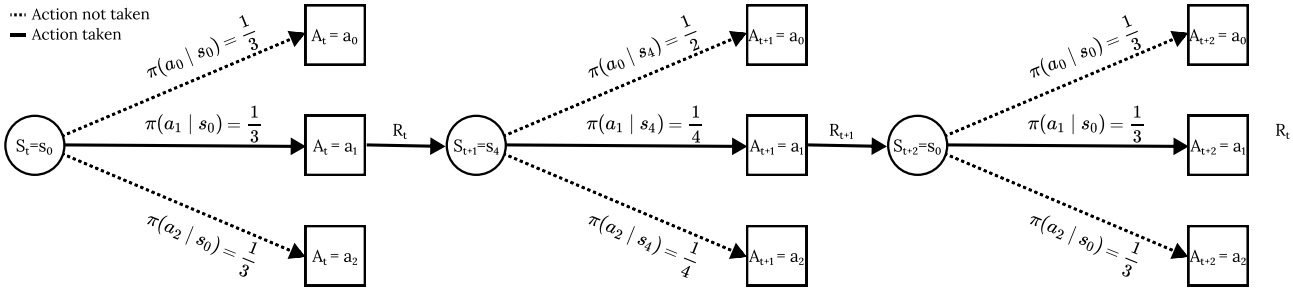

Figure 2.2: A policy defines the probability of each action in a given state

When in state s and under policy π, the expected return can be defined by the state-value function:

$$\begin{aligned} v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] && (2.6)\\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] && \text{(by eq. (2.4))}\\ &= \sum_a \pi(a \mid s) \sum_r \sum_{s'} \mathcal{P}^{ar}_{ss'} \left[r + \gamma \mathbb{E}_\pi[G_{t+1} \mid S_{t+1} = s'] \right] \\ &= \sum_a \pi(a \mid s) \sum_r \sum_{s'} \mathcal{P}^{ar}_{ss'} \left[r + \gamma v_\pi(s') \right] \qquad \forall s \in \mathcal{S} && (2.7) \end{aligned}$$

The Bellman equations

The Bellman equation for v_π is the result given in eq. (2.7); it's a fundamental equation underpinning many RL algorithms. It recursively expresses the relationship between the value of a state and the values of its successor states[78, 86]. As shown on the left of fig. 2.3, this is done by iterating over every possible action (red) and finding the expected reward for each state the action could lead to, along with the reward gained by moving to that state (blue). An average is calculated, weighted by an action's probability of being taken under the policy π, combined with every next-state reward pair's chance of occurring in the environment, $\mathcal{P}^{ar}_{ss'}$[5]. Overall, this gives a state's value as the discounted expected reward from the next state summed with the expected reward moving into the next state[78].

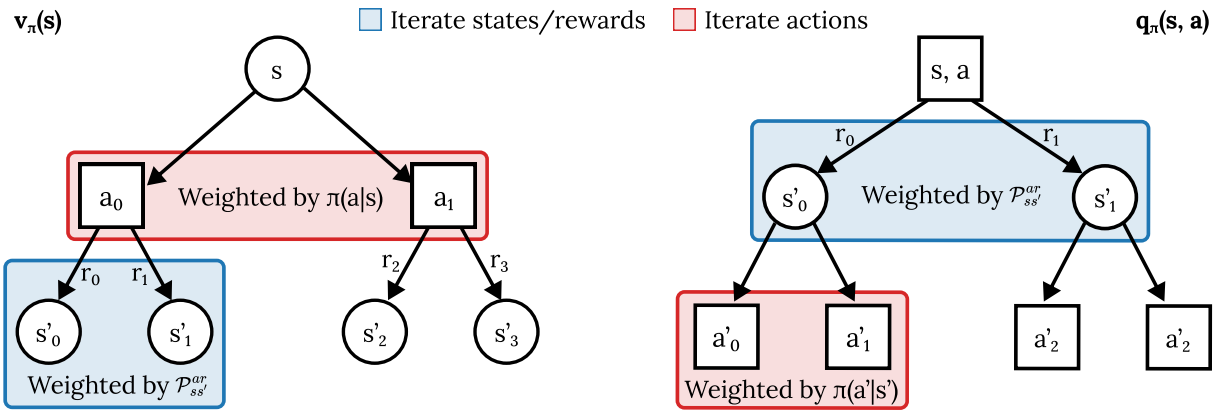

Figure 2.3: Backup diagrams showing how values are iterated over in the Bellman equation

It can also be convenient to know the expected value starting in state s, taking action a, and following the policy π thereafter[78]. This is the action-value function and can be defined by:

$$q_\pi(s,a) \doteq \mathbb{E}\left[G_t \mid S_t = s, A_t = a\right] \tag{2.8}$$

$$= \mathbb{E}\left[r + \gamma v_\pi(s') \mid S_t = s, A_t = a\right] \qquad \forall s \in \mathcal{S},\ \forall a \in \mathcal{A} \tag{2.9}$$

Since the agent's policy π defines the probabilities of each action in a given state (eq. (2.5)), it's possible to give the state-value function (eq. (2.6)) in terms of the action-value function (eq. (2.8)), by taking a weighted sum of the return across all the actions relative to their probability:

$$v_\pi(s) = \sum_a \pi\left(a \mid s\right) q_\pi(s,a) \tag{2.10}$$

Substituting eq. (2.10) into eq. (2.9) derives the Bellman equation for the action-value function:

$$q_\pi(s,a) = \sum_r \sum_{s'} \mathcal{P}^{ar}_{ss'} \left[r + \gamma \sum_{a'} \pi(a' \mid s') q_\pi(s', a') \right] \tag{2.11}$$

This version of the Bellman equation is useful when an agent needs to predict the expected return after taking a specific action in a given state[78] (fig. 2.3, right). However, the Bellman equations given in eq. (2.7) and eq. (2.11) are only the relationship under a given policy π, which if sub-optimal means there's no guarantee the obtained values will be highly rewarding[5]. For this reason, RL algorithms generally aim to estimate the optimal policy and value function[86].

Optimal policies and values

An optimal policy π^* can be defined through a partial ordering based on eq. (2.6). A policy π is said to be better than another π' if the expected return from π is at least as good as π' in all states[78]. Mathematically, this is given by:

$$\pi \geq \pi' \iff v_\pi(s) \geq v_{\pi'}(s), \qquad \forall s \in \mathcal{S} \tag{2.12}$$

There can be multiple optimal policies π^*, but this partial ordering (eq. (2.12)) makes it necessary that for each, the expected reward is the same in all states. Optimal state-value $v^*(s)$ and optimal action-value $q^*(s,a)$ functions are therefore defined such that v_{π^*} is v^* and q_{π^*} is q^*:

$$v^*(s) \doteq \max_\pi v_\pi(s) \tag{2.13}$$

$$q^*(s,a) \doteq \max_\pi q_\pi(s,a) \tag{2.14}$$

The Bellman optimality equations

The Bellman optimality equations are then simply the Bellman equations but under an optimal policy π^*. Intuitively, to achieve the maximum value in any state, the optimal policy must choose the action that provides the maximum expected reward[78] (fig. 2.4), and are therefore calculated by:

$$v^*(s) = \max_a \sum_r \sum_{s'} \mathcal{P}^{ar}_{ss'} \left[r + \gamma v^*(s') \right] \tag{2.15}$$

$$q^*(s,a) = \sum_r \sum_{s'} \mathcal{P}^{ar}_{ss'} \left[r + \gamma \max_{a'} q^*(s',a') \right] \tag{2.16}$$

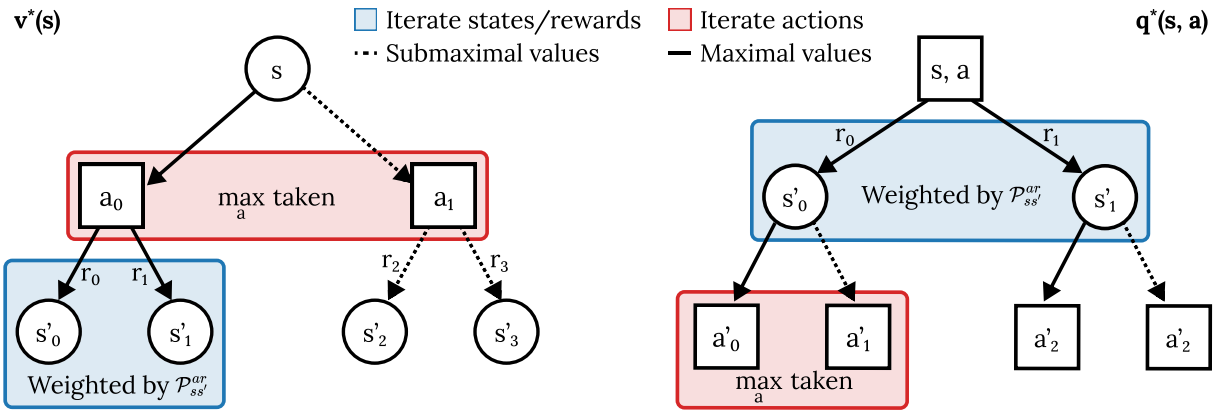

Figure 2.4: Backup diagrams show the Bellman optimality equations select the most rewarding action

Given v^*, the optimal policy π^* can be found by assigning non-zero probability only to actions that maximise the Bellman optimality equation. Accordingly, a greedy one-step search is optimal, as v^* already accounts for the future behaviour. The best short-term action will also maximise long-term reward[78].

Simpler still, choosing any action maximising q^* in each state is an optimal policy π^*. This avoids a one-step search, with the action values effectively caching its results. At the expense of memory of storing all the state-action combinations, q^* allows choosing the optimal action with no knowledge of the following states: i.e. without knowing anything about the environment's dynamics[78, 86].

However, finding v^* or q^* is challenging and sometimes infeasible even with an accurate model of the environment's dynamics. For a game such as chess, there is an estimated 10^{43} possible moves[72], so the enormous computational cost required and memory constraints prevent a solution from being

found. Consequently, RL methods often iteratively apply the Bellman equation to approximate q^* and π^*[88].

2.1.2 Temporal Difference Learning

Temporal difference (TD) learning is an idea that is central and novel to reinforcement learning (RL) that combines ideas from both Monte Carlo[70] methods and dynamic programming[5]. It's an iterative method that learns directly from experience and uses bootstrapping to update estimates of current states based on the estimates of subsequent states, without needing to wait for a final evaluation[78].

2.1.3 Q-Learning

Q-Learning was an early breakthrough in TD learning[78, 88], producing an action-value function Q that directly approximates q^* independent of the policy being followed through the update function:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \Big[\overbrace{R_{t+1} + \gamma \max_a Q(S_{t+1}, a)}^{\text{TD taget}} - Q(S_t, A_t) \Big]_{\text{TD error, } \delta_t} \tag{2.17}$$

A learning rate, α, is used to weigh the influence of the recent TD error relative to those found previously. When α is small or stochastically decreasing so $\sum_t \alpha_t = \infty, \sum_t \alpha_t^2 < \infty$ holds, Q-Learning is proven to converge to q^*[88]. An optimal policy π^* can then be found as shown in section 2.1.1.

The algorithm is given in algorithm 1 and graphically illustrated in fig. 2.5. For a given state, an action is chosen, generating a TD target from the reward received and the expected reward by picking the best action in the next state based on the current estimates. This is fed back to the previous state which updates through eq. (2.17). The algorithm then repeats, choosing an action in the next state, however, since Q-learning is *off-policy* this isn't required to be the perceived maximal action[78, 88].

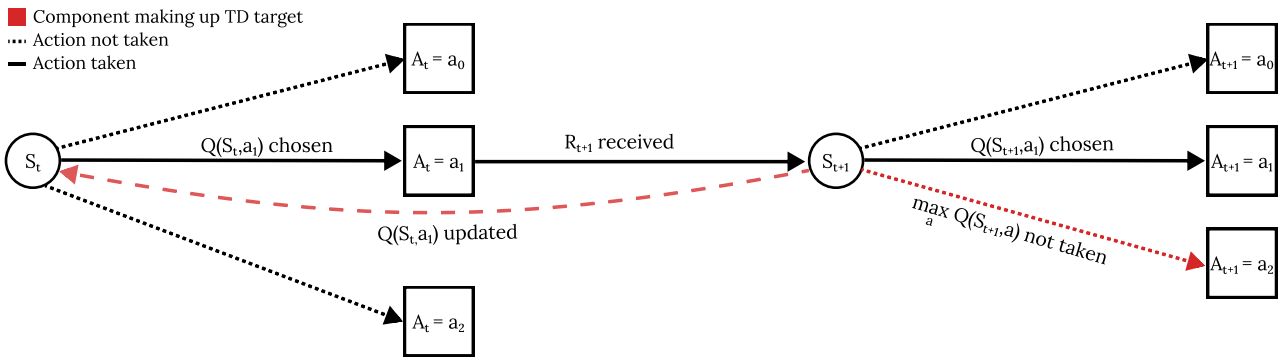

Figure 2.5: Q-learning passes back TD target, enabling state update to occur progression to next state

Algorithm 1 Q-Learning (off-policy TD control) for estimating $\pi \approx \pi^*$ [78]

Initialise $Q(S,A)$ for all $S \in \mathcal{S}, A \in \mathcal{A}$
for each episode **do**
    Initialise S
    for $t \leftarrow 1$ **to** max timesteps **do**
        Choose A from S using a policy derived from Q
        Take action A, observe R, S'
        $Q(S,A) \leftarrow Q(S,A) + \alpha\,[R + \gamma \max_a Q(S',a) - Q(S,A)]$
        $S \leftarrow S'$

Exploration vs exploitation

Despite the idea of choosing sub-optimal actions seeming to contradict the concept of an optimal policy, it's actually important to finding π^*[88]. Environments can be stochastic, and if an agent picks an optimal action followed by a worse action, there's a probability that the optimal action will provide less reward on that occasion. If the agent then exploited this, subsequently choosing only the action it's estimated to be best, it would never discover that the first action is optimal. It's necessary to explore actions so repeated rewards are given, enabling state-action pairs to converge to their statistical means. This leads to generally more accurately informed choices, increasing reward overall[78, 86].

Q-tables

A Q-table of size $\mathcal{S} \times \mathcal{A}$ is used to store the value associated with each state-action pair in the Q-learning algorithm (table 2.1). The appropriate values can be looked up from this table on each timestep and used in eq. (2.17). Through repeated iterations, the Q-table's values will converge on q^* and by picking the maximal column (action) for each row (state) will provide an optimal policy[78, 88]. This is illustrated by the Q-table (table 2.1) for fig. 2.6, which after 100,000 iterations shows that actions representing the optimal policy have maximum value. Furthermore, those that lead to falling into the lakes have been assigned a value of -1, indicating they're bad actions for those states.

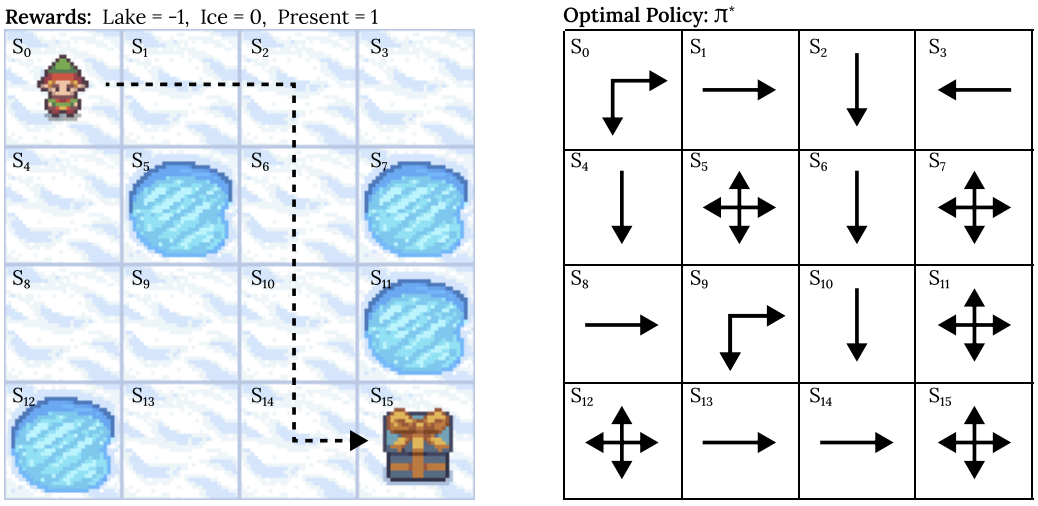

Figure 2.6: Illustration of Q-learning converging upon q^* in the frozen-lake envrionment[1]

[1] Image found along with example at: `https://gymnasium.farama.org/environments/toy_text/frozen_lake/`

Q-Table		Actions			
		Left	Down	Right	Up
States	0	0.9415	0.951	0.951	0.9415
	1	0.9415	-1.0	0.9606	0.951
	⋮	⋮	⋮	⋮	⋮
	13	-1.0	0.9801	0.99	0.9703
	14	0.9801	0.99	1.0	0.9801
	15	0.0	0.0	0.0	0.0

Table 2.1: Q-table of state-action values once q^* is converged upon in the frozen-lake envrionment[1]

2.1.4 Model-Based and Model-Free Reinforcement Learning

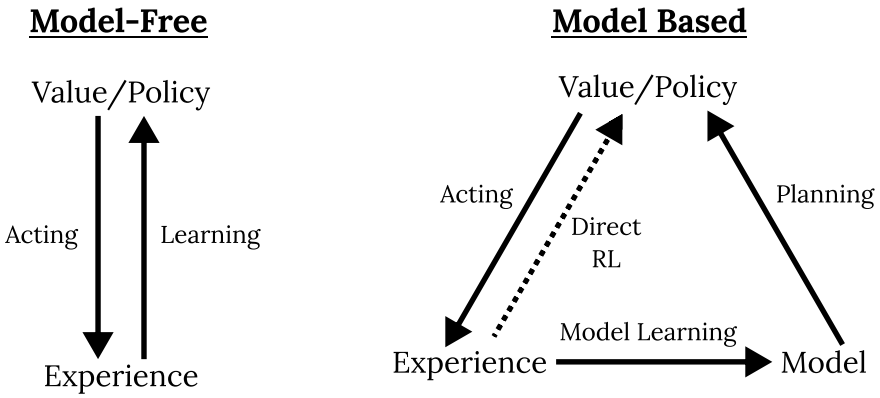

Figure 2.7: Model free vs model based behaviours

TD learning methods are *model-free (MF)*, meaning they can learn directly from an experience without a model of the environment[78]. However, there are alternate *model-based (MB)* approaches where the agent has an internal representation predicting how the environment will transition in response to their actions. This enables planning based on the anticipated next state and the reward from different choices[78] (fig. 2.7).

