



BIROn - Birkbeck Institutional Research Online

Enabling Open Access to Birkbeck's Research Degree output

Understanding efficient reinforcement learning in humans and machines

<https://eprints.bbk.ac.uk/id/eprint/46202/>

Version: Full Version

Citation: Blakeman, Sam (2021) Understanding efficient reinforcement learning in humans and machines. [Thesis] (Unpublished)

© 2020 The Author(s)

All material available through BIROn is protected by intellectual property law, including copyright law.

Any use made of the contents should comply with the relevant law.

[Deposit Guide](#)
Contact: [email](#)

Understanding Efficient Reinforcement Learning in Humans and Machines

Sam Blakeman

Centre for Brain and Cognitive Development
Department of Psychological Sciences
Birkbeck, University of London

A thesis submitted for the degree of
Doctor of Philosophy

Submitted October 2020

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specifically indicated in the text.

Sam Blakeman
21st October 2020

Abstract

One of the primary mechanisms thought to underlie action selection in the brain is Reinforcement Learning (RL). Recently, the use of Deep Neural Networks in models of RL (Deep RL) has led to human-level performance on complex reward-driven perceptual-motor tasks. However, Deep RL is persistently criticised for being data inefficient compared to human learning because it lacks the ability to: (1) rapidly learn from new information and (2) transfer knowledge from past experiences. The purpose of this thesis is to form an analogy between the brain and Deep RL to understand how the brain performs these two processes.

To investigate the internal computations supporting rapid learning and transfer we use Complementary Learning Systems (CLS) theory. This allows us to focus on the computational properties of key learning systems in the brain and their interactions. We review recent advances in Deep RL and how they relate to the CLS framework. This results in the presentation of two novel Deep RL algorithms, which highlight key properties of the brain that support rapid learning and transfer: the fast learning of pattern-separated representations in the hippocampus, and the selective attention mechanisms of the pre-frontal cortex.

External factors in the environment can also impact upon rapid learning and transfer in the brain. We therefore conduct behavioural experiments that investigate how the degree of perceptual similarity between consecutive experiences affects people's ability to perform transfer. To do this we use naturalistic 2D video games that vary in perceptual features but rely on the same underlying rules. We discuss the results of these experiments with respect to Deep RL, analogical reasoning and category learning. We hope that the analogy formed over the course of this thesis between the brain and Deep RL can inform future research into efficient RL in humans and machines.

Acknowledgements

First and foremost I would like to thank my supervisor Denis for all of his guidance and support during this journey. It was thanks to your encouragement to attend CogSci 2017 that I entered the wonderful world of cognitive science. You gave me the confidence to pursue my true passion of using machine learning as a framework for understanding the mind. You have truly inspired me to carry on my research journey and for that I will forever be grateful. Thank you for being so generous with your time and for always encouraging open-ended discussions about cognition and computation. You allowed me explore my own ideas, even when there were perhaps too many, but made sure I never went too far down the rabbit-hole. I could not have asked for a better mentor and I hope to one day emulate the type of supervision you have shown me. When I look back at my PhD I think most of our weekly meetings and the joy they gave me.

I would also like to thank Rick for allowing me to attend his lab meetings and for all of the useful discussions we have had over the years. As with Denis, you have always been open to discussing ideas and I thank you for being so approachable and kind. You were always willing to look over my work and provide another perspective outside of my own little research bubble. You should be very proud of the cognition and computation family that you and Denis have fostered, and I wish it every success in the future.

While not part of my PhD, I would like to take the chance to thank Yann for being my first mentor in the scientific world. You helped me realise what I wanted from my career and gave me invaluable personal insights into life as an academic. This journey can be traced back to your honest advice and kindness that you showed me all those years ago in New Haven.

Many of my working days were spent in the British Medical Association office

and I would like to thank all of my colleagues in the office who made it such an enjoyable experience. Andrea, Georgie, Wikus, Lizzie, Laura, Kathryn, Emily and Matt thank you for always being there to snap me out of work and for your great friendships. A special thanks goes to Georgie for being my science ‘mum’ and Andrea and Matt for those times when a beer was much needed.

Outside of work, I have to thank Peter for always providing a level-head and for giving me perspective on life as a whole. Thanks to you I navigated PhD life in a sustainable way and learnt more about myself than I thought possible. Thank you Toby for all the snooker games you gave me over the years, they were always a delight when times were tough. Thank you Dean for being like a brother to me and for reminding me that life as a student really is not that bad. Finally, I would like to thank Josh for being the best friend I could ever hope for. No matter how long I disappeared down the PhD rabbit hole for, you would always be there as if nothing had changed and I will always treasure that.

To my parents, to put it simply this thesis is a tribute to you. The support and guidance you have given me throughout my life has led me to this point and for that I will be eternally grateful. My love of science has stemmed from you and I cannot thank you enough for giving me the ability to pursue my passions without limits. Most importantly, the lessons you have taught me extend far beyond this thesis by helping to guide me as a person. I hope you realise what wonderful parents you are.

Finally, I would like to thank Rosa for being my shining light in the final year of this journey. This year has been full of surprises but you have been my constant and your belief in me has kept me going. Your levels of understanding, compassion and love inspire me to be better. I look forward to what the future has in store for us.

Contents

1	Introduction	10
1.1	Efficient Learning Relies Upon Rapid learning and Transfer	11
1.2	Behavioural Evidence of Rapid Learning and Transfer in Humans . . .	12
1.2.1	Children and Development	12
1.2.2	Adult Behaviour	17
1.3	Reinforcement Learning as a Basic Mechanism for Learning	20
1.3.1	Operant and Classical Conditioning	20
1.3.2	AI and Reinforcement Learning	21
1.3.3	Reinforcement Learning in the Brain	22
1.4	Into the 21st century: Deep Reinforcement Learning	23
1.4.1	Deep learning	24
1.4.2	Deep Reinforcement Learning	26
1.5	The Problem of Efficiency in Deep Reinforcement Learning	27
1.6	Outline of the Thesis	29
2	Background	32
2.1	Reinforcement Learning	32
2.1.1	The Reinforcement Learning Problem	32
2.1.1.1	Markov Decision Processes	34
2.1.1.2	Value Functions and Policies	36
2.1.2	Solution Methods	38
2.1.2.1	Optimal Policies	38
2.1.2.2	Value-Based Solution Methods	39
2.1.2.3	Temporal Difference Learning and Q-Learning	41
2.1.2.4	Policy-Based Methods	44