Real experience in planning agents can be used to improve the Model: *model learning*. *Direct RL* and *indirect RL* are methods where the value function is updated either directly, or indirectly through the Model. Both indirect and direct methods have advantages and disadvantages. Indirect methods generally achieve better policies using fewer environmental interactions, while direct methods are typically what is seen in humans. Therefore, these especially apply to this project's goal of modelling addiction[78, 86].

In model-based learning, the model's quality is paramount. However, building an accurate model of real-world scenarios is often challenging. Stochastic environments and limited samples being observed can lead to a model being an imperfect approximation. When this is the case, the planning process will have misinformed value estimates and will likely compute a suboptimal policy[78].

Advantages and disadvantages exist when exclusively using MF and MB methods (table 2.2). *Hybrid models* overcome this by integrating elements from both to give a complementary solution. This combined approach mitigates many drawbacks but at the expense of increased design complexity[74].

	Model-free	Model-based
Advantages	• Computationally efficient • Robust to model inaccuracies	• More sample efficient • Better generalisation to new tasks • Better performance when dynamics change
Disadvantages	• Poorer sample efficiency • Poorer generalisation • Limited planning/reasoning	• Requires accurate model • Can be more computationally expensive • Less practical for high-dimensional state/action spaces

Table 2.2: Pros and cons of model-free and model-based methods

2.2 Reinforcment Learning in Psychology and Neuroscience

Since reinforcement learning (RL) is inspired by animal learning, unsurprisingly there are considerable parallels between the computational understanding and the neural basis of reward-related learning. Particularly, ideas of optimising return over an extended period is a key feature in human learning[49, 71].

2.2.1 Classical Conditioning

Pavlov first identified the ability to associate new stimuli with innate reflexes, now known as classical conditioning[56]. Figure 2.8 shows how an unconditional response to an unconditioned stimulus, such as a dog salivating when seeing food, can be associated with a neutral stimulus, such as the tone from a whistle, through repeated paired exposure. This leads to the whistle becoming a conditioned stimulus, which alone produces a conditional response of the dog salivating[56]. These associations are acquired in human learning, enabling anticipation of upcoming events and how best to react[24].

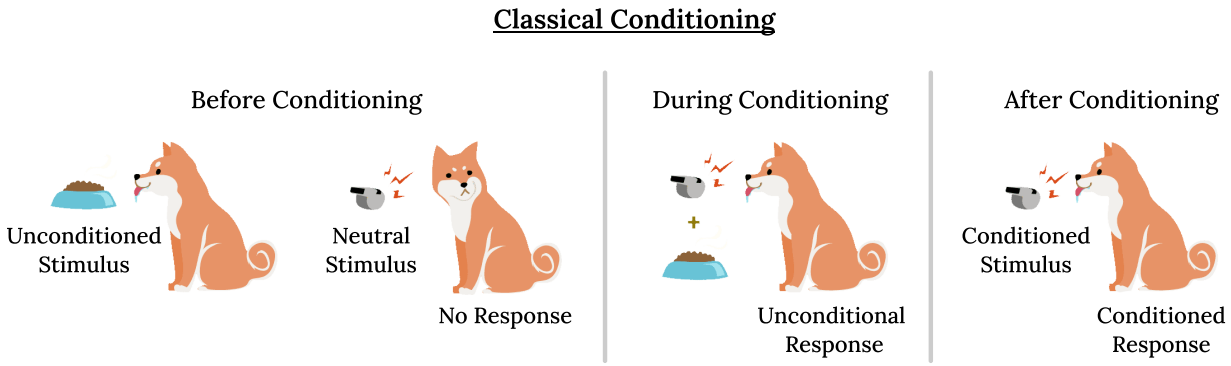

Figure 2.8: Classical conditioning showing a natural stimulus becoming a conditioned stimulus[1]

Blocking

Blocking (fig. 2.9) is the phenomenon in classical conditioning when a previously learned association between a conditional stimulus and response prevents an animal from learning a secondary conditional response to an unconditional stimulus when all three are presented together[33, 46, 65].

[1]Image adapted from: https://www.verywellmind.com/classical-conditioning-2794859

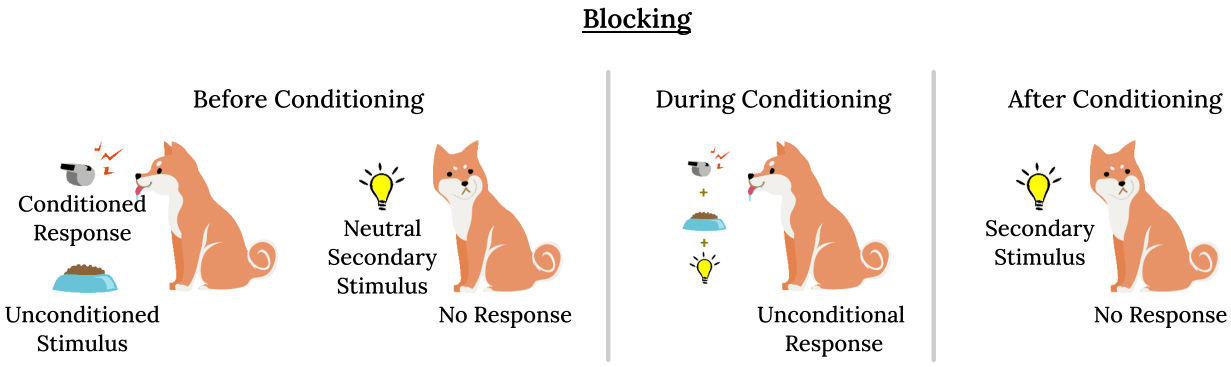

Figure 2.9: Blocking is a failure to learn potential secondary conditional stimulus[1]

Spontaneous recovery

Spontaneous recovery is one aspect of classical conditioning that is particularly relevant to addiction. It is the sudden reemergence of a learned response after a period of extinction when the previously conditioned and unconditioned stimuli are once again presented together[66]. This can explain *cue-triggered relapse* where re-exposure to environmental cues previously associated with addictive behaviour causes the re-uptake of problem behaviours after a period of abstinence[19, 53].

2.2.2 Instrumental conditioning

As summarized at the start of section 2.1, instrumental/operant conditioning is the concept that behaviours are learned contingent on the reward received[76, 78]. This was first shown by early experiments by B.F. Skinner and Edward Thorndike[75, 80], which tested animal response frequency based on reward and punishment (fig. 2.10). Essential aspects of RL algorithms are related to instrumental conditioning, notably, how they're *selectional* and *associative*, exploring different options and then choosing those that produce the best outcome associated with the agent's situation[78].

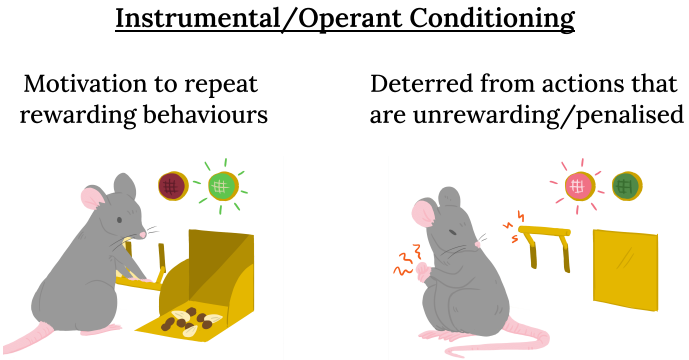

Figure 2.10: Instrumental conditioning shows behaviour is contingent on the quality of its outcome[1]

There's a close relation between Thorndike's Law of Effect[79, 80, 81] and the TD update given by equation (2.17), despite TD learning aligning closer to modern learning theories suggesting that reward prediction errors (RPEs) are more fundamental to behavioural change[65, 82]. Humans use instrumental conditioning across society (e.g. in schools), with reward/punishment structures being

[1]Image adapted from: `https://www.verywellmind.com/operant-conditioning-a2-2794863`

commonplace. Therefore, humans must have a biological *reward signal* analogous to those used in RL, which we do in the form of dopamine.

2.2.3 Dopamine as a Reward Signal

Dopamine is a neurotransmitter playing a critical role in the brain's reward system, signalling TD errors in brain regions such as the striatum, prefrontal cortex, and cerebral cortex[51]. Phasic dopamine is therefore analogous to δ_t in TD learning (eq. (2.17)). Despite these complex and interconnected brain structures, this relationship suggests that a reductionist understanding provided by RL simulations could be useful in reconciling some features of addiction[78].

Reward prediction error

As stated above, early classical theories proposed that reward-directed learning depends on the temporal contiguity between stimuli and reward[56, 81]. However, most modern learning theories contrast this, stating that learning is influenced by prediction errors[46, 57, 65, 82]. The *reward prediction error (RPE) hypothesis of dopamine neuron activity* proposes that phasic dopamine release delivers an error signal between an old and new estimate of expected reward[51, 78]. This RPE drives decision-making towards actions resulting in better outcomes. Findings from neuroimaging studies to animal models have provided experimental support for this being a key mechanism in animals, including humans' ability to adapt to new environments[16, 82]. However, substance use disorders (SUDs) could involve drugs that directly or indirectly stimulate dopamine and lead to incorrect RPEs. As a result, the behaviour could be treated favourably, despite potentially being suboptimal[4, 61, 87].

2.2.4 Cognitive Maps, Habitual and Goal-Oriented Behaviour

Model-based RL methods have elements in common with human cognitive maps[83]. These are mental representations created by interacting environments, enabling people to plan and execute actions[78]. Similar to in MB learning methods, research in humans suggests that cognitive maps are used for decision-making, and updated and refined through new experiences and learning[37] (fig. 2.7).

The distinction between MF and MB based methods further relates to the distinction between habitual and goal-oriented behaviours in humans[43, 63]. Habits are behaviours that are triggered and performed mostly automatically, responding quickly to input from an accustomed environment and making it challenging to adapt the behaviour in response to change[78]. This is reflected by algorithms such as Q-learning, which can quickly learn the correct responses for a given environment, but then struggle to generalise this to new environments (table 2.2). Conversely, goal-oriented behaviour relies on aspects related more to MB behaviour, like planning. It's more adaptable but relies on understanding how the world is likely to respond to your actions[13].

Addiction is made up of a complex interaction MF and MB behaviours[26], with neither being able to fully capture all aspects alone. Furthermore, it's thought that addiction can alter the balance between MF and MB control, resulting in impulsivity and increased risk-taking. This increases the challenge

in modelling addiction and means simplifications must be made[44, 62, 36].

To overcome some of these challenges, *dual-process models*, a more general form of hybrid Model, attempt to integrate these two distinct cognitive processes underlying decision-making. Such a combined approach should more realistically model the intuitive habitual system, its link to dopamine, and the slower more deliberative goal-driven system that often is linked to seeking addictive behaviours[74].

2.2.5 Neuroscience of addiction

Addiction involves both genetic and environmental factors and results in changes to the brain's reward and motivation systems. Addictive actions can directly or indirectly release dopamine, such as in the case of SUD, producing a pleasurable experience in the short term[61]. However, with repeated exposure, the reward system becomes deregulated and the behaviour can become compulsive[32, 40].

The pre-frontal cortex, which is responsible for decision-making, impulse control and judgement, undergoes structural changes in response to addiction. This makes it harder for those with addiction to resist cravings and increases the likelihood of relapse[22, 41].

Finally, addiction can alter the brain's limbic system, which regulates emotions and memory. This can make it easier for individuals to form positive associations with addiction, further reinforcing it even when it's ultimately becoming destructive[45].

Overall, a holistic RL simulation should try to account for each of these factors and account for the interplay between the model-free and model-based components of human decision-making. This will make the results more applicable to real-world understanding of risk factors and treatments, benefitting those with both behavioural and pharmacological addictions.

3 Design

This chapter aims to summarise the design of the neurocomputational simulations created in this project. It does this by first presenting two pre-existing designs that were extended upon, highlighting their respective benefits and limitations in the hope that this makes the project's dual process design more understandable. The dual process design will integrate an action evaluation function from the pre-existing models, along with a state classification mechanism from Pettine's (2023) study on human latent-state generalisation[59]. This will produce a more holistic simulation capable of reproducing a more comprehensive range of addiction characteristics. It will indicate how the introduction of a model-based (MB) component can overcome issues associated with purely model-free (MF) designs and enable both behavioural and pharmacological addictions to be predicted, as well as additional features of addiction such as cue-triggered relapse.

3.1 Why Traditional RL Algorithms Fail to Capture Addiction

Addiction is a multifaceted problem consisting of complex behaviours linked to factors such as compulsion, relapse and irrationality. It's characterised by maladapted behaviour, even in the face of negative consequences such as physical or social harm[3]. As a result, traditional reinforcement learning (RL) algorithms that always choose the optimal reward aren't inherently capable of capturing addiction, as many of its features are beyond the ability of simple reward-based learning[63].

However, since addictive behaviours are hypothesised to access the same neural circuitry as natural reward[49, 78], it is possible to adapt temporal difference (TD) RL methods to simulate the reward prediction errors (RPEs) produced by dopamine and predict the adoption and maintenance of addiction[14, 61, 78]. Furthermore, by encompassing both MF and MB aspects of human behaviour, the dual process model can capture the complex relationship between cues, actions and expected reward[48, 74, 63]. Additionally, this enhancement can explain compulsivity in behavioural addictions that don't share the pharmacologically induced dopamine increase seen in substance use disorders (SUDs)[12, 34, 35, 60]. This facilitates its use across a broader range of applications, such

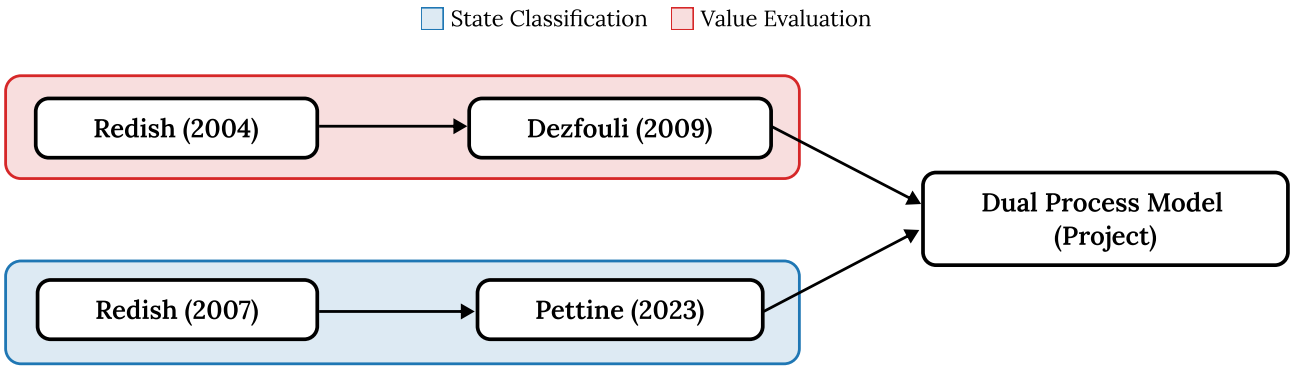

Figure 3.1: Influence of the pre-existing designs in this project's dual process model

as modelling gambling and social media addiction.

3.2 Simulating Addiction Through Transient Dopamine Increase

Redish (2004) built upon the first of these ideas, noting that addictive drugs produce transient increases in dopamine[67]. He, therefore, created a TD RL agent with an additional non-compensable drug-induced dopamine increase encoded into the value function. As a result, this research constructs an agent that inappropriately chooses addictive stimuli over potentially more rewarding actions[61].

Redish used an alternate form of the TD error that also discounts the reward received:

$$\delta_t = \gamma \left[R_{t+1} + V(S_{t+1}) \right] - V(S_t) \tag{3.1}$$

By incorporating a non-compensable reward for drug receipt and modifying the Q-learning algorithm in eq. (2.17) to align with the discounting changes from eq. (3.1), an updated version of the TD error can be obtained accounting for the dopamine surge:

$$\delta_t^c = \max \left(\gamma \left[R_{t+1} + \max_a Q(S_{t+1}, a) \right] - Q(S_t, A_t) + D(S_t),\ D(S_t) \right) \tag{3.2}$$

The Q-learning update can remain unchanged other than including δ_t^c:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \delta_t^c \tag{3.3}$$

The $D(S_t)$ term models the phasic dopamine activity induced by the pharmacological effects of drug receipt. It's notable that when $D(S_t) = 0$, Redish's δ_t^c from eq. (3.2) reduces to δ_t from eq. (3.1). Since natural rewards don't provide dopamine surge, $D(S_t) = 0$, in the adapted model these actions continue to converge upon the value defined by their expected reward[61, 78]. However, in states representing drug administration $D(S_t) > 0$ and the modified δ_t^c removes the error correcting feature of TD learning through the outer maximisation. This means δ_t^c cannot be cancelled out due to changes in the value function and can lead to the value of addictive states increasing without bound. Since Q-learning uses bootstrapping, after repeated exposure to drugs, this reward propagates throughout the Q-table can cause the agent's optimal policy to be seeking out drug consumption in preference to all other actions[61, 78].

However, because Redish's model's policy is to choose actions in proportion relative to their predicted value, the unbounded increase doesn't mean that drugs are always selected over natural rewards. The likelihood of drug receipt depends on the contrasting natural reward values of the states leading to drug consumption[61]. This aligns with psychological studies in both animals and humans, showing that the quality of alternatives can reduce the uptake and continuation of drug addiction. Rats in environments with plentiful natural rewards such as food and sex have significantly reduced chances of self-administering cocaine or morphine[27, 52, 69]. Similarly, in humans, the monetary value of food vouchers is correlated with increasing periods of abstinence from cocaine abuse[28, 61].

Redish's model satisfactorily establishes the link among neural mechanisms of decision-making, drug-induced alterations, and behavioural evidence[14]. It goes some way to explaining the transition into compulsive drug taking and the successes and failures of voucher schemes that predict vouchers will be most effective if used as an early intervention[63].

Nevertheless, Redish's model's simplicity leads to its shortcomings. The unbounded nature of drug-induced reward cannot reasonably reflect the biological mechanisms in our brain which have finite dopamine[14]. Furthermore, the model cannot predict blocking since the new neutral stimulus will always be associated with the positive error signal, which results from the ever-increasing reward[33]. Most importantly, the model doesn't address the reduced motivation for natural rewards and decrease in natural reward processing that can result from repeated drug exposure[31]. The model predicts natural reinforcers maintain their reward after drug experience, which is inconsistent with clinical trials that show long-term changes in the processing of rewards in humans and animals addicted to cocaine[1, 14].

3.3 An Improved Approach to Modelling Addiction

Dezfouli's (2009) research[14] improves the value function provided by Redish's (2004) design[61], building upon the same hypothesis of drug addiction directly increasing dopamine, with increased focus on the long-lasting dysregulation of the reward processing system as a result of SUDs[14, 40]. It overcomes problems with the previous model, capturing blocking and introducing a maximal basal reward level for addictive actions to reflect dopamine more realistically. Moreover, Dezfouli's extended model captures the progressive elevation of the reward threshold with long-term drug consumption, making it consistent with experiments showing a reduction in natural reward perceived after prolonged abuse[18, 21].

To do this, Dezfolui built upon Daw's (2003) model[11], based on average reward RL[47]. This represents state-action values as the sums of differences between observed and average reward[14]:

$$Q(S_t, A_t) = E\left[\sum_{i=t}^{\infty}(R_{i+1} - \bar{R}_i) \mid S_t, A_t\right] \tag{3.4}$$

where $\bar{R}_t$ is the exponentially weighted moving average of experienced reward defined by:

$$\bar{R}_{t+1} \leftarrow (1-\sigma)\bar{R}_t + \sigma R_{t+1} \tag{3.5}$$

In the case $\sigma \ll \alpha$, this would produce a TD error defined by:

$$\delta_t = R_{t+1} + V(S_{t+1}) - V(S_t) - \bar{R}_t \tag{3.6}$$

Under this error, at each time step t, if no rewarding action is taken, the reward is perceived to be a loss of $\bar{R}_t$, which the agent interprets as missing out on gaining the expected reward. This value function, therefore, guides action selection to π^* based on maximising the expected reward per time

step[14]. Through $\bar{R}_t$, the basal reward level is determined, which prevents the unbounded growth seen in Redish's model when estimating reward from substance use. Daw proposed that $\bar{R}_t$ represents the tonic dopamine level in our brains: the baseline level of dopamine which helps regulate motor function, mood, and motivation[11]. It's neurologically best to think of $\bar{R}_t$ as the level at which phasic rewards δ_t are measured against. However, as TD RL mediates reward through the error signals, it's reasonable to consider this computationally as the level relative to which rewards are measured[14]. To decrease the sensitivity to natural reward with prolonged drug intake, it's necessary to elevate the basal level $\bar{R}_t$ beyond the normal threshold[18, 21]. A factor κ is therefore introduced to represent the level of deviation from the baseline $\bar{R}_t$, giving a biased basal reward level through:

$$\rho_t = \bar{R}_t + \kappa_t \tag{3.7}$$

The κ_t component models the prolonged effect of drug-taking on the system:

$$\kappa_{t+1} = \begin{cases} (1-\lambda)\kappa_t + \lambda N_d & \text{if drug state} \\ (1-\lambda)\kappa_t & \text{else} \end{cases} \tag{3.8}$$

where N_d is the maximum level of deviation, and $\lambda \ll \sigma$ controls the speed of devation. Initially, κ_t starts at a stable level of 0, where it remains unless an action representing drug consumption is taken. If this occurs, the deviation increases slightly, raising the basal threshold as a consequence. Through repeated drug exposures, this has the effect of gradually decreasing the agent's sensitivity to reward. When natural rewards are taken the deviation decreases slightly, reducing the alterations caused by the drug exposure. This means if no drugs are consumed, the basal level will correct itself over time.

Substituting ρ_t from eq. (3.7) in place of $\bar{R}_t$ in eq. (3.5) and eq. (3.6) gives:

$$\bar{R}_{t+1} \leftarrow (1-\sigma)\rho_t + \sigma R_{t+1} \tag{3.9}$$

$$\delta_t = R_{t+1} + V(S_{t+1}) - V(S_t) - \rho_t \tag{3.10}$$

Combining the new error signal in eq. (3.10) with Redish's δ_t^c in eq. (3.2), Dezfouli's model gives:

$$\delta_t^c = \max\left(R_{t+1} + \max_a Q(S_{t+1}, A_t) - Q(S_t, A_t) + D(S_t),\ D(S_t)\right) - \rho_t \tag{3.11}$$

Note that since ρ_t is not related to phasic dopamine activity, being independent of drug-based stimulation, it is not included inside the max operator. To keep the model consistent, it's necessary to account for $D(t)$ in the average reward (eq. (3.9)). This requires updating the reward experience at each time step based on eq. (3.10), replacing R_{t+1} in eq. (3.9) with:

$$R_{t+1} = \begin{cases} \delta_t^c - V(S_{t+1}) + Q(S_t, A_t) + \rho_t & \text{if drug state} \\ \delta_t - V(S_{t+1}) + Q(S_t, A_t) + \rho_t & \text{else} \end{cases} \tag{3.12}$$

Overall, this model is reasonably good, producing an agent that can incorrectly choose addictive

behaviours over more rewarding actions and result in compulsive drug taking. It maintains the advantages of Redish's (2004) model while overcoming some of its flaws. Importantly, it doesn't require an unbounded reward, making it more concrete and explainable at the neuronal level[14]. It also stimulates the decrease in reward sensitivity seen in those with prolonged drug exposure[18, 21]. These both reflect the alteration in the brain's reward systems, biologically interpretable as the model predicting brain plasticity, a key feature of learning.

Furthermore, the model can predict the blocking effect for drug rewards, and increasing impulsive choice as a result of addiction. The direction of causation between drug addiction and increased impulsivity isn't straightforward and depends on the drug being used. There have been several studies supporting that cocaine use leads to increased impulsivity[6, 55]. As cocaine is the basis for both Redish's and Dezfouli's models, seeing the effect emerge from the abnormal elevation of the basal reward level could be seen as an advantage of the design.

Finally, rather than discounting exponentially as in the case of the eq. (2.17) and eq. (3.1), Dezfouli's TD error in eq. (3.10), δ_t^c, discounts hyperbolically. This is consistent with results in human decision-making[2], reinforcing the biological basis of Dezfouli's design and increasing the plausibility and applicability of its predictions.

Nevertheless, there are still drawbacks to this design with several stemming from the agent's lack of a model. This means that the transition of control from the goal-oriented to the habit-based brain system cannot be accurately modelled. Multi-process frameworks of decision making[62, 74] must be used to overcome this, and they are an important step to an understanding the effect on different brain regions throughout the various stages of addiction[14].

Another crucial aspect of addiction that cannot be predicted is relapse. Standard TD learning models don't differentiate learning and unlearning. This means, despite being able to characterise the slow decrease in values associated with extinction of behaviour, they cannot capture the spontaneous recovery when it is renewed[63]. Therefore, in the pre-existing models, learning occurs at the same rate as before exposure to drugs, and relapse is not shown. Extinction must be about learning more, rather than removing what's been previously learned, an effect that cannot be captured by TD RL and consequently mandates a MB component, such as that in this project's dual process model, to be simulated accurately.

3.4 A Dual Process Model Capturing Cue Triggered Relapse

The RL algorithms used in the pre-existing designs have relied on the fact that TD learning converges to an optimal policy π^* in a world that is completely described and stable. However, this doesn't reflect the real world, where a multitude of information is processed to derive our current situation. It's necessary for us to identify what's important, what's not, and infer hidden information from our surrounding environment to categorise ourselves into a state based on prior experience sharing similar properties[63]. Practically, this means our internal states represent salient observations, including notable events and environmental configurations. As with RL, the value in each state is represented

as a time-discounted sum of future reward[25, 78, 86], enabling us to make informed choices based on our expectations of the outcomes from each of our actions[63].

Redish (2007) proposed a model consisting of two processes: a TD RL evaluation function determining the value of taking an action given a certain state, and a situation recognition mechanism that categorises observable cues based on known situations[63]. Having been in contact with Redish, the recommendation was made for the dual process model in this project to build a classification component from a preprint of Pettine's (March 2023) work on latent-state learning[59]. This simulates how people approximate external environments with simplified internal representations based on experience, and how these can then be generalised to new situations[59]. The building blocks of the internal model are latent causes, explanatory constructs that can be inferred from the environment and partition observations into coherent clusters[20].

Combined with the previous designs[14, 61], this gives both a value evaluation component through eq. (3.11), and a state classification mechanism through internal latent-states[59]. The integration between these two components is illustrated in figure 3.2. New states are produced when the consequences of an action provide significantly less reward than would be expected in the current context. This adds to the body of previous experience, and enables new, similar observations to be classified in future. Through the state expansion mechanism, acquisition, extinction and relapse can be predicted[14, 63]. Initially, the addictive behaviour is acquired by associating cues from the context with the reward provided. This builds an internal representation of the expected outcome of addictive behaviour. During extinction, a new, parallel state space is created containing different estimates for the same behaviours. Consequently, when the agent is re-exposed to the cues related to addictive choice, its internal model returns to the initial space, relapsing back into addiction[63].

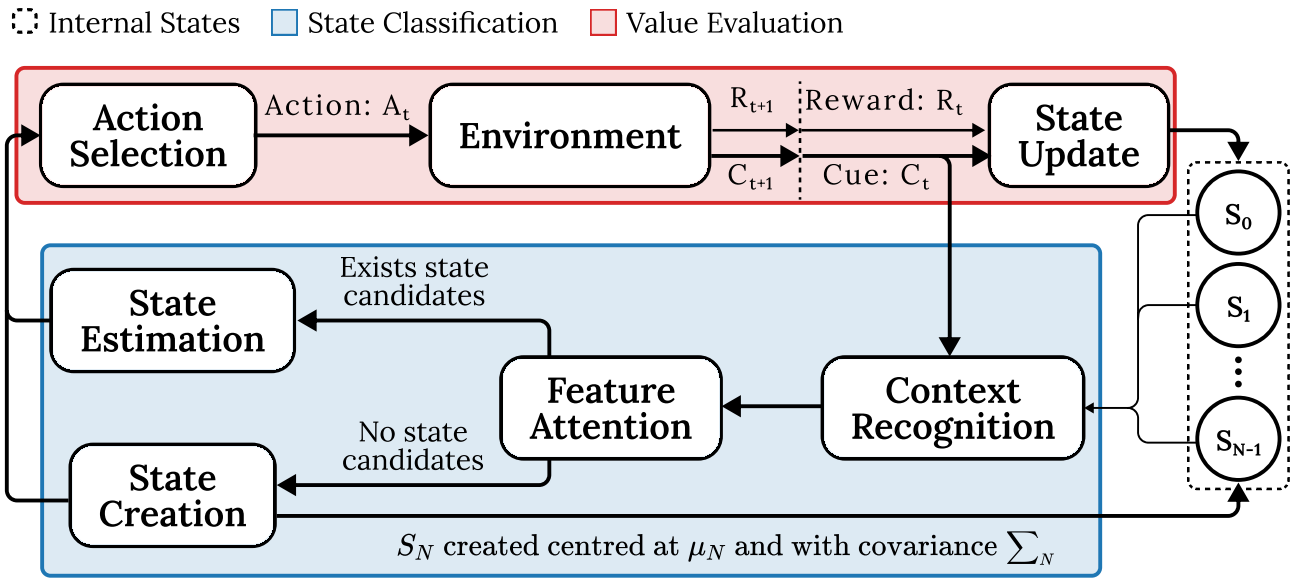

Figure 3.2: The dual process model unifies state classification and value evaluation components

3.4.1 State Classification

On each time step, the environment provides a multi-dimensional cue $c(t)$, e.g. $[0, 1, 1]$, distorted with a small amount of noise ξ_{bl} to the agent. This is compared with each idealised internal state, to predict how the cue can be best generalised to an internal representation based on prior experience.

Context Recognition

To estimate the context, it's necessary to find how surprising an association between each state and the cue is[59]. First the distance from each states' prototype μ and the cue $c(t)$ is calculated:

$$Z_i\left(c(t)\right) = w_A(1-\xi_{\text{bl}})(c(t)-\mu_i) \tag{3.13}$$

where w_A (defined in eq. (3.21) implements the top-down attention representing the cue weight, altered based on a weighting function w_k (defined in eq. (3.20)) that depends on the reward history $\bar{\delta}$ (defined in eq. (3.24)). ξ_{bl} represents a linear function that reduces the distance between cues and internal states, representing decision uncertainty or other factors bringing stimuli and internal states closer in a perceptual space[59]. For each $Z_i(c(t))$ the Mahalanobis distance is calculated:

$$D_i^2(c(t)) = Z_i(c(t))^T \Omega_i Z_i(c(t)) \tag{3.14}$$

where Ω_i is the precision matrix, the inverse of the covariance matrix ($\sum_i^{-1}$) of a multivariate distribution. This is then transformed into a Gaussian distribution, normalised so $\forall w_k = 1$, that measures the probability a cue $c(t)$ could be produced by a multivariate normal of n_c dimensions, centred at μ_i and with covariance matrix $\sum_i$. From this an activation for the state can be calculated:

$$P(c(t)|S_i) \propto \frac{1}{\sqrt{2\pi^{n_c}\,|\sum_i|}}\left(-\frac{1}{2}D_i^2(c(t))\right) \tag{3.15}$$

The surprise for each state can then be given as:

$$F(S_i) = -\ln P(c(t)|S_i) \tag{3.16}$$

When a cue is given by the environment, the surprise is calculated for each internal state through eq. (3.16). All states exceeding a threshold ξ_{th} are then discounted from the set of state candidates.

Feature attention

If there are multiple candidate states for a cue, a discriminative step is taken weighting each feature of the cue to maximally differentiate between potential states[59]. This means if a feature is common across numerous contexts, it will be paid less attention than one unique to a certain situation. This is implemented through the computation of mutual information (MI) for each cue (eq. (3.19)) which

first involves calculating the entropy of cue k, across all states:

$$H(C_k) = \sum_t P(c_k) \log_2 P(c_k(t)) \tag{3.17}$$

where $P(c_k(t))$ is the probability of cue k based on the results from past trials up to the time t. C_i is the matrix of all cue vectors up to time t in the memory of state S_i. The entropy of a cue vector for each individual state can be calculated with:

$$H(C_k^i) = P(S_i) \sum_{t \in S_i} P(c_k) \log_2 P(c_k(t)) \tag{3.18}$$

where C_k^i is a slice of C_i for feature k. The agent calculates the MI between each cue with:

$$I_M(C_k, \mathcal{S}) = H(C_k) - \sum_i H(C_k^i) \tag{3.19}$$

The mutual information (MI) is defined as the increase in entropy in the observed cue distribution. It's a vector containing the I_M value for each feature across all internal states. Intuitively, the mutual information measures the dependence between the cue's feature and the cue. Low mutual information indicates that the feature isn't very relevant or informative. Alternatively, high mutual information indicates that attention should be increased for that feature[59].

The agent modulates the MI of cues based on its reward history $\bar{\delta}$ (eq. (3.24)):

$$w_k = 0.5 + 0.5 \tanh \frac{I_M(C_k, S) - 0.5}{\xi_{\text{cw}}} \tag{3.20}$$

$$w_k^e = \left(1 + \tanh \frac{\bar{\delta}}{\xi_{\text{DB}}}\right) w_k - \tanh \frac{\bar{\delta}}{\xi_{\text{DB}}}$$

$$w_A = (1 - \xi_{\text{distortion}})\, w_k^e + \xi_{\text{distortion}} \tag{3.21}$$

where ξ_{cw} controls the slope of the sigmoid curve, and ξ_{DB} is a standard squashing parameter controlling the rate at which changes in $\bar{\delta}$ change w_A. The idea behind this modulation is that if the current internal states of the agent do not correspond well to any of the latent states, then the agent should attend more to cues. This occurs since as $\bar{\delta}_t \to -\infty$, $\tanh \frac{\bar{\delta}}{\xi_{\text{DB}}} \to -1$ and $w_A \to 0$, it increases the attention given. In contrast, as $\bar{\delta}_t \to 0$, $\tanh \frac{\bar{\delta}}{\xi_{\text{DB}}} \to 0$ and $w_A \to w_k$, it reduces the attention paid[63].