2.1.2.5	Monte-Carlo Policy Gradient	44
2.1.2.6	Actor-Critic Methods	47
2.1.2.7	Model-Based Methods	48
2.2	Deep Learning	50
2.2.1	Basic Principles of Neural Networks	50
2.2.2	Learning in Deep Neural Networks	52
2.2.2.1	Objective Functions	52
2.2.2.2	Backpropagation	54
2.2.3	Deep Learning Architectures	58
2.2.3.1	Deep Convolutional Neural Networks	58
2.2.3.2	Long Short-Term Memory Networks	61
2.2.3.3	Autoencoders	64
2.3	Deep Reinforcement Learning	65
2.3.1	Deep Q-Learning	66
2.3.2	Advantage Actor-Critic (A2C)	69
2.4	Summary	70
3	CLS Theory as the Basis for Efficient RL	71
3.1	CLS Theory and Deep RL	72
3.2	1. Connections Between the Neocortex and Striatum	74
3.2.1	Catastrophic Forgetting	75
3.2.2	Objective Functions	76
3.2.3	Disentangled Representations	79
3.3	2. Connections Between the Hippocampus and Striatum	84
3.3.1	Fast Learning in the Hippocampus Supports Efficient Rein- forcement Learning	85
3.3.2	Recurrent Similarity Computation	87
3.3.3	Relational Representations and Cognitive Maps	89
3.3.4	Model-Based Reinforcement Learning and the Successor Rep- resentation	94
3.4	3. Connections Between the Neocortex and Hippocampus	98
3.4.1	Re-play	98
3.4.2	Pre-play	100

3.5	Conclusions	103
4	Complementary Temporal Difference Learning	106
4.1	Introduction	107
4.2	Methods	109
4.2.1	Complementary Temporal Difference Learning (CTDL)	109
4.2.2	Simulated Environments	112
4.2.2.1	Grid World Task	112
4.2.2.2	Cart-Pole	112
4.2.2.3	Continuous Mountain Car	113
4.3	Results	113
4.4	Neural Underpinnings	125
4.5	Discussion	130
4.6	Conclusions	135
5	Extending CLS Theory to Include Pre-Frontal Cortex	138
5.1	Pre-Frontal Cortex as an Additional Learning System	139
5.2	4. Connections Between the Pre-Frontal Cortex and Striatum	140
5.2.1	Meta-Reinforcement Learning	141
5.3	5. Connections Between the Pre-Frontal Cortex and Hippocampus	145
5.3.1	Memory Recall	145
5.3.2	Concept Formation	148
5.4	6. Connections Between Pre-Frontal Cortex and Sensory Cortex	152
5.4.1	Selective Attention	153
5.5	Conclusions	163
6	Selective Particle Attention	166
6.1	Introduction	167
6.2	Methods	172
6.2.1	Tasks	172
6.2.1.1	Multiple Choice Task	172
6.2.1.2	Object Collection Game	173
6.2.2	Selective Particle Attention	174
6.2.2.1	VGG-16	175

6.2.2.2	Attention Layer	175
6.2.2.3	Particle Filter	176
6.2.2.4	Deep Reinforcement Learning Algorithm	180
6.3	Results	186
6.3.1	Multiple Choice Task	186
6.3.2	Object Collection Game	190
6.4	Discussion	192
6.5	Concluding Remarks	198
7	The Effect of Perceptual Similarity on Transfer in Humans	201
7.1	Introduction	202
7.1.1	Analogy: The Relational Shift and Progressive Alignment	202
7.1.2	Concept Learning and Category Structure	204
7.1.3	Insights From Deep Reinforcement Learning Approaches	206
7.1.4	Overview of Human Experiments	210
7.2	Experiment 1	212
7.2.1	Methods	212
7.2.2	Results	218
7.2.3	Discussion	227
7.3	Experiment 2	228
7.3.1	Methods	229
7.3.2	Results	237
7.3.3	Discussion	247
7.4	Experiment 3	250
7.4.1	Methods	251
7.4.2	Results	251
7.4.2.1	Object Interactions	257
7.4.3	Discussion	260
7.5	General Discussion	262
8	Discussion	271
8.1	Goal of The Thesis	271
8.2	Summary of Computational Findings	272

8.3	Summary of Empirical Findings	273
8.4	Limitations and Future Work	275
8.4.1	Transfer vs. Rapid Learning	275
8.4.2	Efficient Reinforcement Learning as a Combination of Mechanisms	278
8.4.3	Further Demarcation of Learning Systems in the Brain	280
8.4.4	Issues With Comparing Deep Reinforcement Learning to Human Learning	282
8.4.5	The Notion of Perceptual Similarity	283
8.4.6	Learning in Naturalistic Tasks	284
8.5	Lessons For The Future	286
8.5.1	Starting With Simple Problems	287
8.5.2	The Intersection between Artificial Intelligence and Cognitive Science	289
8.5.3	The Importance of Developmental Studies	291
8.5.4	Proposing a New Experimental Paradigm	293
8.6	Concluding Remarks	294
9	Bibliography	298
	Appendices	328
A	Supplementary Data for Experiment 1	328
B	Supplementary Data for Experiment 2	331
C	Experiment 2.5	336
C.1	Methods	336
C.2	Results	338
C.3	Discussion	347
D	Supplementary Data for Experiment 3	350

Chapter 1

Introduction

Overview

This chapter outlines the general goal of the thesis and sets the stage for subsequent chapters. We start by proposing that efficient Reinforcement Learning (RL) in the brain is supported by two key processes: rapid learning and the transfer of past knowledge (Section 1.1). We then review several behavioural examples of how humans are able to use these two processes to learn which actions to take in a new situation based on limited feedback (Section 1.2). This feedback is typically in the form of sparse rewards and is thought to engage the brain's RL machinery. We therefore follow these examples with a brief review of RL from both an Artificial Intelligence (AI) and a cognitive science perspective (Section 1.3). In particular, Deep RL has emerged as a promising candidate for exploring how the brain transforms high-dimensional perceptual input into actions based on reward. We therefore discuss recent advancements in Deep RL (Section 1.4) and highlight how it lacks the efficiency displayed by human RL in new situations (Section 1.5). This chapter concludes with a general outline of how we plan to use an analogy between Deep RL and the brain, along with a combination of computational and empirical approaches, to understand the processes that support efficient RL (Section 1.6).