State estimation and creation

A list of suitable state candidates can be found using eq. (3.13) with the top-down attention w_A modulated by the discriminative attention weights as described in eq. (3.21). If there are no candidates under a threshold surprise ξ_{th}, a new state S_N is appended to the N current internal states, centred at $\mu_N = c(t)$ and with a spherical covariance matrix $\sum_N$ with variance $\sigma_N^0 = 25$. A burn-in time of n observations is applied to each new state S_N, after which μ_N and $\sum_N$ are updated respectively as the mean and covariance matrix of all cues classified as S_N. This causes states with stable observations to

tighten their variance, while those with variable observations will increase it to cover a broader range in cues[63]. When multiple internal states were found to be below the surprise threshold, the agent employs a standard asymmetric softmax function across all candidates to estimate the most likely internal state:

$$P(S_t|c(t)) = \frac{e^{\eta_{Sn} P(c(t)|S_i)}}{\sum_i e^{\eta_{Sd} P(c(t)|S_i)}} \tag{3.22}$$

η_{Sn} and η_{Sd} are the "temperature" parameters that alter how sharp the softmax function is. Since two parameters are used, it enables control over the balance between selecting high-scoring and low-scoring states. Overall, these values, therefore, govern the exploration/exploitation trade-off for state selection. When η_{Sn} is high and η_{Sd} is low, it emphasises the low-surprise states, encouraging the exploitation of the best options. Conversely, when η_{Sn} is low and η_{Sd} is high, the agent explores more with the shallower curve, providing a wider distribution across the values. Being able to alter this exploration and exploitation is especially useful when done in relation to $\bar{\delta}$, incentivising the agent to take more risks when the policy is unclear or is performing poorly, then become more conservative as it becomes more confident of the relationship between cues and states.

3.4.2 Agent Value Evaluation

Once the agent has predicted its internal state based on the cue, an action is chosen based on policy π and a reward will be returned from the environment. This is used to update both the Q-table value for the internal state $Q(S, A)$ and the reward history $\bar{\delta}$.

Action Selection

Like with the state selection, the policy for action selection uses a standard asymmetric softmax function based on the predicted values for the internal representations, $Q(S_t, A_t)$:

$$P(A_t|S_t) = \frac{e^{\eta_{An} Q(S_t, A_t)}}{\sum_a e^{\eta_{Ad} Q(S_t, a)}} \tag{3.23}$$

As was the case in state selection, η_{An} and η_{Ad} control the exploration/exploitation trade-off, altering how risk-taking the agent is in its chosen actions by the varying gradient of the softmax distribution.

State update

The reward prediction error (RPE) δ_t^c is taken from eq. (3.10) in the second design, to also capture the influence of drug taking within this model. The update remains the standard update from the first model given in eq. (3.3). Alongside this update, an history additional recorded an exponentially decaying average of the recent adverse rewards:

$$\bar{\delta} = \xi_0 \bar{\delta}(t-1) + \xi_1 \min(0, \delta_t^c) \tag{3.24}$$

where ξ_0 and ξ_1 alter the rate at which $\bar{\delta}$ changes. Both fast and slow timescales are used to enable the agent to distinguish between volatile and stable environments with low rewards. These were given by $\bar{\delta}_{\text{fast}}$ and $\bar{\delta}_{\text{slow}}$ and both were calculated as shown in 3.26. Initially, baselines were set for $\xi_0 = 0.99$ and $\xi_1 = 1.5$. Both timescales were then shifted using:

$$\Delta\xi_0 = \xi_0 + \omega \tag{3.25}$$

$$\Delta\xi_1 = \xi_1 \frac{1 - \Delta\xi_0}{1 - \xi_0} \tag{3.26}$$

Again, fast and slow timescales are used, with $\omega_{\text{fast}} = -0.009$ and $\omega_{\text{slow}} = 0.004$. A $\bar{\delta}_{\text{effective}}$ was determined on each trial through the difference between the fast and slow timescales:

$$\bar{\delta}_{\text{effective}} = \min(0, \bar{\delta}_{\text{fast}} - \bar{\delta}_{\text{slow}}) \tag{3.27}$$

with both $\bar{\delta}_{\text{fast}}$ and $\bar{\delta}_{\text{slow}}$ being reset to zero when a new internal state was created.

3.4.3 Capabilities Relative to the Pre-Existing Designs

This new model builds on the pre-existing designs, and therefore maintains all of their advantages, but provides further features due to containing a model. Primarily, this enables it to simulate the spontaneous recovery of a behaviour as a result of re-exposure to a context associated with the addiction. It can, therefore, predict the effect of cue-triggered relapse[19, 53, 66]. It differentiates the learning and unlearning process by creating a parallel state space, which causes re-exposure to move the agent back into a situation overvaluing the problem behaviour, leading to it again being chosen inappropriately.

Furthermore, this model can capture behavioural addictions that don't stimulate the dopamine system directly. This is beneficial in moving the model beyond pharmacological addictions that drive increases in dopamine. Even within SUDs, dopamine isn't the sole effect that makes drugs addictive, and isn't relevant to behavioural addictions. As indicated in the introduction, this has been focused on less than drug addictions in research, so it could be particularly insightful with rising problematic behaviours around gambling, overeating, social media, etc.

Nevertheless, some issues could still be addressed. The model doesn't have any control over the dosage of drug-taking activities, which limits its ability to create realistic simulations of tolerance and withdrawal. Equally, the role of dopamine in aversive situations remains controversial, especially for reward-related processing in mammals, including humans[78]. This means that tests examining the role of punishment on addiction would need to be taken with caution since the model's validity may be limited for this.

4 Implementation

Having explained the mathematical details of the design, this chapter aims to focus on the technical implementation, providing algorithms explaining some of its key features. Despite this project's research focus, there was a sizeable software engineering component. This involved the creation of a custom OpenAI Gym[9] environment and a class to represent agents for each of the designs outlined previously. Object-oriented programming practices[89] were used to ensure the implementations were modular, flexible, and extensible. Hopefully, the foundations built could reduce the time required to develop new simulations in future work and support productivity.

4.1 OpenAI Gym

OpenAI Gym is an open-source framework for developing and comparing reinforcement learning (RL) algorithms. Alongside many standardised RL environments ranging the frozen lake example given in the background (fig. 2.6) to Atari Games[1], OpenAI Gym enables the development of custom environments.

4.1.1 Core Environment Components

There are five key aspects that encapsulate most behaviour of an environment:

1. An *observation space* describing the format and number of states in the environment.
2. An *action space* defining the set of actions an agent can take in each state within the environment.
3. An internal *reward function* acting as a reward mapping over the actions and states.
4. A *step function* providing the dynamics of the environment. Given an action, this updates the environment's state, returning a new observation, reward and flags indicating additional information about the episode.
5. An optional *render function* displaying the environment in a graphical user interface (GUI).

Combined, these components provide a standardised interface for RL agents to interact with the environment, enabling focus to be placed on agent development without needing to consider the specifics of how the agent and environment communicate. This was particularly useful in this project, facilitating different design iterations to be tested in the same environment, ensuring consistency across the experiments and simplifying the comparison of results.

4.1.2 Creating a custom environment

The creation of a custom environment involved extending the inbuilt `gym.Env` class, then creating implementations for the properties and methods outlined above, shown in fig. 4.1 and algorithm 2.

[1] Examples can be found at `https://gymnasium.farama.org/environments/atari/`

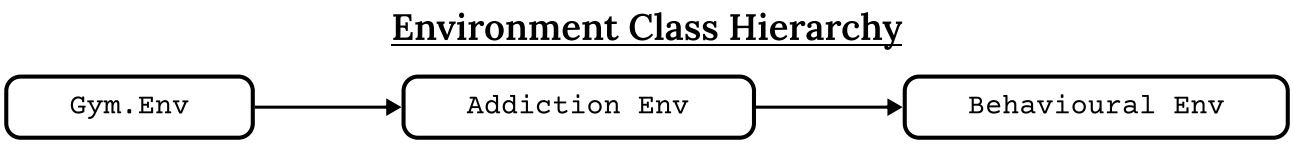

Figure 4.1: Class Hierarchy for the custom OpenAI Gym environment

Algorithm 2 Environment initialisation

Require: MAX STEPS
Ensure: $S_{\text{init}} \in \mathcal{S}$

Class ADDICTION ENV **extends** GYM ENV
  ⋮
  function CONSTRUCTOR(N_{states}, N_{states})
    $Observation\ Space \leftarrow N_{\text{states}}$
    $Action\ Space \leftarrow N_{\text{Actions}}$
    $render \leftarrow None$
    $S \leftarrow S_{\text{init}}$
  ⋮
  function STEP(A)
    $R \leftarrow$ GET REWARD(A)
    $S \leftarrow$ INCREMENT STATE(A)
    $C \leftarrow$ GET CUE() // Cue required for design 3

    $terminated \leftarrow t \geq$ MAX STEPS
    $truncated \leftarrow A_t \notin \mathcal{A}$ // Environment is robust to incorrect actions
    $info \leftarrow$ RECORD INFO(A)

    return $\langle S$ or C, R, $terminated$, $truncated$, $info\rangle$ // Choose S or C depending on design

In the constructor of the environment, the action space and observation space were initialised. The Python implementation used Gym's `Discrete` type, the same as used for the states and actions in the frozen lake example (fig. 2.6), indicating that the states and actions are best described as values that can be indexed as integers. It differs from the `Box` type in Gym, which represents a continuous range and would be more suitable for representing values like angles. The constructor also sets the render function to *None*, revealing the simulations have no GUI. This is because there isn't any useful graphical information to be displayed throughout the training of the agents in this project. Furthermore, hiding the GUI removes the overhead of running the virtual screen, improving the performance during the simulation.

The STEP method followed the typical structure of the base `gym.Env` class, taking an action, and returning a tuple with five pieces of information:

1. An observation, O, which is either the state or cue provided having taken the given action.
2. A numeric reward, R, obtained having taken the action, this gained through the reward function, and was varied in different experiments to test the effect of reward on addictive behaviour.
3. A boolean *terminated* signal reporting that the end goal of the MDP has reached, using STEP beyond this point could result in undefined behaviour.
4. A boolean *truncated* signal indicating that something beyond the scope of the MDP has ended the

simulation - e.g. an invalid action selection.

5. An *info* dictionary which provides auxiliary information about the environment at that point. For the simulations in this project, it details things such if an action was optimal or addictive.

The step function can then be used in a training loop similar to that in figure 2.1 of the background, enabling the agent to receive rewards and update its policy. For the agents in this design, a typical training loop is given by:

Algorithm 3 Agent Training Loop

Require: MAX EPISODES
Ensure: $N_{\text{states}} \in \mathbb{Z}^+$, $N_{\text{states}} \in \mathbb{Z}^+$

$env \leftarrow$ ADDICTION ENV($N_{\text{states}}, N_{\text{states}}$)
$agent \leftarrow$ AGENT(...) // Initialise agent based on design

for $e \leftarrow 1$ **to** MAX EPISODES **do**
    O, $info \leftarrow env$::RESET() // Return environment to default
    repeat
        $A \leftarrow agent$::GET ACTION(O)
        O', R, $terminated$, $truncated$, $info \leftarrow env$::STEP(A)
        $agent$::TRAIN(O, A, R, O')
        $agent$::RECORD INFO($info$) // Record info for experiments
        $O \leftarrow O'$
    until $terminated$ **or** $truncated$

Algorithm 3 follows the standard "agent-envrionment loop" for OpenAI Gym. When the environment is reset through the RESET method, it returns to the default state for that environment, along with some initial information. The agent then uses its GET ACTION method to choose an action based on the observation, O. The environment uses this in its STEP method to increment the world state, returning a new observation, O', a reward, R, and some additional information, *info*, as a result. From this feedback, the agent can train its policy through its TRAIN method, and record the details for later interpretation. The next observation, is then stored in place of the current one with the process repeating until the environment indicates that the episode is over, indicated by either *terminated* or *truncated*. This occurs after a predetermined number of time steps (MAX STEPS in algorithm 2). The agent repeats the process for numerous episodes, as determined by MAX EPISODES. This is useful since it enables performance to be evaluated at different intervals and enables the agent to move between different implementations of the environment, aiding the evaluation of how the designs react in changing conditions. The custom environment and agent implementation make this easier by including lots of tracking, meaning the data is readily available for analysis.

4.1.3 Evaluation of OpenAI Gym

OpenAI Gym provided a valuable interface for creating the custom RL environment that enabled the simulations to be produced more quickly. Through inbuilt functions such `check_env`, it was simple to verify that the custom environment conformed to best practices for the API, preventing issues from being missed and then arising later in development.

Due to its flexibility, there were very few drawbacks to using OpenAI Gym within this workflow. One

minor inconvenience was converting the latent state model provided with Pettine's work, which was used for the dual process model's state classification component, into the format required for OpenAI gym. This was challenging since his work didn't align well with the API used in Gym. However, Gym itself is not really at fault for this issue, and in fact, Gym is popular among researchers and developers, boasting a large community who work on RL problems.

4.2 Creating agents

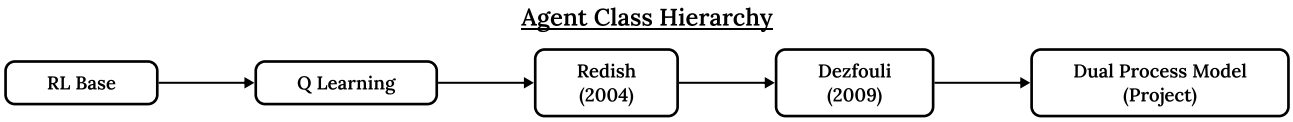

Figure 4.2: Class Hierarchy for the agent classes

The agents were all inherited from a custom RL BASE AGENT abstract class created for this project. This provided an interface including the GET ACTION, TRAIN and RECORD INFO methods previously seen in algorithm 3, alongside also other useful methods for gathering policy and saving the agent. Agents for each design inherited aspects sequentially, (fig. 4.2), with the value evaluation component of the dual process model being inherited directly from TD UPDATE method from the Dezfouli (2009) implementation (algorithm 4). The state classification component was implemented directly in the dual process model since this was unique to this design. Using inheritance meant fixing mistakes within any of the designs also propagates the correction to the inheriting designs. This made the code more maintainable, increasing the likelihood that the output results were correct, and making them at least consistent in the worst case. Furthermore, extensively logging events throughout the simulations in the RL BASE AGENT class made the information readily available throughout all implementations. This meant the results were readily available for analysis, and future experiments could be achieved by altering the data collected in one place. The TD UPDATE step within the agent learning requires

Algorithm 4 TD UPDATE implementation in the final design

function TD UPDATE(S, R, A, S')
  $\delta \leftarrow R + \max_a Q[S'] - Q[S,A]$
  $X \leftarrow 0$

  if A **is** Addictive **then**
    $\delta \leftarrow \max(\delta + D(S),\ D(S)) - \rho$
    $X \leftarrow N_d$

  $R \leftarrow \delta - \max_a Q[S'] + Q[S,A] + \rho$
  $\delta \leftarrow \delta - \bar{R}$

  $\bar{R} \leftarrow (1-\sigma)\bar{R} + \sigma R$
  $\kappa \leftarrow (1-\lambda)\bar{\kappa} + \lambda X$

  $Q[S,A] \leftarrow Q[S,A] + \alpha\delta$
  return δ

passing both a state, S, and action, A as arguments. The Q-learning implementation used an ϵ-greedy strategy for exploration, in which there's a $1-\epsilon$ chance of exploiting the best action, and an ϵ chance that any other action (as in algorithm 1). The remaining designs used a softmax function as shown inalgorithm 5, with the probability of each action being picked relative to their weights. This alteration is because ϵ-greedy's simplicity results in the drawback of equally weighting actions regardless of their expected reward when exploring. Those thought to be almost optimal will be chosen with the same chance as those considered terrible as the random selection doesn't differentiate between anything but the best action and everything else. Instead, when a softmax function is used this is no longer the case, with better actions being made preferable, even if they weren't the best action. Additionally, this has the benefit that there aren't any sharp, discontinuous changes in behaviour as the optimal action changes. If two actions are perceived as optimal, they will be chosen equally often. This is more realistic of human behaviour, where changes occur gradually and with variation between choices where the expected outcomes are similarly rewarding. The dual process model also

Algorithm 5 GET ACTION implementation in the final design

```
1: function GET ACTION(O)
2:     S ← IDENTIFY STATE(O)
3:     action values ← Q[S]
4:     P(A| S) ← SOFTMAX(action values)
5:     A ← RANDOM ACTION(probabilities = P(A| S))      // Use softmax weights in policy
6:     return A
```

uses a softmax function within its state classification component as shown in Algorithm 6. Through implementations of through equations (3.16), (3.19) and (3.22), the agent can choose an internal representation based on the likelihood of each internal state based on the previous history of observed cues. It's helpful to look at the algorithm 6 alongside fig. 3.2 to understand how each part in the state classification is actually implemented. The final states selection is done through a softmax function based on the *surprise* assigned to each internal state, S, based on the observed cue, O:

Algorithm 6 State identification implementation

```
1: function IDENTIFY STATE(O)
2:     State Suprises ← Array[]
3:     for S in Q do                                   // Iterate internal states
4:         μ_S ← μ_S w_A
5:         surprise ← −ln P(O| S)
6:         if suprise ≤ suprise threshold then
7:             State Suprises::APPEND(S)
8:     if State Surprises is empty then
9:         S ← CREATE NEW STATE(O)
10:        Q::APPEND(S)
11:    else
12:        P(S| O) ← SOFTMAX(State Suprises)
13:        S ← RANDOM STATE(probabilities = P(S| O))
14:    return S
```

The dual process model's state classification differs from the previous Redish (2004) and Dezfouli

(2009) designs which had a complete understanding of the world, and therefore need to do no further identification. It was certain whatever state is retrieved from the environment is the actual state. However, this added realism enables the dual process model actually to capture key aspects of addiction such as relapse. It also allows the dual process model to move beyond dopamine-mediated addictions to behavioural addictions since adjustments to behaviour can result from changes in both the state space and the value evaluation. This means addiction is possible without features of transient dopamine increase eq. (3.1).

All agent designs can interact with the environment as shown in algorithm 3. While the RECORD INFO method is useful, its implementation is pretty independent of any technical details for each of the designs, and could be equally applied in any RL scenario. The methods permitting the capturing of addiction within these simulations are the GET ACTION (algorithm 5) and TRAIN methods. The TRAIN for each agent is actually just a combination of the IDENTIFY STATE and TD UPDATE methods seen previously as outlined in algorithm 6 and algorithm 4. This is shown in algorithm 7.

Algorithm 7 Agent Train Method

```
1: function TRAIN(O, R, A, O')
2:     S ← IDENTIFY STATE(O)
3:     S' ← IDENTIFY STATE(O')
4:     δ ← TD UPDATE(S, R, A, S')
5:     UPDATE δ̄(δ)
6:     RECORD INFO(...)
```

All of these methods then can be encapsulated within the AGENT class as shown in algorithm 8:

Algorithm 8 Agent Implementation For Design

```
1: Class AGENT extends PARENT AGENT          // inheritance as shown in fig. 4.2
2:     ⋮
3:     function GET ACTION(O) ...
4:     function IDENTIFY STATE(O) ...
5:     function TD UPDATE(S, R, A, S') ...
6:     function TRAIN(O, R, A, O') ...
7:     function UPATE δ̄(δ) ...
8:     function RECORD INFO(info) ...
9:     ⋮
10:
```

The rest of the implementation of the designs involved creating private methods and implementing the mathematical equations provided in the design section. These don't need to be called directly from outside the environment, which is one of the benefits of the implementation being set up in an abstract way. Despite the designs differing, since they share a common interface, it's possible to use polymorphism within the experiments. This interoperability also makes this work extensible, with the experiments potentially being used on new agents so long as they share the same interface.

5 Experiments and Results

This chapter details the experiments carried out on the dual process model, measuring its performance relative to the pre-existing designs and with respect to the design objectives. There are three phases to the experiments:

1. Acquisition of standard and addictive behaviours to verify that the designs are correct and the results of later experiments are truly a result of the development.
2. Reproducing findings from experiments in the Redish (2004)[61] and Dezfouli (2009)[14] model to validate the model maintains the ability to predict pharmacological addictions.
3. Creation of experiments testing the dual process model's ability to capture relapse and behavioural addiction. This illustrates how this project moves beyond the pre-existing Redish and Dezfouli models.

Having outlined the results, the overall effectiveness of the model will be evaluated in comparison to approaches of modelling addiction outside of RLs. Finally, an assessment with regard to Redish's (2008) criteria for creating a holistic model of addiction[62] is done to judge the generalisability of results produced by this design.

5.1 Simulations

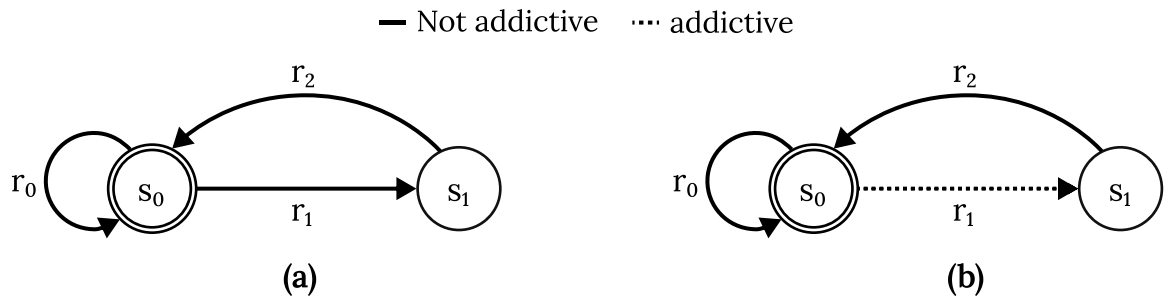

Figure 5.1: Envrionment Dynamics for the Simulations.

Most simulations took place in an environment consisting of two states, shown in figure 5.1. The rewards for each action were normally distributed with varying mean and standard deviations as shown in table 5.1. For addictive actions, the default value of the dopamine surge $D(s)$ was 15.0.

	Mean	Standard Deviation
r_0	10.0	2.0
r_1	−3.0	1.0
r_2	−5.0	3.0

Table 5.1: Reward Dynamics in the Standard Testing Environment

5.1.1 Acquisition of a Regular Response

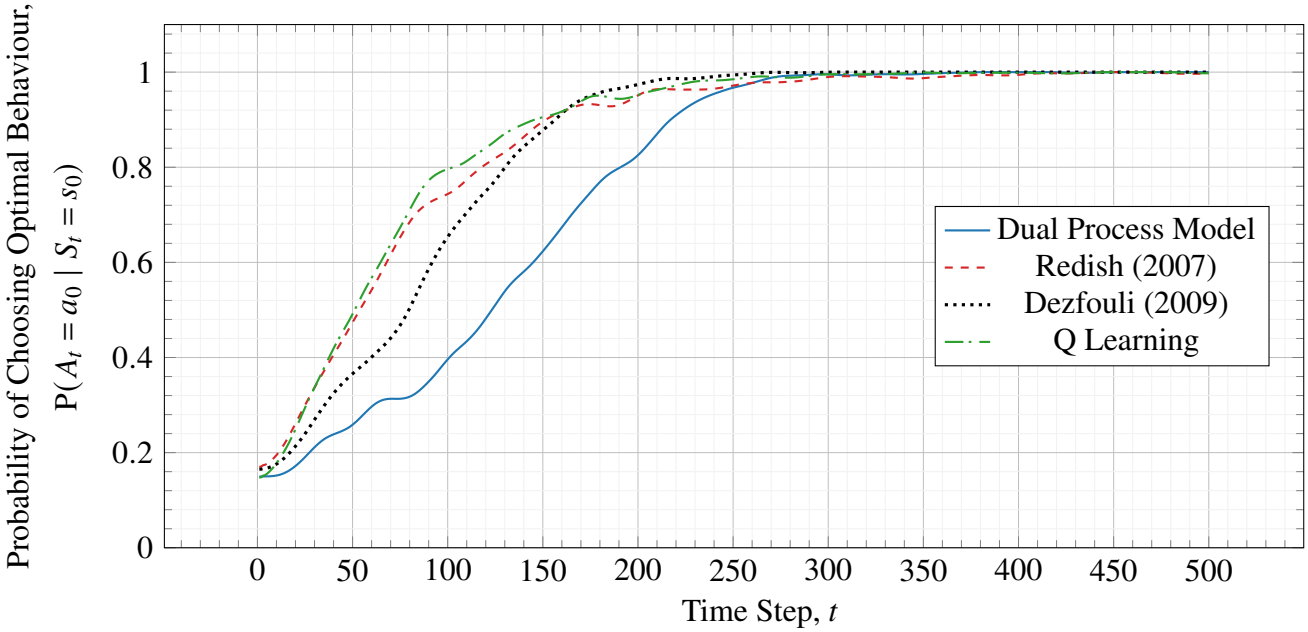

Figure 5.2: Results Showing the Adoption of an Optimal Policy in Response to Natural Reward

First, the simulations verified that the implementations of each agent could adopt an optimal policy in the case of a natural reward as shown in fig. 5.1 (a). This was done by tracking the number of times the agent chose a_0 in s_0, taking an average over 100 repeats of 2000 timesteps. The results of the first 500 timesteps are shown in fig. 5.2.

All the implementations quickly learned the optimal behaviour was to stay in s_0 repeatedly, rising to a probability of around 1. This policy is quickly identified, with the choice of exploring over-exploitation being the main factor limiting the proportion it was chosen early on. This exploration rate decays as the number of timesteps increases, altering the distribution of actions chosen to favour those seen as providing high rewards. This adjustment of the softmax "temperature" is why the probability still tends to 1 in the latter three designs despite the policy selecting proportionally based on the relative reward of actions.

One notable feature of fig. 5.2 is that the dual process model learns slower than the other designs. This is a result of the agent initially having one internal state that tried to represent all world states. Therefore, rewards are assigned from r_0, $r1$ and $r2$ to the same internal state until the state space splits. For stability reasons this only happens after the burn in duration of 15 trials, meaning the initial learning process is distorted. Nevertheless, the optimal policy is still guaranteed to be converged on, meaning it detracts little from the quality of the findings.

Since the designs extended upon each other, this experiment was useful in ensuring every level of the implementation is working as expected and will produce reasonable results from the more involved upcoming simulations.

5.1.2 Acquisition of a Pharmacologically Addictive Response

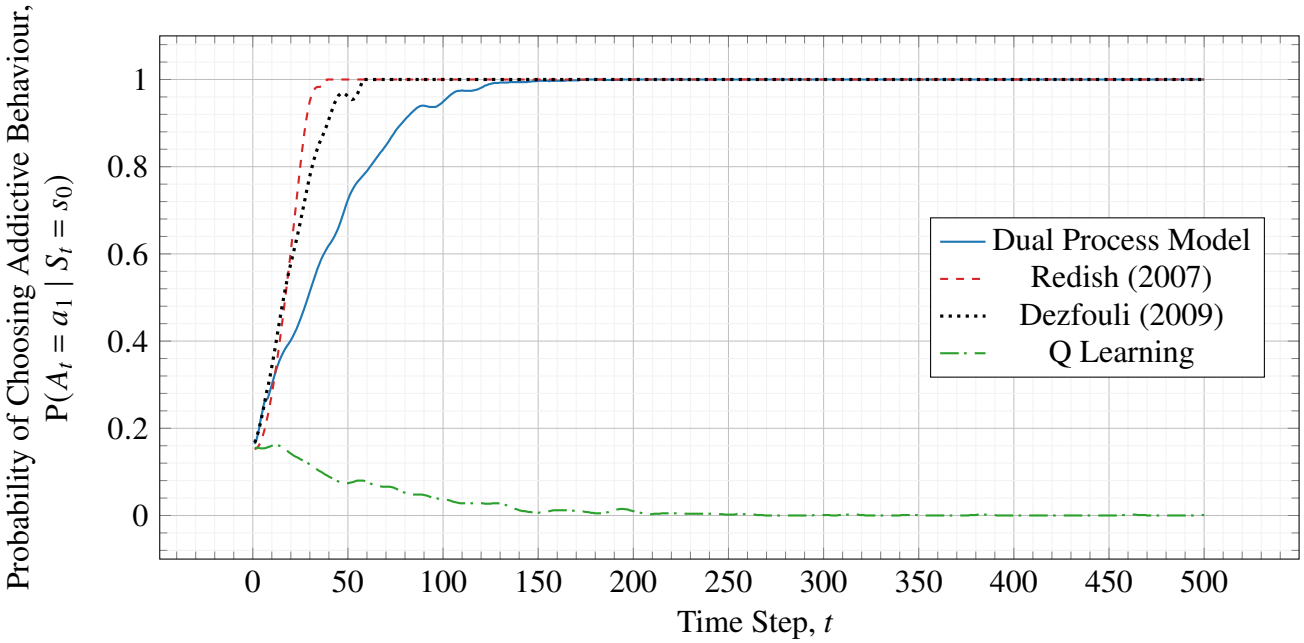

Figure 5.3: Results Showing the Adoption of a Suboptimal Policy in Response to Addictive Reward

Following the verification of the dual-process model's ability to select an optimal policy based on a natural reward, the environment was altered to match fig. 5.1 (b), with an addictive action taking the agent from s_0 to s_1. The first experimental design was then repeated in this updated environment, averaging the trajectories of 100 agents over 2000 timesteps. Figure 5.3 plots the probability of the addictive action a_1 being chosen in state s_0. The results show that both the pre-existing designs and the dual process model choose the suboptimal addictive action. In contrast, the standard Q-learning algorithm continues to pick the optimal policy. This illustrates how standard reinforcement implementations fail to capture addiction and demonstrates how adapted designs can predict pharmacological addictions.

This replicates the findings of Redish (2004)[61] and Dezfouli (2009)[14], supporting that the dual process model implementation is correct. Since this used a value evaluation component based on Dezfouli's designfig. 3.2, it's reasonable that the dual process model follows a similar increase to this model. Again, the slightly slower rate of learning can be explained as a result of the time required for the state space to split to match that of the environmental state space.

The gradient of each of the designs in fig. 5.3 alludes to the disadvantage of the reward tending to infinity in Redish's model. Since this is the case, the softmax quickly provides these actions with a very high probability, making it almost certain they'll be chosen. Comparatively, Dezfouli's design and the dual process model don't have this same indefinite value increase, so the relative values of addictive and non-addictive actions remain closer for longer. This means the agent explores more in the early stages, decreases its probability of choosing a_1 in s_0 until the softmax "temperature" decays and increasingly prioritises exploitation.

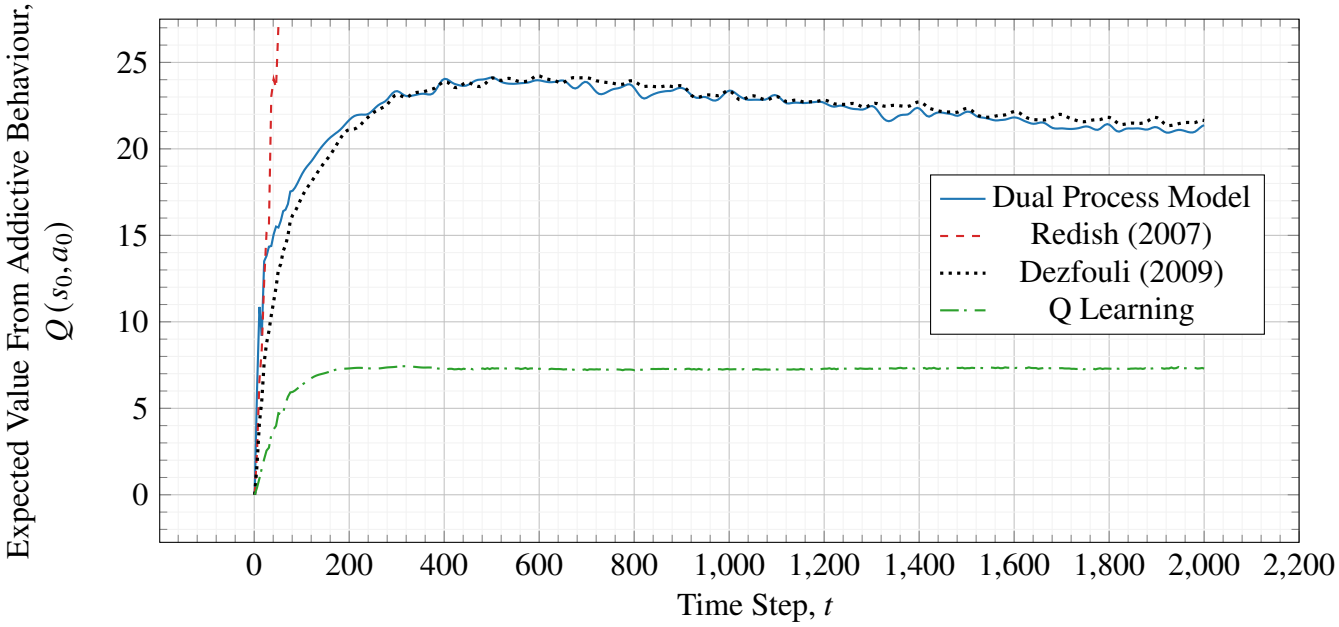

Figure 5.4: Results Showing A Reduction In Perceived Reward After Prolonged Drug Taking

Figure 5.4 reinforces this, showing how the expected value of reward changes throughout 2000 time steps in the experiment. The Redish (2004) model within 50 trials reaches a value beyond the graph limits at 27, and increases to 2400 at $t = 2000$. Conversely, the Dezfouli and project implementations show decreasing reward due to using an exponentially weighted moving average for its temporal difference (TD) error. Furthermore, the experiment replicates Dezfouli's results, showing that the use of a basal limit is effective in replicating the decline in the perception of reward seen in humans after prolonged drug use. This elevation is not limited to only drug rewards; it applies more widely to natural rewards, too. This matches experimental findings in humans, supporting the model's applicability[27, 52, 69, 28, 61].

Impulsivity

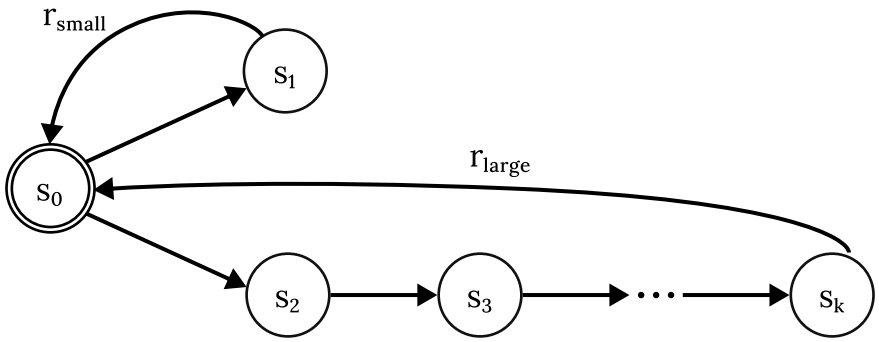

Figure 5.5: Environment for Testing Relationship Between Drug Taking and Impulsivity

After demonstrating the dual process model's design ability in matching Redish (2004)[61] and Dezfouli (2009)[14] and inappropriately selecting addictive actions, further experiments were carried out

to see if the introduction of an internal state space[59, 63] altered the findings from their experiments.