1.1 Efficient Learning Relies Upon Rapid learning and Transfer

Throughout our daily lives we are often presented with situations we have never experienced before. From an evolutionary stand-point, slow learners are naturally at a disadvantage in new situations because they are more likely to repeat detrimental actions. Our ability to quickly identify the best actions in a new situation depends on two key process: (1) the ability to rapidly learn from information provided by the new situation and (2) the ability to transfer past knowledge to the new situation. These two processes are fundamental to human cognition and interact with each other to help guide us through the world when data is limited or expensive. Indeed, it is unlikely that any two experiences in our lifetime are truly the same and so we are constantly recruiting these two processes to some degree. As the philosopher Heraclitus once said, “No man ever steps in the same river twice, for it’s not the same river and he’s not the same man” (Robinson, 1987). It is for these reasons that life is often framed as a continual learning process whereby past experiences constantly interact with the learning of new information.

When it comes to investigating these two processes, it is often difficult to distinguish between them as they operate in a reciprocal relationship. On the one hand, learning in a new situation can be influenced by our past knowledge, and on the other, the past knowledge that we transfer can be influenced by learning in the new situation. The one exemption to this is the case of ‘zero-shot’ transfer whereby people use past experiences to infer the best action in a new situation without using any feedback. No learning occurs in the new situation and so action selection is made purely based on transfer and the current perceptual input. From an experimental point of view, transfer can make it hard to compare humans to computational models because people do not start tasks *tabula rasa* and so models have to be imbued with similar prior knowledge. Thus, the goal of this thesis is to try to understand the internal computations underlying rapid learning and transfer and how external factors may affect them.

1.2 Behavioural Evidence of Rapid Learning and Transfer in Humans

Before exploring how the brain might perform rapid learning and transfer, we will first explore some canonical examples of these processes and demonstrate their vital role in human cognition. These processes occur early on in development and stay with us throughout adult life. As we shall see, this means that a range of examples exist both in children and adults.

1.2.1 Children and Development

Children often represent good subjects for studying rapid learning and transfer because they are still in the process of learning relatively simple skills and concepts that are amenable to experimental manipulation. In comparison, adults enter experiments with a wealth of prior knowledge that can be readily transferred to any given task, which can be difficult to control for. From a developmental perspective the ability to learn from very few experiences and infer optimal actions rapidly in new situations is critical for making sense of the world, obtaining reward and avoiding danger. Imagine a child that could not quickly acquire the concept of object permanence and transfer it to new situations. This would make playing a simple game such as hide and seek or retrieving a biscuit from a jar highly challenging.

One of the fundamental challenges facing young children is learning language. Interestingly, it has been repeatedly shown that children can quickly learn new words after very few examples (Carey and Bartlett, 1978). These fast associations are crucial as the human vocabulary can be large and so the presentation of repeated labelled examples is unlikely. Not only can children learn word associations quickly, but they can also apply newly acquired words to novel exemplars that they have never seen before (Brown, 1957; Waxman, 1998; Waxman and Booth, 2000; Childers and Tomasello, 2003). This is important because without the ability to transfer meaning, the utility of a word would be drastically reduced and further learning would be slow. For example, when a child first learns the word ‘dog’, they may only be given a few word-image pairings and they need to be able to quickly acquire its meaning and transfer it to a wide-array of other breeds or situational contexts.

This in turn helps to bootstrap the learning of further knowledge.

This rapid and transferable learning is not limited to just language. For example, it has also been demonstrated when learning how to use tools. In a study by Casler and Kelemen (2005), they presented children with an array of appropriate novel tools that could be used to obtain a desired goal. An adult then demonstrated how one of the tools could be used to obtain the goal. As a result, the children consistently used the same tool after just a single demonstration by the adult. This demonstrates how children can rapidly use a single observation to associate a tool with its ability to acquire a goal.

Children learning to use tools do not only demonstrate highly efficient learning but also the ability to perform transfer without the need for feedback. In a series of experiments, Brown (1990) provided children aged 24-42 months with a set of 'tools' that could be used to retrieve a toy that was beyond their reach (Set 1 in Figure 1.1). Some of these tools, such as a long rake and a long hook, were sufficient to pull the toy towards them as they were both long, rigid and had a useful end. Other tools lacked these necessary attributes and were therefore insufficient to pull the toy.

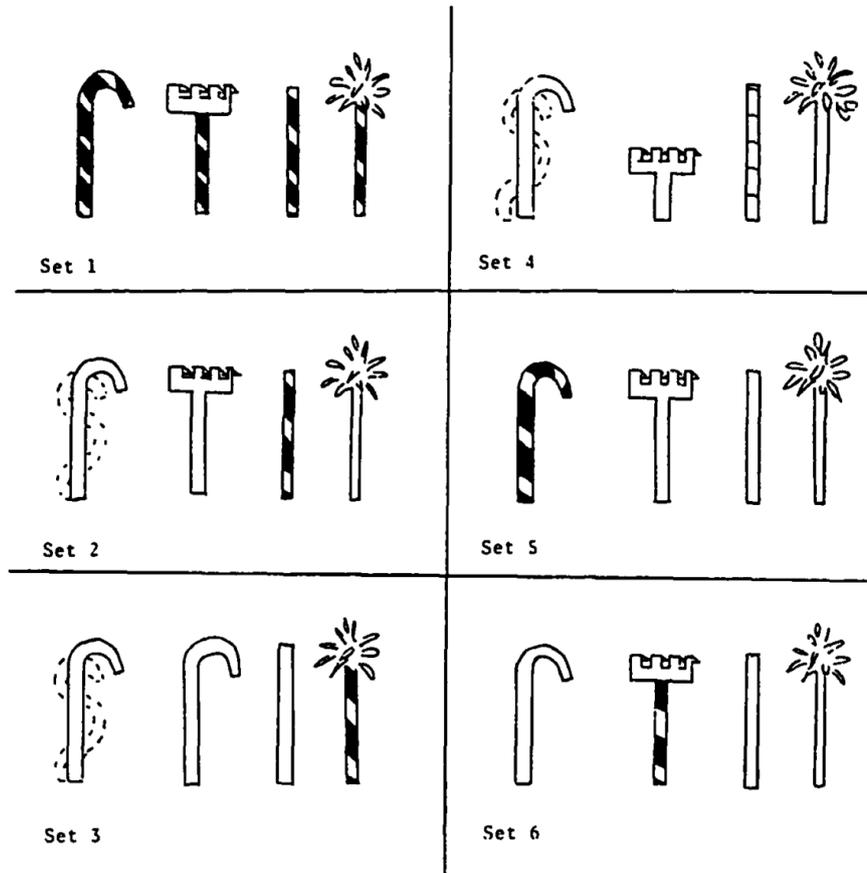


Figure 1.1: *Example tool sets used by Brown (1990) to explore transfer in young children. Children were first given set 1 and learnt that either the long, rigid hook or the long, rigid rake were appropriate for reaching a toy that was out of their reach. They were then given one of the other sets (2-6) to test their ability to transfer their knowledge of an appropriate pulling tool to a novel set of tools. The other sets in this figure are for children that showed an initial preference for the hook. Those that showed a preference for the rake were given similar sets but with the rake manipulated in the same way as the hook. As an example, set 2 tested children's ability to switch to a non-preferred perceptually novel tool in order to reach the toy. Figure adapted from Brown (1990).*

After the child had successfully learned to pull the toy towards them using an appropriate tool (with help from the mother if needed) they were given a second set of tools. There were five different options for this second set of tools and each one tested a different aspect of the knowledge transferred by the child. For example Set 2 in Figure 1.1 required the child to select the long rake, which was painted differently to the rake in the training set. This corresponds to selecting a perceptually novel and non-preferred tool that is appropriate for reaching the toy. Impressively, children performed extremely well when presented with the second set of tools, for example

92% of children made the switch to the perceptually novel rake in Set 2. Interestingly when the second tool set had an appropriate hook or rake only 62% of children showed preference for the tool they used in the training set. The conclusion of such experiments was that children were able to transfer the knowledge of what made a good tool (long, rigid and a useful end) to novel tool sets and thus act optimally to reach the toy. This represents one of the first clear demonstrations of transfer in young children. More recent work has since shown that the transfer of tool knowledge is even possible in young children when the original task is a video demonstration and the target task is a perceptually different real-world problem (Chen and Siegler, 2013).

In addition to learning words and how to use tools, more recent work by Lucas et al. (2014) has shown that 4 and 5 year old children are also able to transfer causal relationships to perceptually novel problems. In the study by Lucas et al. (2014) the authors investigated whether children and adults could learn disjunctive or conjunctive causal principles and transfer them to a novel problem. A disjunctive causal principle means that each individual cause has an independent probability of causing an event. In comparison, a conjunctive causal principle means that causes need to occur in conjunction to cause an effect and are not causal on their own. The basic paradigm to explore the learning of these two principles and their subsequent transfer can be seen in Figure 1.2.



Figure 1.2: *Behavioural paradigm used by Lucas et al. (2014) to explore the transfer of disjunctive and conjunctive causal principles in both children and adults. Participants were given either the conjunctive or disjunctive condition as a training set and then tested on the same test set. In the conjunctive condition, the combination of A and C objects was required to turn on the ‘blinket’ machine. In comparison, in the disjunctive condition only the presence of object A or C was required to turn on the ‘blinket’ machine. The test set was designed to be ambiguous so that F could be interpreted as the sole cause for turning on the ‘blinket’ machine or the combination of D and F could be interpreted as the cause. If transfer is successful then the interpretation chosen should match whether the participant was trained on the conjunctive or disjunctive condition. Figure adapted from Lucas et al. (2014).*

The paradigm consisted of two phases; a training phase and a test phase. In the training phase, participants saw pairs of objects on top of square bases (termed ‘blinket machines’), with the objects representing potential causal factors that could turn on the blinket machines (light up the square). The general goal for the participants was to identify which of the objects were blinkets i.e. which objects caused the blinket machine to turn on. One group of participants received a conjunctive set of objects, where a combination of objects was required to activate the blinket machine, while another group received a disjunctive set, where individual objects could turn on the blinket machine (Figure 1.2). The training phase was then followed by a test phase, which was the same for all participants regardless of which training set they were given. This test phase had perceptually novel objects and was designed to be ambiguous so that either a conjunctive or disjunctive principle could be correct. If transfer occurred then the participants should interpret either a combination of objects or individual objects as activating the blinket machines, depending on whether they had the conjunctive or disjunctive training set respectively.

For example, if a participant had received the disjunctive training set then they should infer that object D in the test set is not a blinket (Figure 1.2) because they have transferred over the disjunctive causal principle and inferred that object F is

the sole cause for the activation of a blanket machine. In comparison if a participant had received the conjunctive training set then they should infer that object D is a blanket because it activates the blanket machine in combination with blanket F, as per the conjunctive causal principle.

Interestingly, Lucas et al. (2014) found that both children and adults demonstrated the ability to transfer the disjunctive causal principle. However, children were better than adults at transferring the conjunctive causal principle, with adults tending to favour a disjunctive causal principle even after conjunctive training. While this provides another demonstration that young people are able to perform transfer in perceptually novel problems, these findings also raise some interesting questions about the discrepancies between children and adults. The authors suggest a Bayesian account for these differences. They suggest that children have weaker more diffuse priors, which mean that they are able to update their beliefs more readily. In other words, because children have received less real-world training than adults they are more flexible in their learning and weight new information more highly. This is interesting in terms of transfer because it is the development of these strong priors that provides the substrate required for transfer. Having strong, reliable abstract beliefs or ‘priors’ about the world allows one to select beneficial actions in perceptually novel environments with minimal feedback. In the case where an adult has to learn an abstract concept that violates their prior, they will do worse than a child because they are more likely to transfer their real-world prior. In contrast, children will readily update their prior beliefs based on the information in the psychological study and transfer the new abstract concept more readily. In the case of Lucas et al. (2014), this lead to better performance by children on their task.