First, impulsivity was investigated, with agents being tested based on varying levels of drug exposure. A control agent was trained in the environment given in fig. 5.1 (a), while two other agents were trained in the environment given by fig. 5.1 (b). During training these agents were exposed to drugs 500 and 2000 times respectively. After training, each agent was placed in a new environment described by fig. 5.5, in which the optimal policy is to take action a_2 to the long path resulting in delayed gratification through r_{large}. However, the results (fig. 5.6) show the proportion of times that a_2 is chosen is altered with prolonged drug taking, with increasing exposure resulting in the agent being more likely to take the small reward in favour of the larger delayed reward. This suggests that increasing drug use can induce impulsivity in the agent, consistent with Dezfouli's model[14]. Moreover, it verifies that the introduction of a state creation component in this project's model doesn't appear to affect the agent's value evaluation concerning impulsivity, with the main difference between the results in fig. 5.6 being the timing of results, which can be explained by variations in the learning rate and the parameters used in the exponential averages (σ and λ).

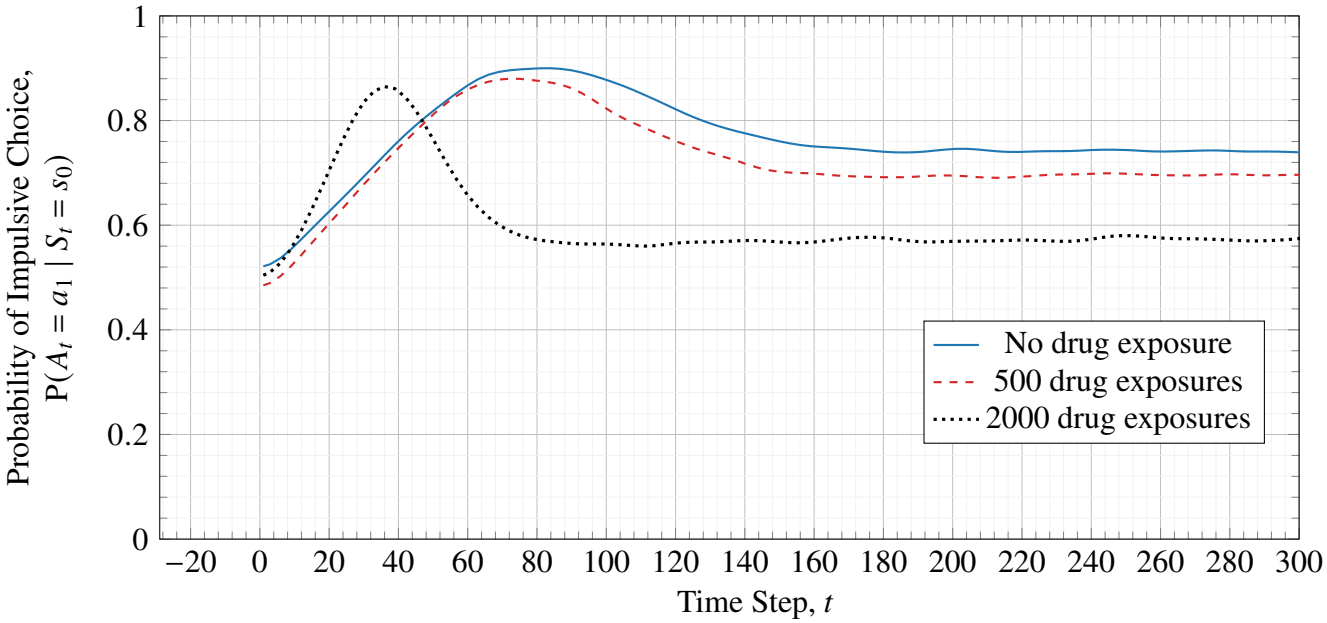

Figure 5.6: Results for Relationship Between Drug Taking and Impulsivity

This finding is reasonable, since impulsivity results from the use of average reward RL, which through inheritance in the implementation is used consistently and Dezfouli's model. Repeated drug use alters the basal reward level ρ_t, making there a high cost of waiting[14]. This means the agent favours immediate reward over the prolonged delay associated with gaining the larger reward. Comparatively, in basal reward alters based on the actual reward received and iin the health system sn't biased. This means the average reward for delayed gratification is higher, and the agent prefers this as its policy.

Blocking

Blocking[33, 46, 65] was identified as an area of interest based on Dezfouli's work, which proposed it as a feature to be improved upon Redish's work. It was also likely something that would interact with the cues involved in the state creation component of this project's agent. Since the state creation system based on Pettines's instrumental latent state system[59] could take multiple cues, this was used over the linear function approximator in Dezfouli's work[14].

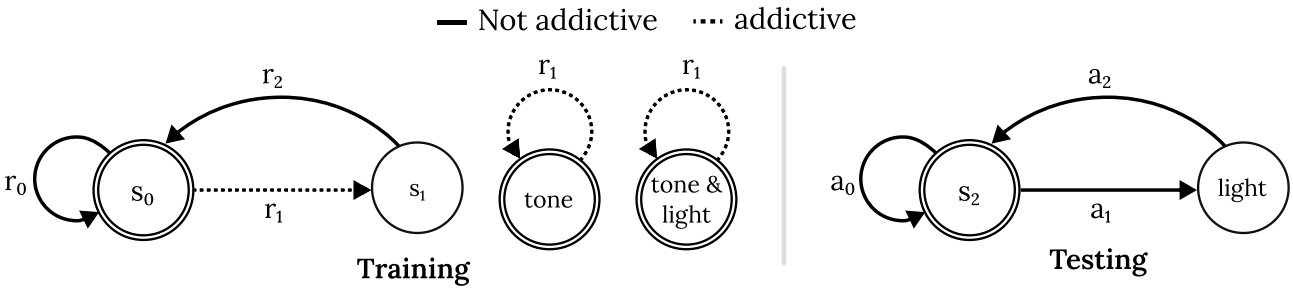

Figure 5.7: Testing Environment for Blocking Simulation

The training stage consisted of three sequential phases as shown in fig. 5.7. Initially, the agent learns an addictive response. Next, the blocked agent is presented with the addictive substance alongside the cue representing the tone, while the non-blocked agent is presented only with the addictive substance. Finally, both the non-blocked and blocked instances are presented with a cue for the tone and a cue for the light alongside the reward for the drug receipt. The testing phase then involved moving the agent to a new environment, shown on the right of fig. 5.7. From s_2, it was possible to keep the agent in the same state with a_0, or move through a_1 to a state providing the cue just the light.

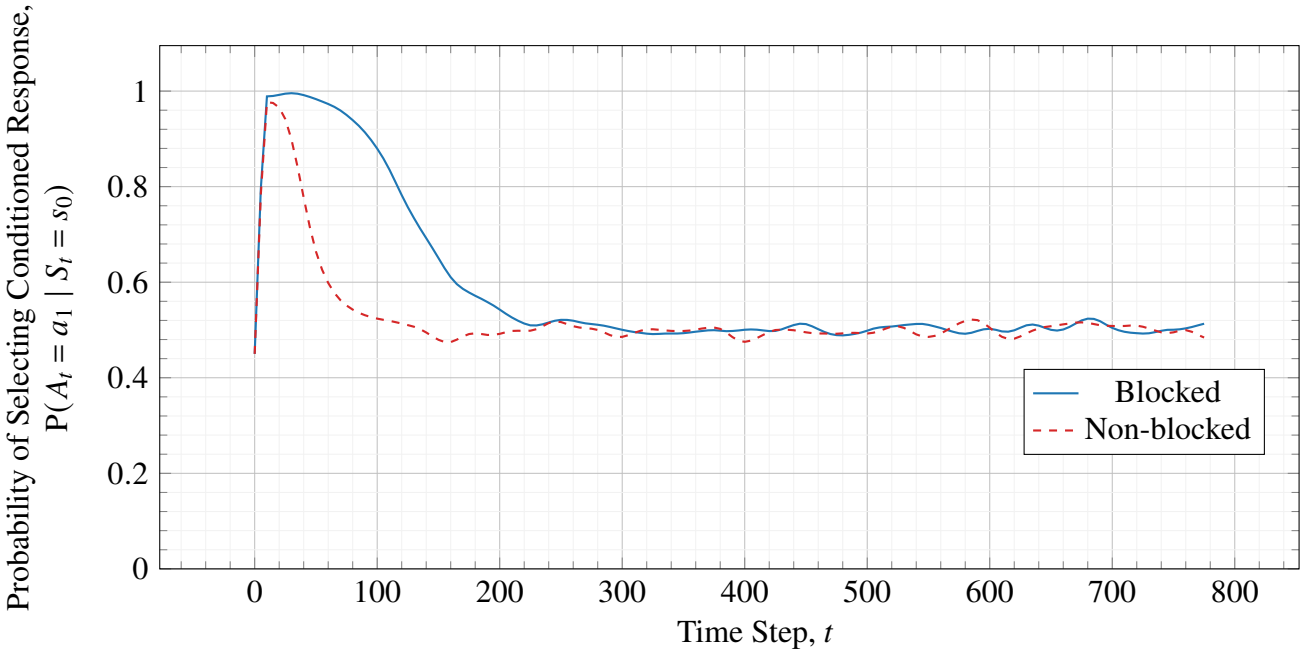

Figure 5.8: Results of Blocking Simulation

The results were plotted in fig. 5.8 and do not show blocking in either case. This was unexpected, meaning the dual processes model misaligns with Dezfouli's (2009) findings, matching instead with those in Redish (2004). An explanation for this is that in both the blocked and non-blocked agents, update their prototype representing the internal state associated with the tone during the third training phase. This causes it to account for the light and regardless of which variation the of blocking phase is used for the agent, the light cue will produce a conditioned response [33, 46, 65].

This further explains why the non-blocked agent unlearns the behaviour quicker than the blocked agent. Since the non-blocked agent is consistently presented with the tone and light presented together the covariance matrix is tightened. Conversely, the tone being presented independently means as the cue is altered to match the simultaneous tone and light presentation, the covariance matrix increases, covering a broader range of cues. Consequently, the light being presented independently causes more surprise in the first agent, which along with the reduction in reward influencing $\bar{\delta}$ (eq. (3.24)) alters the softmax function and increases exploration. This occurs slower in the blocked case since it's already expecting a range of rewards, and thus the change from presenting both cues to a single cue results in a slower alteration in behaviour.

5.1.3 Cue Trigger Relapse

In order to move the dual process model beyond the Redish (2004) and Dezfouli (2007) papers, a model containing internal representation was used. This enables a separate state space to be created during extinction, which in turn can lead to relapse. To test this, the agent was once again trained as shown in section 5.1.2 (b) for 250 trials during which a cue is presented resulting in the adoption of an addictive behaviour. After this, a period of 250 trials of extinction is undertaken, where the cue is no longer presented and the reward dynamics for natural reward are made more favourable. This increase could represent factors such as peer support in helping a family member rehabilitate from an addiction. Finally, the cue is reinstated, and the reward space is returned to the initial environment, triggering the agent to relearn the addictive behaviour.

The results from fig. 5.9 illustrate how both Redish's (2004) and Dezfouli's (2009) designs fail to capture relapse, relearning the addiction behaviour at a rate analogous to initial adoption after extinction. Conversely, the use of latent state representation in the dual process model enables the production of a spontaneous recovery upon re-exposure to the context in which the addiction is learned. This reflects experimental findings of human behaviour[19, 53, 66] and demonstrates that the project's model can capture a broader range of features than the previous designs.

Figure 5.10 exemplifies that state space is altered in response, showing when the reward is consistently provided the typical number of internal representations matches the world states in fig. 5.1. However, when extinction occurs the number of states increases due to the development of parallel state space. This results in the original addicted state space being maintained and enables it to be quickly reinstated. This overcomes the issue that TD learning generally, in which learning and unlearning are characterised by the same process, and thus it's not possible to create relapse with model-free solutions alone[78].

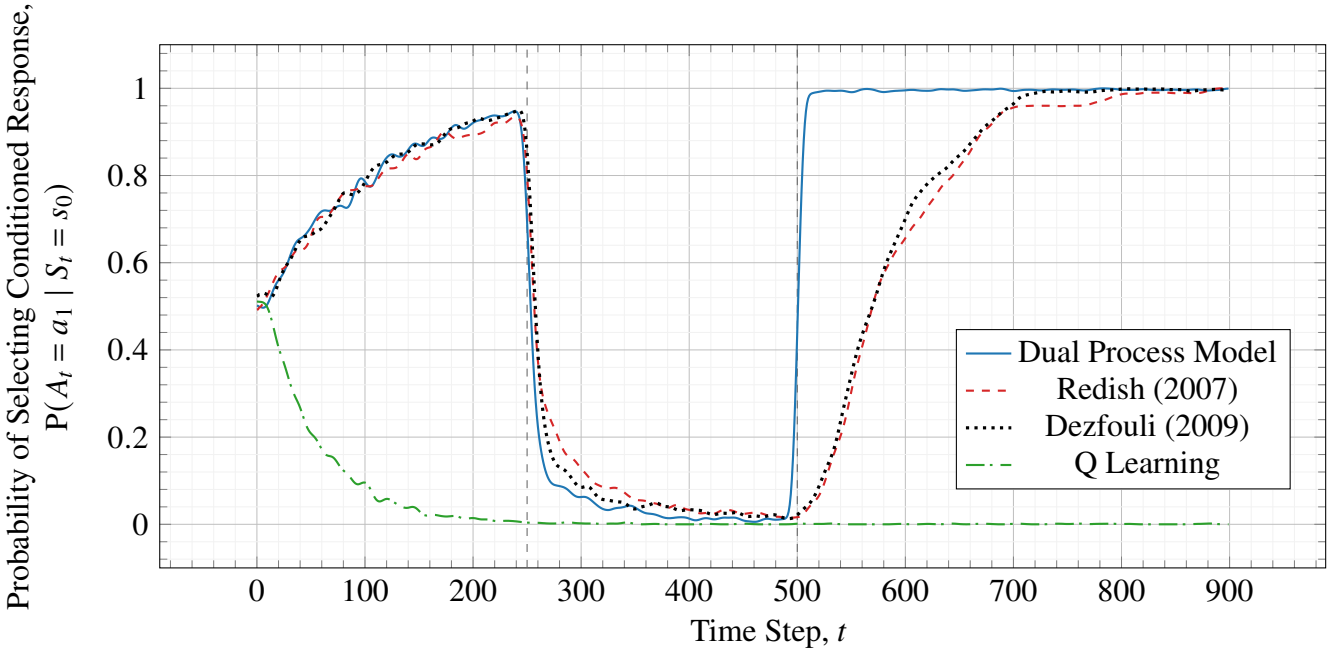

Figure 5.9: The dual process model shows relapse whereas pre-existing models cannot

It's notable that while forfig. 5.9 the reward dynamics return to the original values table 5.1 after extinction, this isn't necessary for relapse in the case of the dual process model. Figure 5.11 shows how cue-triggered relapse can occur even in an environment with an altered reward space, whereas neither Redish nor Dezfouli's models will produce relearning behaviour at all. This is because, for these designs, the reward for the addictive behaviour is now outweighed by that of natural reward, and the optimal policy is to choose this. This added element is an advantage of this project's model since despite there being experimental evidence for cue-triggered relapse, it's unlikely in the real world that an alteration in natural reward will change as the exposure occurs.

5.1.4 Behavioural addictions

The third design aim is related to behavioural addictions. These are becoming increasingly problematic with the rise of social media addiction and gambling addiction in society. Once again, these features cannot be inherently captured in the pre-existing Redish and Dezfouli models[14, 34]. These models focused purely on substance use disorders (SUDs), and therefore are limited by factors that directly stimulate the dopamine surge. Experimentally it's been shown that behavioural addiction doesn't produce the same magnitude of dopamine stimulation as drug behaviours, and therefore this experiment tested to see if the model of gambling addiction can be simulated without direct dopamine simulation. This simulation aims to test the influence of win streaks in the adoption of gambling, similar to what Redish did in his 2007 work[61].

The agent was placed in the environment shown in fig. 5.12, where possible actions and three states exist. The first action is a no operation a_{nop} action, in which the agent gains no reward $r_{\text{nop}} = 0$ and is kept in state s_init. The second is placing a bet $a$bet. With high probability P_{loss}, the bet results in the

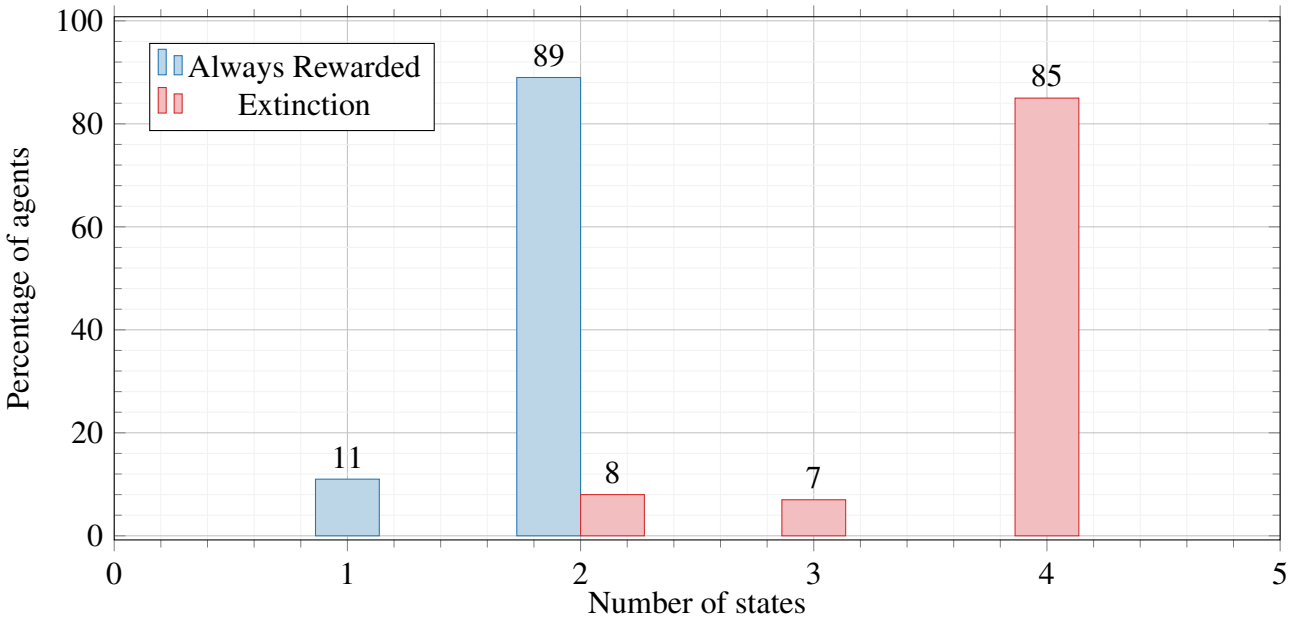

Figure 5.10: In the dual process model the number of internal states increases during extinction

agent moving to a loss state s_{loss}, and subsequently receiving a negative reward is given r_{loss}. With low probability P_{win} the agent wins the bet moving to a winning state s_{win} and then receiving a large reward r_{win}. The odds are set with high probability so that betting on average results in a loss, making a_{nop} the optimal policy.