1.2.2 Adult Behaviour

Many examples of adults performing rapid learning and transfer in new situations exist. The aforementioned work by Lucas et al. (2014) actually demonstrates an instance where adult transfer can hinder the efficiency of learning in a new situation. However, often rapid learning and transfer can be beneficial tools for adults. For example, Lake et al. (2015) have proposed that when adults see an image of a novel two-wheeled vehicle they only need one example to parse it into its constituent parts

and learn the new visual concept. It is likely that this efficiency of learning is due to the transfer of knowledge from past experience with the constituent parts. Aside from concept learning, rapid learning can even occur in adults during perceptual learning, which involves consistent changes to perception. For instance, Poggio et al. (1992) have shown that people participating in a novel perceptual discrimination task can significantly improve performance after only a few trials.

To demonstrate how pervasive rapid learning and transfer is in adult cognition, Dubey et al. (2018) investigated the strategies used by adults when playing 2D video games for the first time. Throughout this thesis we shall use the domain of video games to explore rapid learning and transfer. Video games represent a useful medium for studying transfer because the perceptual features and underlying rules of the world can be easily manipulated. In addition to this tight control of the environment, video games also involve sequential decision-making, basic physics and fine motor control, which are all hallmarks of naturalistic behaviour in the real world.

In the study, Dubey et al. (2018) performed a range of game manipulations that targeted different forms of prior knowledge (Figure 1.3). The types of prior knowledge included: semantics, object identity, affordances, visual similarity, object interaction and physical laws (gravity). All of the manipulations significantly affected performance in terms of the time taken to complete a level, the number of deaths and the number of states explored. This indicates that adults transfer a variety of prior knowledge from past experiences in order to solve video games. For example, if the objects in the 2D game, such as keys and spikes, were changed to coloured squares people were unable to use past semantic knowledge to infer which objects to avoid and which objects to pursue.

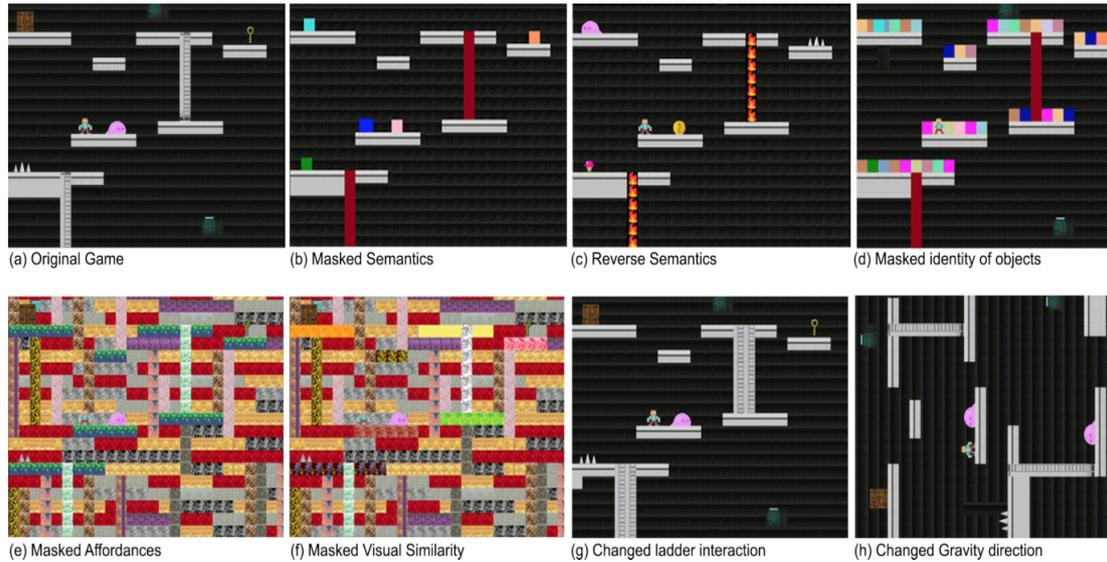


Figure 1.3: 2D platform game used by Dubey et al. (2018) to explore the effect of human prior knowledge on video game performance. In order to complete the original game (a), players had to move the avatar to the key and then to the door. The trial ended when the avatar died, which could happen either by falling to the bottom of the screen or by colliding with either the pink blob or the spikes. Different game manipulations were performed in order to inhibit the transfer of different forms of prior knowledge to the game. The manipulations were as follows: (b) Masked Semantics - Object identities were masked by changing them to coloured squares. (c) Reverse Semantics - Object identities were changed so that their semantics were reversed i.e. the pink blob became a coin. (d) Masked Identity of Objects - Coloured squares were placed on all platform areas including the objects. (e) Masked Affordances - Free space was filled with different textures and platforms were changed to a similar texture. (f) Masked Visual Similarity - Platforms and ladders were given different visual appearances. (g) Changed Ladder Interaction - Instead of pressing up to climb the ladder, participants had to alternate between pressing left and right. (h) Changed Gravity Direction - The original game was rotated 90 degrees. Figure adapted from Dubey et al. (2018).

Another interesting finding reported by Dubey et al. (2018) involved the relative effect sizes of the different manipulations. The manipulations with the largest impact upon performance, such as masking objects and perceptual similarities, were also those that disrupted priors thought to form earliest in human development. It therefore seems that knowledge formed early on development is highly relied upon in novel environments to guide decisions and action selection. Whether this knowledge is learnt early on in life because it is more useful in guiding actions or because it is more prevalent in the statistics of the environment remains to be elucidated.

1.3 Reinforcement Learning as a Basic Mechanism for Learning

With the aforementioned examples in mind, the purpose of this thesis is to explore the internal computations and external factors that support rapid learning from new experiences and the transfer of knowledge from past experiences. This is a broad topic and so we require a method of constraining the problem and grounding it in a theoretical framework. Reinforcement Learning (RL) has emerged as a predominant framework for describing how people map perceptual input to action based on reward, and is thought to be a fundamental mechanism for learning in the brain (Niv, 2009; O’Doherty et al., 2015). Crucially, RL also has strong mathematical foundations, which allows us to explore the computational properties needed to perform rapid learning and transfer. In addition, RL provides us with a natural measure of efficiency; the fewer actions needed to achieve a certain amount of reward the more efficient an agent is. We can therefore investigate how mechanisms of rapid learning and transfer directly affect the efficiency of RL based on the number of actions needed to obtain reward in a new situation. The remainder of this section describes the basic principles behind RL and how they manifest themselves in the brain.