The experiment design has several phases. First, the agent is trained with regular probabilities for 800 time steps. Then, the probability of winning is greatly increased, such that this becomes the probability. This is maintained for 100, 200, or 400 time steps. Following this, the probabilities are returned to their original values and the simulation runs until time step 2000.

At each timestep, whether or not the agent places a bet is recorded. A mean was taken over 100 trials within the betting environment to provide a probability of betting at each timestep.

The results are shown in fig. 5.13, illustrating how an increased win streak leads to a higher probability of compulsively gambling later on. This is explained again by splitting into the state spaces during an extended period of winning. This creates a state space to which the agent assigns wins and a parallel state to which losses are, even though the cue is the same. Therefore, the agent at each time step chooses to bet since it thinks it can gain the high reward, and each time it loses, it assigns this to the loss state, assuming its state prediction is wrong. This mimics the hindsight bias seen in problem gambling[68, 84], with the agent in effect forgetting its losses by classifying them separately from its victories.

The influence of win duration can be explained by the tightening of the covariance matrix. After a long win streak, the covariance matrix is tightened significantly, making the agent more likely to reclassify losses to a separate state. Then when it gets a win it then effectively relapses, with its expectation of victory being overestimated. Therefore, it bets at an increased rate in response to the

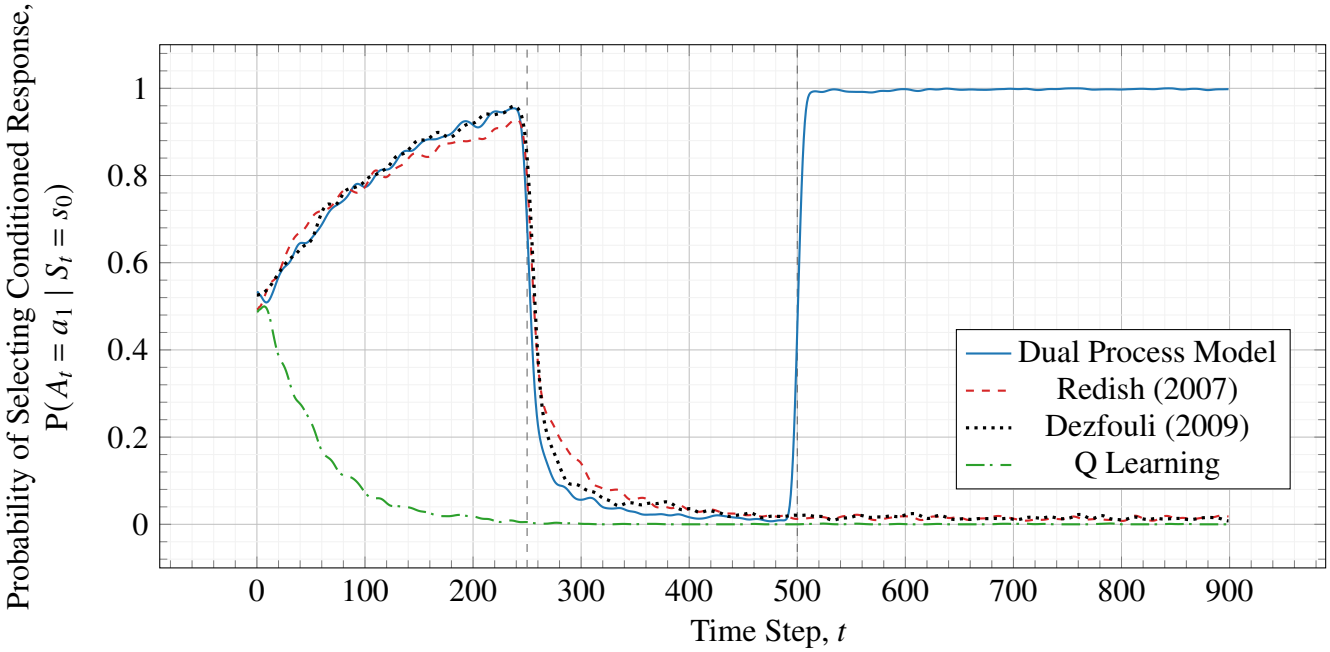

Figure 5.11: Dual process model can show relapse without lowering the rewards after extinction

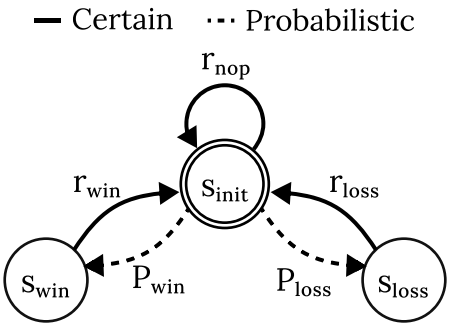

Figure 5.12: Environment dynamics for the gambling experiment

win despite having no probabilistic basis.

Furthermore, if multiple cues are used for wins and losses, it's possible for this design to create the near-miss effect[64], where those suffering from problem gambling are likely to associate cues similar to victories with an increased likelihood of winning.

Overall, this feature of the project's model makes it able to classify addictions that aren't caused by direct dopamine simulation. This is an advantage over the previous models that cannot do this, as shown by a control trial of the same experiment ran on the project's Dezfouli implementation for 400 trials (fig. 5.14). Since a significant objective of the project was to create a model that can capture both behavioural and pharmacological addictions, this is a significant strength of this project's dual process design over the pre-existing designs implemented.

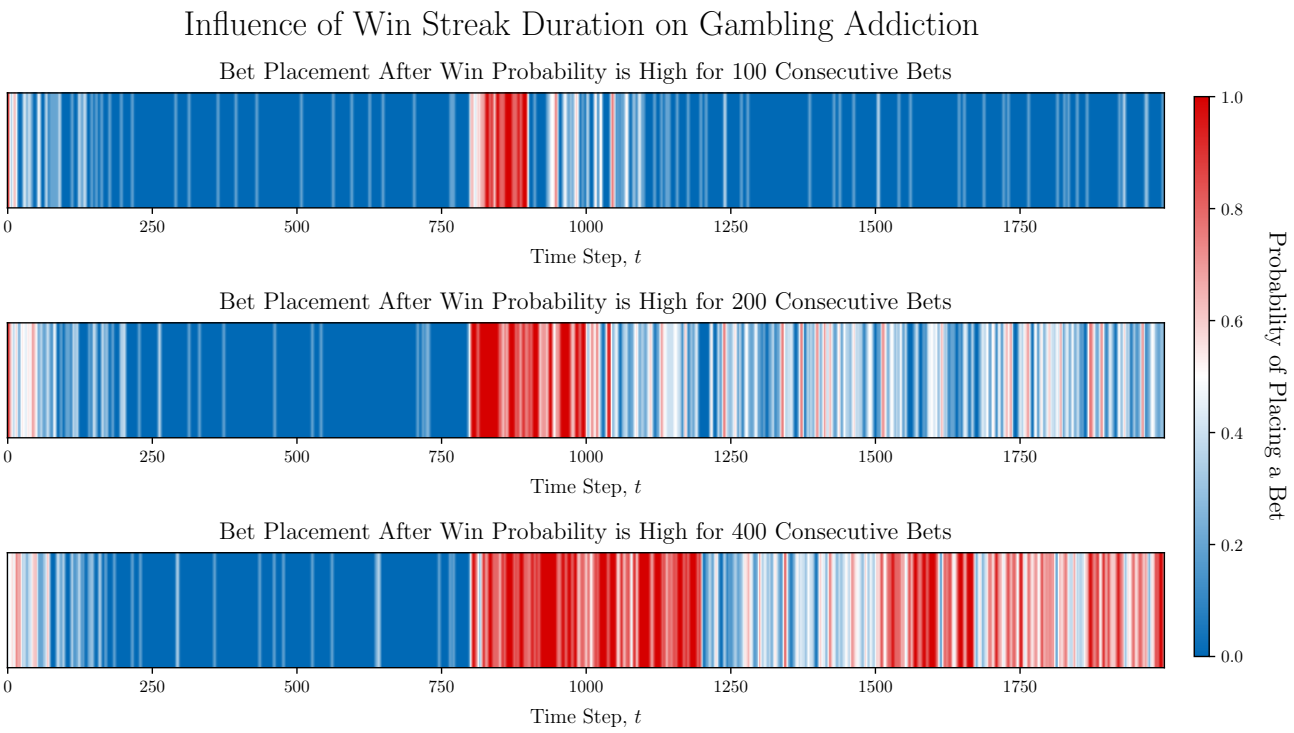

Figure 5.13: The duration of a winning streak increases the likelihood of gambling addiction

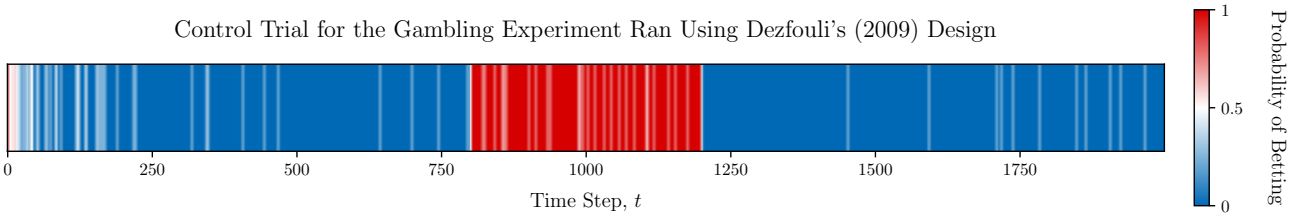

Figure 5.14: In the pre-existing models, win streak duration did not influence gambling addiction

5.1.5 Influence of Environmental and Genetic Factors

The final area of focus for the project was to look at the interacting genetic and environmental factors relating to addiction. This was done through the experiment based on the experiment outlined in section 5.1.2 using the environment in fig. 5.1 (b) ranging the natural reward produced by action a_0 in state s_0 from 7.5 to 12.5, and the dopamine surge $D(s)$ produced from 12.5 to 17.5 at a resolution of 0.1 in each axis. The results were plotted in a heatmap fig. 5.15, and it's clear that the disparity between natural reward and dopamine surge alters the chances of adopting addiction. This is the case despite addiction being suboptimal, and results from the maximisation relative to the dopamine surge as shown in eq. (3.11). This matches well-established psychological theory, showing that when people have higher quality outcomes, they are less likely to become addicted[27, 52, 69], and relates to why individual differences cause some to get addicted when others don't.

The model produced is reductionist with a linear relationship between the two factors which isn't realistic for humans. Nevertheless, this was necessary for the computational complexity of the model to be manageable, and it demonstrated the wider motivation behind the project and met the first

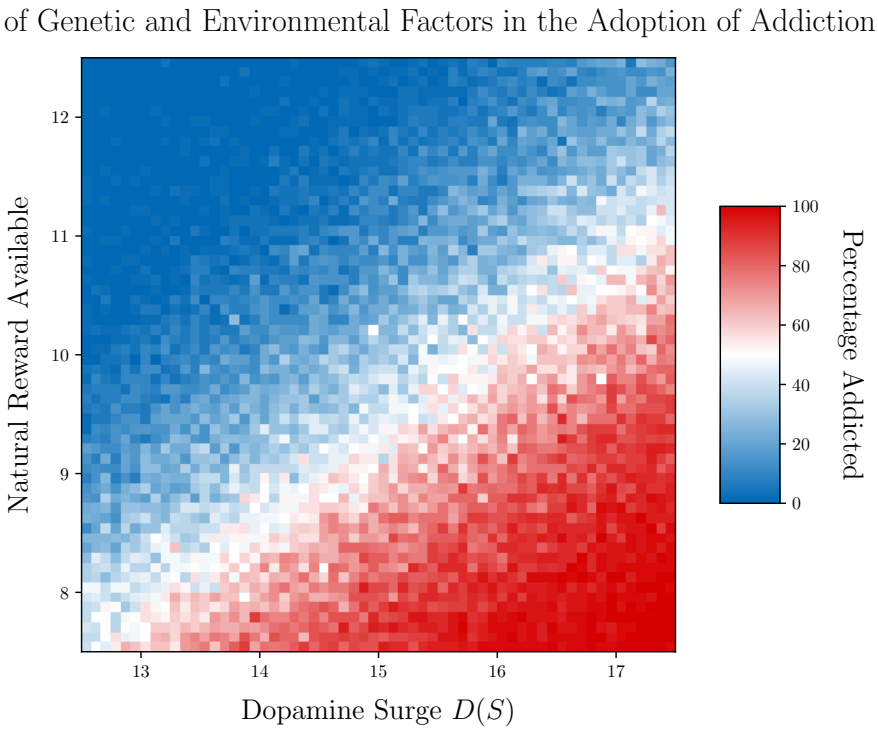

Figure 5.15: Influence of Genetic and Environmental Factors In Uptake of Drug Addiction

design objective. Production of data such as this to help identify risk factors in the population and provide information which could reduce the rates of addiction in society. Moreover, the influence of interventions and treatments could be applied on top of the models to identify those that most effectively mitigate against and treat addiction.

5.2 Evaluation of the design

5.2.1 Evaluation of Experiments Reproducing Pre-Existing Findings

Overall, this project's model could reproduce the key factors related to pharmacological addictions while introducing a state classification component. The experiments show that this project's model is consistent with both psychological theories surrounding dopamine altering the reward prediction error to result in the adoption of suboptimal addictive choices and existing implementations for simulating addiction. This satisfies the first of the design objectives and means the model could go some way to being used to predict addiction in humans, which is the project's overarching aim.

However, the inconsistency is disappointing, creating an additional limitation that had been resolved in the progression from Redish's to Dezfouli's model. This is particularly unfortunate given that it stems from increased design complexity, so it cannot be seen as a tradeoff. It is notable that blocking is less robust in humans than in animals, with association typically not being so binary which could be due to differences in the cognitive processes involved in learning about binary and continuous outcomes[58]. Nevertheless, forms of blocking do occur, which is a disadvantage of the project's

model.

5.2.2 Evaluation of Features Beyond the Pre-existing Designs

Nevertheless, the state splitting aspect preventing blocking from being simulated is the main feature of the additional behaviour this model can capture. It enables additional features of addiction to be captured, notably relapse. Since this is a characterising aspect of addiction, often being referred to as a chronic and relapsing conditon[39], this addition is a distinguishing feature of this project's model that helps achieve its overall aim of creating a novel RL algorithm building on previous designs.

Capturing behavioural addictions also helps in this regard, satisfying another of the design objectives and helping the model be a more holistic representation of addictive behaviour.

However, this additional complexity did come at a cost, and in particular, the implementation of the dual-process model was considerably less performant. For this application, this wasn't limiting since the timescales of simulations were always short, however for larger-scale testing this could be an issue. The heatmap testing environmental and genetic factors fig. 5.15 took 6 hours to run and was only over a limited region of interest. If a more exploratory analysis was undertaken, performance would be more crucial, and it would have to be judged if the additional features are necessary.

Finally, the behaviour still cannot account for the primary features of addiction such as increasing levels of tolerance or withdrawal. Alternate actor-critic RL agents are also capable of capturing both relapse and behavioural addictions, however, since these are model-based approaches they are unlikely to resolve this issue.

5.2.3 Fulfilment of Redish's (2008) Unified Framework for Addiction[62]

Redish 2008 proposed a set of criteria for understanding addiction as a process arising from multiple, interacting systems: a planning system, habit system, and situation recognition system. It suggests ten potential vulnerabilities drive people towards maladaptive behaviour(table 5.2).

Of these, the experiments above show that this project's model can comprehensively capture the misclassification of situations, overvaluation of actions, overfast discounting processes and changes in learning rates.

While it could be argued that the use of the basal limit, by moving the deviating natural reward level so the agent desires drug rewards to compensate can predict features of deviations from homeostasis and changes in allostatic set points, it doesn't capture withdrawal well. This is characterised by much more than a negative temporal difference error, with an interplay of physical and psychological symptoms occurring when a person stops using a substance to which they have become addicted, including cravings, anxiety, irritability, and insomnia.

The model is, therefore, capable of covering a range of features of addiction but is far from capturing all aspects. Overall this is not a huge problem of the design, however, with addiction being highly complex, with interacting neurological, social and cognitive components. The design complexity for

	Description	Key Systems	Clinical Consequence
1	Deviations from homeostasis	Planning	Withdrawal
2	Changes in allostatic set points	Planning	Changed physiological set points, craving
3	Mimicking reward	Planning	Incorrect action-selection, craving
4	Overvaluation in the planning system	Planning	Incorrect action-selection craving
5	Incorrect search of cue response behaviours	Planning	Obsession
6a	Misclassification of situations: overcategorisation	Situation recognition	Illusion of control, hindsight bias
6b	Misclassification of sitautions: overgeneralisation	Situation recognition	Perseveration in the face of losses
7	Overvaluation of actions	Habit	Automated, robotic drug-use
8	Selective inhibition of the planning system	System selection	Fast development of habit learning
9	Overfast discounting processes	Planning, habit	Impulsivity
10	Changes in learning rates	Planning, habit	Excess drug-related cue association

Table 5.2: Criteria for Addiction in Redish's (2008) unified model of addiction

creating a model that captures all features of addiction would be immense, and this project's model, while limited to certain areas, could have exploratory power and be used in treating a specific set of addictions. It's still more general than some previous models, such as the Dezfouli and Redish designs, and can consequently capture a broader range of behaviours.

5.2.4 Alternate Methods of Modelling Addiction

There have been non-RL approaches to model addiction, including previous research on mathematical, economic and neurological simulations. However, compared to these models, RL generally offers a more comprehensive understanding of addiction, integrating aspects of rational decision-making, non-conscious processes, and neural mechanisms. Nonetheless, each approach provides valuable perspectives that emphasise different aspects of addiction, so combined, it's possible to produce complementary rather than competing explanations from the different models.

Mathematical models of addiction typically use equations to describe the dynamics of drug use, tolerance, and dependence. These models offer quantitative predictions regarding the impact of different factors on drug consumption, but they tend to overlook the complexities of decision-making processes and individual differences. They also lack the dynamic learning aspect that RL captures, so cannot predict the interactive nature and continuous evolution of behaviour in response to changing environmental stimuli and internal states.

Economic models of addiction focus on the principles of rational choice, emphasising the role of

cost-benefit analysis and the impact of incentives. These models capture how individuals weigh the immediate rewards of drug use against potential long-term costs. The economic approach has been criticised for oversimplifying the complexity of addiction and downplaying the role of non-conscious processes. RL, alternatively, has the potential to incorporate both rational and non-conscious aspects of decision-making through dual process models like the one produced in this project, enabling RL to capture the multifaceted nature of addiction more effectively.

Neurobiological methods for modelling addiction are rooted in the understanding of the brain's structure and function, shedding light on the neurochemical and neural circuitry alterations induced by drug use. While RL shares many similarities with neurobiological models, particularly in terms of the dopamine-driven reward system, RL is a more general framework that can be applied to a wide range of addictive behaviours. Furthermore, RL models can be used to bridge the gap between neurobiological findings and behavioural patterns observed in addiction.

Overall, research in each of these areas can be combined with RL to provide a more comprehensive understanding of addiction and inform the development of more effective prevention and treatment strategies.

Bibliography

[1] S. H. Ahmed. Addiction as compulsive reward prediction. *Science*, 306(5703):1901–1902, 2004.

[2] G. Ainslie. Specious reward: a behavioral theory of impulsiveness and impulse control. *Psychological bulletin*, 82(4):463, 1975.

[3] D. American Psychiatric Association, A. P. Association, et al. *Diagnostic and statistical manual of mental disorders: DSM-5*, volume 5. American psychiatric association Washington, DC, 2013.

[4] T. E. Baker, Y. Zeighami, A. Dagher, and C. B. Holroyd. Smoking decisions: altered reinforcement learning signals induced by nicotine state. *Nicotine and Tobacco Research*, 22(2):164–171, 2020.

[5] R. E. Bellman. *Dynamic programming*. Princeton university press, 2010.

[6] M. A. Bornovalova, S. B. Daughters, G. D. Hernandez, J. B. Richards, and C. Lejuez. Differences in impulsivity and risk-taking propensity between primary users of crack cocaine and primary users of heroin in a residential substance-use program. *Experimental and clinical psychopharmacology*, 13(4):311, 2005.

[7] K. T. Brady, M. L. Verduin, and B. K. Tolliver. Treatment of patients comorbid for addiction and other psychiatric disorders. *Current psychiatry reports*, 9(5):374–380, 2007.

[8] T. H. Brandon, J. I. Vidrine, and E. B. Litvin. Relapse and relapse prevention. *Annu. Rev. Clin. Psychol.*, 3:257–284, 2007.

[9] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym. arxiv. *arXiv preprint arXiv:1606.01540*, 10, 2016.

[10] C. Dackis and C. O'Brien. Neurobiology of addiction: treatment and public policy ramifications. *Nature neuroscience*, 8(11):1431–1436, 2005.

[11] N. D. Daw. *Reinforcement learning models of the dopamine system and their behavioral implications*. Carnegie Mellon University, 2003.

[12] P. Dayan. Dopamine, reinforcement learning, and addiction. *Pharmacopsychiatry*, 42(S 01):S56–S65, 2009.

[13] P. Dayan and K. C. Berridge. Model-based and model-free pavlovian reward learning: revaluation, revision, and revelation. *Cognitive, Affective, & Behavioral Neuroscience*, 14:473–492, 2014.

[14] A. Dezfouli, P. Piray, M. M. Keramati, H. Ekhtiari, C. Lucas, and A. Mokri. A neurocomputational model for cocaine addiction. *Neural computation*, 21(10):2869–2893, 2009.

[15] M. Feltenstein and R. See. The neurocircuitry of addiction: an overview. *British journal of pharmacology*, 154(2):261–274, 2008.

[16] C. D. Fiorillo, P. N. Tobler, and W. Schultz. Discrete coding of reward probability and uncertainty by dopamine neurons. *Science*, 299(5614):1898–1902, 2003.

[17] M. T. French, I. Popovici, and L. Tapsell. The economic costs of substance abuse treatment: Updated estimates and cost bands for program assessment and reimbursement. *Journal of substance abuse treatment*, 35(4):462–469, 2008.

[18] H. Garavan, J. Pankiewicz, A. Bloom, J.-K. Cho, L. Sperry, T. J. Ross, B. J. Salmeron, R. Risinger, D. Kelley, and E. A. Stein. Cue-induced cocaine craving: neuroanatomical specificity for drug users and drug stimuli. *American journal of psychiatry*, 157(11):1789–1798, 2000.

[19] E. L. Gardner. Addiction and brain reward and antireward pathways. *Chronic Pain and Addiction*, 30:22–60, 2011.

[20] S. J. Gershman and Y. Niv. Exploring a latent cause theory of classical conditioning. *Learning & behavior*, 40:255–268, 2012.

[21] R. Z. Goldstein, N. Alia-Klein, D. Tomasi, L. Zhang, L. A. Cottone, T. Maloney, F. Telang, E. C. Caparelli, L. Chang, T. Ernst, et al. Is decreased prefrontal cortical sensitivity to monetary reward associated with impaired motivation and self-control in cocaine addiction? *American Journal of Psychiatry*, 164(1):43–51, 2007.

[22] R. Z. Goldstein and N. D. Volkow. Drug addiction and its underlying neurobiological basis: neuroimaging evidence for the involvement of the frontal cortex. *American Journal of Psychiatry*, 159(10):1642–1652, 2002.

[23] J. E. Grant and S. R. Chamberlain. Expanding the definition of addiction: DSM-5 vs. ICD-11. *CNS spectrums*, 21(4):300–303, 2016.

[24] A. M. Graybiel. Habits, rituals, and the evaluative brain. *Annu. Rev. Neurosci.*, 31:359–387, 2008.

[25] L. Green and J. Myerson. A discounting framework for choice with delayed and probabilistic rewards. *Psychological bulletin*, 130(5):769, 2004.

[26] S. M. Groman, B. Massi, S. R. Mathias, D. Lee, and J. R. Taylor. Model-free and model-based influences in addiction-related behaviors. *Biological psychiatry*, 85(11):936–945, 2019.

[27] P. F. Hadaway, B. K. Alexander, R. B. Coambs, and B. Beyerstein. The effect of housing and gender on preference for morphine-sucrose solutions in rats. *Psychopharmacology*, 66:87–91, 1979.

[28] S. T. Higgins, S. H. Heil, R. Dantona, R. Donham, M. Matthews, and G. J. Badger. Effects

of varying the monetary value of voucher-based incentives on abstinence achieved during and following treatment among cocaine-dependent outpatients. *Addiction*, 102(2):271–281, 2007.

[29] Y. Hou, D. Xiong, T. Jiang, L. Song, and Q. Wang. Social media addiction: Its impact, mediation, and intervention. *Cyberpsychology: Journal of psychosocial research on cyberspace*, 13(1), 2019.

[30] C. L. Hull. The conflicting psychologies of learning—a way out. *Psychological Review*, 42(6):491, 1935.

[31] P. W. Kalivas and C. O'Brien. Drug addiction as a pathology of staged neuroplasticity. *Neuropsychopharmacology*, 33(1):166–180, 2008.

[32] P. W. Kalivas and N. D. Volkow. The neural basis of addiction: a pathology of motivation and choice. *American Journal of Psychiatry*, 162(8):1403–1413, 2005.

[33] L. J. Kamin. Predictability, surprise, attention, and conditioning. In *Symp. on Punishment*, number TR-13, 1967.

[34] A. Kato, K. Shimomura, D. Ognibene, M. A. Parvaz, L. A. Berner, K. Morita, and V. G. Fiore. Computational models of behavioral addictions: state of the art and future directions. *Addictive Behaviors*, page 107595, 2022.

[35] M. Keramati, A. Dezfouli, and P. Piray. Understanding addiction as a pathological state of multiple decision making processes: a neurocomputational perspective. *Computational neuroscience of drug addiction*, pages 205–233, 2012.

[36] M. Keramati, A. Durand, P. Girardeau, B. Gutkin, and S. H. Ahmed. Cocaine addiction as a homeostatic reinforcement learning disorder. *Psychological Review*, 124(2):130, 2017.

[37] R. M. Kitchin. Cognitive maps: What are they and why study them? *Journal of environmental psychology*, 14(1):1–19, 1994.

[38] J. Kober, J. A. Bagnell, and J. Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[39] G. F. Koob. Neurobiology of addiction. *Focus*, 9(1):55–65, 2011.

[40] G. F. Koob and M. Le Moal. Plasticity of reward neurocircuitry and the'dark side'of drug addiction. *Nature neuroscience*, 8(11):1442–1444, 2005.

[41] G. F. Koob and N. D. Volkow. Neurobiology of addiction: a neurocircuitry analysis. *The Lancet Psychiatry*, 3(8):760–773, 2016.

[42] L. Lander, J. Howsare, and M. Byrne. The impact of substance use disorders on families and children: from theory to practice. *Social work in public health*, 28(3-4):194–205, 2013.

[43] S. W. Lee, S. Shimojo, and J. P. O'Doherty. Neural computations underlying arbitration between model-based and model-free learning. *Neuron*, 81(3):687–699, 2014.

[44] T. V. Lim and K. D. Ersche. Theory-driven computational models of drug addiction in humans: fruitful or futile? *Addiction Neuroscience*, page 100066, 2023.

[45] C. Lüscher and R. C. Malenka. Drug-evoked synaptic plasticity in addiction: from molecular changes to circuit remodeling. *Neuron*, 69(4):650–663, 2011.

[46] N. J. Mackintosh. A theory of attention: Variations in the associability of stimuli with reinforcement. *Psychological review*, 82(4):276, 1975.

[47] S. Mahadevan. Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Recent advances in reinforcement Learning*, pages 159–195, 1996.

[48] K. K. Mak, K. Lee, and C. Park. Applications of machine learning in addiction studies: A systematic review. *Psychiatry research*, 275:53–60, 2019.

[49] S. M. McClure, N. D. Daw, and P. R. Montague. A computational substrate for incentive salience. *Trends in neurosciences*, 26(8):423–428, 2003.

[50] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[51] P. R. Montague, P. Dayan, and T. J. Sejnowski. A framework for mesencephalic dopamine systems based on predictive hebbian learning. *Journal of neuroscience*, 16(5):1936–1947, 1996.

[52] J. Nader, C. Claudia, R. El Rawas, L. Favot, M. Jaber, N. Thiriet, and M. Solinas. Erratum: Loss of environmental enrichment increases vulnerability to cocaine addiction. *Neuropsychopharmacology*, 39(3):780–780, 2014.

[53] R. S. Niaura, D. J. Rohsenow, J. A. Binkoff, P. M. Monti, M. Pedraza, and D. B. Abrams. Relevance of cue reactivity to understanding alcohol and smoking relapse. *Journal of abnormal psychology*, 97(2):133, 1988.

[54] D. Ognibene, V. G. Fiore, and X. Gu. Addiction beyond pharmacological effects: The role of environment complexity and bounded rationality. *Neural Networks*, 116:269–278, 2019.

[55] T. A. Paine, H. C. Dringenberg, and M. C. Olmstead. Effects of chronic cocaine on impulsivity: relation to cortical serotonin mechanisms. *Behavioural brain research*, 147(1-2):135–147, 2003.

[56] P. I. Pavlov. Conditioned reflexes: an investigation of the physiological activity of the cerebral cortex. *Annals of neurosciences*, 17(3):136, 2010.

[57] J. M. Pearce and G. Hall. A model for pavlovian learning: variations in the effectiveness of conditioned but not of unconditioned stimuli. *Psychological review*, 87(6):532, 1980.

[58] P. F. Peter F, S.-L. Been, C. J. Mitchell, M. E. Bouton, and R. Frohardt. Forward and backward blocking of causal judgment is enhanced by additivity of effect magnitude. *Memory & Cognition*, 31:133–142, 2003.

[59] W. W. Pettine, D. V. Raman, A. D. Redish, and J. D. Murray. Human generalization of internal representations through prototype learning with goal-directed attention. *Nature Human Behaviour*, pages 1–22, 2023.

[60] P. Piray, M. M. Keramati, A. Dezfouli, C. Lucas, and A. Mokri. Individual differences in nucleus accumbens dopamine receptors predict development of addiction-like behavior: a computational approach. *Neural computation*, 22(9):2334–2368, 2010.

[61] A. D. Redish. Addiction as a computational process gone awry. *Science*, 306(5703):1944–1947, 2004.

[62] A. D. Redish, S. Jensen, and A. Johnson. Addiction as vulnerabilities in the decision process. *Behavioral and brain sciences*, 31(4):461–487, 2008.

[63] A. D. Redish, S. Jensen, A. Johnson, and Z. Kurth-Nelson. Reconciling reinforcement learning models with behavioral extinction and renewal: implications for addiction, relapse, and problem gambling. *Psychological review*, 114(3):784, 2007.

[64] R. Reid. The psychology of the near miss. *Journal of gambling behavior*, 2(1):32–39, 1986.

[65] R. A. Rescorla. A theory of pavlovian conditioning: Variations in the effectiveness of reinforcement and non-reinforcement. *Classical conditioning, Current research and theory*, 2:64–69, 1972.

[66] R. A. Rescorla. Spontaneous recovery. *Learning & Memory*, 11(5):501–509, 2004.

[67] M. C. Ritz, R. Lamb, S. R. Goldberg, and M. J. Kuhar. Cocaine receptors on dopamine transporters are related to self-administration of cocaine. *Science*, 237(4819):1219–1223, 1987.

[68] N. J. Roese and K. D. Vohs. Hindsight bias. *Perspectives on psychological science*, 7(5):411–426, 2012.

[69] M. E. Roth and M. E. Carroll. Sex differences in the escalation of intravenous cocaine intake following long-or short-access to cocaine self-administration. *Pharmacology Biochemistry and Behavior*, 78(2):199–207, 2004.

[70] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo method*. John Wiley & Sons, 2016.

[71] W. Schultz, P. Dayan, and P. R. Montague. A neural substrate of prediction and reward. *Science*, 275(5306):1593–1599, 1997.

[72] C. E. Shannon. Xxii. programming a computer for playing chess. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 41(314):256–275, 1950.

[73] K. Shimomura, A. Kato, and K. Morita. Rigid reduced successor representation as a potential mechanism for addiction. *European Journal of Neuroscience*, 53(11):3768–3790, 2021.

[74] D. A. Simon and N. D. Daw. Dual-system learning models and drugs of abuse. *Computational neuroscience of drug addiction*, pages 145–161, 2012.

[75] B. F. Skinner. *The behavior of organisms: An experimental analysis*. BF Skinner Foundation, 1938.

[76] B. F. Skinner. *Science and human behavior*. Number 92904. Simon and Schuster, 1965.

[77] J. L. Sorensen and A. L. Copeland. Drug abuse treatment as an HIV prevention strategy: a review. *Drug and alcohol dependence*, 59(1):17–31, 2000.

[78] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[79] E. L. Thorndike. Animal intelligence: An experimental study of the associative processes in animals. *The Psychological Review: Monograph Supplements*, 2(4):i, 1898.

[80] E. L. Thorndike. *The elements of psychology*. AG Seiler, 1905.

[81] E. L. Thorndike. *Animal intelligence: Experimental studies*. Transaction Publishers, 1911.

[82] P. N. Tobler, J. P. O'Doherty, R. J. Dolan, and W. Schultz. Human neural learning depends on reward prediction errors in the blocking paradigm. *Journal of Neurophysiology*, 95(1):301–310, 2006.

[83] E. C. Tolman. Cognitive maps in rats and men. *Psychological review*, 55(4):189, 1948.

[84] T. Toneatto. Cognitive psychopathology of problem gambling. *Substance use & misuse*, 34(11):1593–1604, 1999.

[85] UNODC. Booklet 2: Global overview of drug demand and drug supply. *In World Drug Report 2022*, 2022.

[86] H. van Hasselt, D. Borsa, and M. Hessel. Reinforcement learning lecture series 2021. *DeepMind x UCL*, 2021.

[87] J. M. Wang, L. Zhu, V. M. Brown, R. De La Garza II, T. Newton, B. King-Casas, and P. H. Chiu. In cocaine dependence, neural prediction errors during loss avoidance are increased with cocaine deprivation and predict drug use. *Biological psychiatry: cognitive neuroscience and neuroimaging*, 4(3):291–299, 2019.

[88] C. J. C. H. Watkins. Learning from delayed rewards. 1989.

[89] P. Wegner. Concepts and paradigms of object-oriented programming. *ACM Sigplan Oops Messenger*, 1(1):7–87, 1990.