1.3.1 Operant and Classical Conditioning

Reinforcement Learning (RL) describes how agents can use perceptual observations and reward signals to select actions that subsequently maximise future reward. The birth of RL can be traced back to seminal work carried out on how animals associate motor actions with sensory stimuli and rewards. In particular the work by Thorndike (1911) and Skinner (1935) on operant conditioning provided the first descriptions of how animals select actions to increase the probability of favourable events and reduce the probability of adverse events. This ability to select actions in order to alter the environment in one’s favour is seen as one of the hallmarks of intelligent behaviour and is central to RL.

Work on Pavlovian conditioning also had an impact upon the development of RL as it was concerned with learning the predictive relationships between sensory

stimuli (Yerkes and Morgulis, 1909). This Work gave rise to one of the earliest computational accounts of what is now considered RL; the Rescorla-Wagner model (Rescorla, 1972). In the Rescorla-Wagner model two crucial assumptions were made: (1) learning occurred when there was a mismatch between the predicted value of an event and the actual value, and (2) the value of each stimuli was summed to obtain the predicted value of an event. These simple assumptions were able to explain a multitude of phenomenon in the Pavlovian learning literature including blocking (Kamin, 1967), overshadowing (Reynolds, 1961) and inhibitory conditioning (Rescorla and Lolordo, 1965).

1.3.2 AI and Reinforcement Learning

While this early work in ‘computational psychology’ highlighted some of the key principles for learning from reward, it was the field of Artificial Intelligence (AI) that provided the first rigorous mathematical description that we now refer to as RL. Proposed by Sutton and Barto (1998), RL built on the assumptions of the Rescorla-Wagner model by also taking into account the timing of stimuli and relating learning to actions. More specifically, RL uses a scalar reward signal to learn which actions to select in order to maximise future rewards based on the consequences of those actions. RL describes this learning problem as an interaction between an agent and its environment. The agent observes the state of the environment (s_t) at a given time point (t) and selects an action (a_t). This action then leads to a change in state (s_{t+1}) and an associated reward (r_{t+1}). This interaction is repeated over and over as the agent learns the best mapping from states to actions in order to maximise reward. Importantly, the reward can often be 0 meaning that the teaching signal is sparse and actions made many time-steps before the reward may have been responsible for generating the reward. This highlights the crux of the RL problem and is often referred to as the *credit assignment problem*; which actions in which states are responsible for generating reward?

With only three main signals between the agent and its environment (s , a and r), RL appears on the surface to be a very simple framework for exploring reward-driven learning. However, with just these three signals a plethora of methods have been proposed to solve the credit assignment problem (see Section 2.1.2 for a detailed

discussion of these methods) and ultimately learn the best policy (a mapping from states to actions) for a given task. The majority of these solution methods rely on learning a value function, much in the same way the Rescorla-Wagner model updates value estimates of conditioned stimuli using errors between predicted and actual values. These value functions often denoted $V(s_t)$ for state evaluation, or $Q(s_t, a_t)$ for state-action evaluation, are an estimate of the expected future reward from a given state or state-action pairing. In mathematical terms, the expected future reward is usually calculated as an arbitrary function of the sequence of rewards experienced after time t from a state or state-action pairing. In practice this function is usually a discounted sum of the rewards so that rewards nearer in time are given more weighting in the value computation. Crucially these values store information about future consequences of actions, which is needed to solve the credit assignment problem.

1.3.3 Reinforcement Learning in the Brain

Despite modern RL starting out as predominantly an AI framework, it shares many similarities with the problem faced by biological agents; using reward to select actions based on the state of the environment in order to obtain more reward. As a result, neuroscience has taken many of the predictions made by algorithms attempting to solve the RL problem and looked for neural correlates in the brain. Most famously, Schultz et al. (1997) found midbrain dopaminergic neurons appear to encode Reward-Prediction Errors (RPEs), which are a key component of Temporal Difference (TD) learning algorithms. TD learning algorithms express the value of a state or state-action pair as the reward received after that state or state-action pair plus the value of the subsequent state or state-action pair. This bootstrapping of value estimates allows TD learning to occur at every time step and propagates reward information backwards in time. Crucially TD learning relies on RPEs to drive learning; the difference between the predicted and actual values are used to update the value estimate.

TD learning makes specific predictions about how RPEs should change during learning. For example, at the start of learning there should be a large positive RPE when the reward is presented because it is unexpected and so the difference

between the expected and actual value is large and positive. Subsequently this large positive RPE should occur earlier and earlier in learning until it is centered on the earliest reliable predictor of reward i.e. the conditioned stimulus. There should no longer be a positive RPE near the reward because it has already been predicted by the conditioned stimulus. In comparison, there is always a positive RPE when the conditioned stimulus is presented because the agent cannot know when the conditioned stimulus will occur. Subsequently if the reward is removed then the positive RPE will remain when the conditioned stimulus is presented but there will be a negative RPE when the agent expected to receive the reward because the predicted value is now larger than the actual value. Strikingly, this distinct learning profile of RPEs was found to occur in the firing of phasic midbrain dopaminergic neurons of monkeys as they were trained to associate a stimulus with a juice reward (Schultz et al., 1997). This led to a theory known as the *reward prediction error hypothesis of dopamine*, which has subsequently been validated by a host of other studies (Hollerman and Schultz, 1998; Tobler et al., 2003; Bayer and Glimcher, 2005). The theory represents a prime example of how the interaction between AI and neuroscience can be beneficial, and shows the power of using RL as a framework for trying to understand computations occurring in the brain.

1.4 Into the 21st century: Deep Reinforcement Learning

While simple computational models of Reinforcement Learning (RL) are able to describe a wealth of behavioural and neurological findings, substantial obstacles still prevent it from being a complete account of human reward-driven behaviour. Until recently, one of the greatest problems RL models faced was mapping complex, naturalistic stimuli to actions. The perceptual input that the brain receives is extremely rich, from the activation of rods and cones in the retina to the movement of hair cells in the ear. This represents a significantly difficult learning problem as these high-dimensional inputs lead to the curse of dimensionality. The curse of dimensionality refers to the fact that as the number of dimensions increases so does the volume of space being represented. This causes data to become increasingly sparse

and so more data is required to provide the same coverage of space. Until recently, classic RL models had to rely on hand-crafted state representations that manually solved the curse of dimensionality.

Fortunately, advances in the field of machine learning have started to provide solutions to this problem. In particular, the combination of both Deep Neural Networks (DNNs) and RL, often referred to as Deep RL, has led to the development of algorithms that can achieve human-level performance on complex perceptual-motor tasks such as playing video games from raw pixel images (Mnih et al., 2015). This represents an exciting advance for cognitive scientists because it provides a computational framework that can be used as a reference point to explore how the brain might use RL to process raw perceptual input into action based on reward. The rest of this section explores how Deep Learning and Deep RL relate to computations occurring in the brain.

1.4.1 Deep learning

Deep Learning refers to Artificial Neural Networks (ANNs) that typically contain many hidden layers (Schmidhuber, 2015). This property allows them to learn complex hierarchical representations of high-dimensional input. While largely considered a machine learning approach, Deep Learning's origins can be traced back to classic connectionist or parallel distributed processing approaches in cognitive science (McClelland et al., 1986). Connectionist approaches in cognitive science aim to use ANNs to describe cognition using learning mechanisms inspired by the brain. ANNs represent information as a network of connected units, where both the units and the connections can take on numerical values. The value of each unit represents its level of activation and the value of each connection represents the strength of that connection. Unit values are computed using an activation function, which is commonly a non-linear function that is applied to a linear summation of a unit's inputs. Interestingly, evidence from neuroscience has suggested that this linear summation of inputs may exist at the neuronal level in the brain (Morel et al., 2018; Cash and Yuste, 1999). Typically input will be presented to the network by setting a selection of the units to the value of the input and the output of the network will be read from another subset of units. Crucially learning is implemented in ANNs by altering

the strength of the connections between units in order to achieve the desired input-output mapping. ANNs can vary in terms of activation functions, architecture and training rules.

The central contribution of the connectionist approach is the idea that information can be processed in parallel and represented across many units, with distributed representations being an emergent property of learning. Connectionists argue that this central principle is a vital computational property of the brain and hence cognition, whether it be via interconnected brain regions or neurons. Several studies have found a close correspondence between the distributed representations learnt by Deep neural Networks and those found in sensory areas of the human brain. In particular, Deep Neural Networks with similar processing constraints to the human visual system known as Deep Convolutional Neural Networks (DCNNs) appear to produce hierarchical representations similar to the ventral visual stream when trained to categorize natural images (Güçlü and van Gerven, 2015; Yamins and DiCarlo, 2016). The way connectionist approaches learn distributed representations has been particularly influential in developmental psychology, where they have been proposed to model how children learn representations of the world based on the statistical properties of their environment (Plunkett et al., 1997; Quartz and Sejnowski, 1997; Mareschal, 2010). More specifically they have offered explanations for non-linear development profiles, specialization of specific brain regions and behavioural dissociation's (Munakata and McClelland, 2003).

Despite the success of connectionist approaches in cognitive science their biological plausibility is still a topic of debate. This is particularly true of Deep Learning, which typically relies upon backpropagation to propagate prediction errors from the output layers to the input layers in order to update connections between layers. This has been declared biologically implausible because it does not rely on locally available values for learning but instead requires information to be passed sequentially through each of the layers. However, a growing body of research is suggesting that this criticism may be ill founded and that the brain could well be implementing solutions that approximate backpropagation (Sacramento et al., 2018; Guerguiev et al., 2017; Mazzoni et al., 1991; O'Reilly, 1996; Scellier and Bengio, 2017; Whittington and Bogacz, 2017; Lillicrap et al., 2016). In general, these approaches aim

to re-create the success of backpropagation in training Deep Neural Networks while maintaining biological plausibility. It is also worth mentioning that Deep Learning and its core properties are not tied to backpropagation. For example, Deep Belief Networks (DBNs) rely upon a local biologically plausible learning rule known as Hebbian Learning (Testolin et al., 2017). DBNs also use unsupervised learning, which is thought to be a critical component of learning in the brain as it is unlikely to have access to a teaching signal for each learning event. Unsupervised methods are therefore required to make the most of the data that the brain is exposed to (Testolin and Zorzi, 2016).

In summary, Deep learning has a rich history in cognitive science through its relationship to connectionist approaches. One of its most appealing properties is its ability to learn representations in a distributed and hierarchical fashion that appears to mimic the representations learnt by the brain. While Deep Learning often receives criticism for the biological implausibility of backpropagation, it is not dependent on it and a variety of other training rules may be able to achieve a similar outcome.

1.4.2 Deep Reinforcement Learning

Deep Reinforcement Learning (Deep RL) takes Deep learning one step further by combining it with RL principles. In technical terms, it relies on using Deep Learning to approximate value functions and policies in order to solve the RL problem. The previous sections have highlighted how the computations used in RL and Deep Learning may be similar to those used by the brain. This section now focuses on evidence that both of them may be used in combination by the brain.

The use of Deep Neural Networks (DNNs) in Deep RL allows an agent to deal with high-dimensional state representations by learning non-linear continuous functions of their input. Given enough units, DNNs are able to represent any continuous non-linear function. This means that DNNs are prone to over-fitting and are highly sensitive to noise in the input, particularly when the number of data points is small. This problem is amplified in Deep RL because the reward value is sparse and there are many potentially actions that could be attributed to a particular reward outcome. This introduces a large source of noise or variance, which can be the culprit of over-fitting. In order to overcome this problem and help to reduce variance, Deep

RL approaches typically learn the value of actions relative to the overall value of the current state. This is often referred to as the ‘advantage’ of an action. For example, if the value of a state is the same regardless of which action is taken then the advantage will be zero and the Deep RL algorithm will not update its policy because the reward outcomes are not attributable to that particular action. This greatly helps to reduce the variance of the Deep RL policy updates by reducing sources of noise. The use of advantages in Deep RL is of interest from a biological perspective because it has been proposed that such quantities are also encoded in the brain (FitzGerald et al., 2009; Philiastides et al., 2010; Morris et al., 2014). Deep RL therefore provides a computational reason for why the brain should encode these relative action values.

Aside from over-fitting and the bias-variance trade-off, another common problem in RL is the balance between exploration and exploitation. Without some degree of exploration RL agents are likely to settle on sub-optimal policies as they have not sampled potentially better actions. This is particularly problematic in Deep RL because the state is typically high-dimensional and so it is not possible to just keep a record of which states still need to be visited. Interestingly, Plappert et al. (2017) have shown that adding noise to the parameters of the DNNs used in Deep RL can promote useful exploratory behaviour. The fact that this approach works from a Deep RL perspective is interesting because the brain is also known to have several sources of noise in its connections between neurons. This biological noise can act on short time-scales e.g. probabilistic synapses (Llera-Montero et al., 2019), or longer timescales e.g. fluctuations in the size of dendritic spines (Yasumatsu et al., 2008). It is therefore possible that the brain and Deep RL are utilising similar strategies to solve the exploration-exploitation trade-off, lending further support to an analogy between the two (Gershman and Ölviczky, 2020).

1.5 The Problem of Efficiency in Deep Reinforcement Learning

The previous sections have highlighted how Deep Reinforcement Learning (RL) can serve as an attractive analogy for how the brain achieves a mapping from high-

dimensional perception to action, based on reward. For example, the emergence of distributed hierarchical representations in Deep Neural Networks (DNNs) appears to share similarities with the representations learnt by the brain. Equally, key error signals utilised by RL algorithms appear to be encoded by dopaminergic midbrain neurons. Finally, solutions to the problems of over-fitting and exploration, which are particularly pertinent to Deep RL, appear to mirror mechanisms that can also be found in the brain such as advantage values and weight perturbations.

Despite this apparent harmony between Deep RL and computations in the brain, there are several striking differences. Most importantly for the purpose of this thesis, Deep RL algorithms have been heavily criticised for being extremely inefficient and requiring vast amounts of training data (Lake et al., 2017). For example, one of the first demonstrations of Deep RL came in the form of the Deep Q-Network (DQN) by Mnih et al. (2015). DQN was able to learn to play Atari 2600 video games at or above human-level performance without the hand-coding of any game-specific knowledge. However despite this impressive feat, DQN required 50 million game frames for training which equates to approximately 38 days of game-play. In contrast, the human participants used to form the baseline only had 2 hours of training before evaluation and brought with them vast amounts of prior knowledge (Dubey et al., 2018). This serves to demonstrate the significant lack of efficient learning in Deep RL models.

This gulf in efficiency between human RL and Deep RL suggests that Deep RL lacks the two processes that are the focus of this thesis; (1) the ability to rapidly learn from new information and (2) the ability to transfer past knowledge to the current task. One could argue that one of the reasons for lack of efficiency displayed by DQN is its lack of prior knowledge. Human participants entered the training process with a wealth of prior knowledge and experience that was relevant for playing the Atari games. In comparison DQN started off with no prior knowledge that it could utilise. However, simply training DQN on other games beforehand to imbue them with prior knowledge is unlikely to address the differences in efficiency. This is because Deep RL algorithms have a tendency to catastrophically fail when some aspect of the current task changes (Lake et al., 2017). For this reason the weights of DQN had to be reset for each game so that it could learn anew. This highlights how Deep RL

approaches, such as DQN, are unable to transfer knowledge between environments and that they can actually exhibit negative transfer whereby prior learning has a negative impact upon performance.

1.6 Outline of the Thesis

The purpose of this thesis is to investigate the brain’s internal computations and the external environmental factors that contribute to rapid learning and transfer in Reinforcement Learning (RL). Throughout the thesis we shall use Deep RL as analogy to the brain to help guide our thinking. This will allow us to explore what fundamental computations Deep RL is missing in order to replicate the efficiency shown by human RL. Indeed, this comparison is a two-way street in that both cognitive science and Artificial Intelligence (AI) may benefit in the process. Advances in Deep RL algorithms can provide testable predictions about underlying computations in the brain, while properties of the brain can be utilised in Deep RL algorithms to improve their data efficiency.

The research presented in this thesis follows two main streams. Firstly, we explore how the computational properties of different learning systems in the brain support efficient RL and how they can be used to improve the efficiency of Deep RL algorithms. This research utilises Complementary Learning Systems (CLS) theory as a guiding framework for how the brain is organised and allows parallels with Deep RL algorithms to be drawn. In its original form, CLS theory describes how the neocortex and hippocampus have complementary properties that support learning and complex behaviour. Importantly, CLS theory highlights how the brain has several learning systems with different computational properties. This is in contrast to classic Deep RL algorithms that tend to rely on a single network for learning. This suggests that combining multiple systems with fundamentally different computational properties, may be a fruitful approach to improving the capabilities of Deep RL algorithms.

We begin this line of thinking in Chapter 3 by using a CLS framework to review computational work that attempts to address the efficiency problems of Deep RL (see Section 1.5). This review helps to motivate the proposal of a new Deep RL algorithm in Chapter 4, termed Complementary Temporal Difference Learning (CTDL).

Importantly, CTDL exploits the benefits of both a neocortical and a hippocampal learning system to improve the efficiency of Deep RL and makes predictions about how the two systems interact in the brain. In Chapter 5 we build upon this work and argue that CLS theory should be extended to include the dissociation between the Pre-Frontal Cortex (PFC) and sensory cortices. We support this argument by reviewing further computational modelling that improves the efficiency of Deep RL and that captures cognitive phenomenon associated with the PFC. As a result of this review, we propose another novel algorithm called Selective Particle Attention (SPA) in Chapter 6. Crucially, SPA mimics interactions between the PFC and sensory cortices in order to implement visual feature-based attention and improve the efficiency of current Deep RL approaches. Both CTDL and SPA highlight the importance of considering the brain as a network of complementary learning systems and describe how their interactions can support efficient RL in the brain.

The second stream of research in this thesis focuses on how the degree of perceptual similarity between consecutive experiences affects people’s ability to transfer knowledge. Several theories from domains such as analogical reasoning and concept learning make conflicting predictions about how the degree of perceptual similarity between tasks affects transfer ability. Equally many Deep RL algorithms rely on interleaved training to remove spurious similarities between consecutive experiences. To help address these conflicting predictions, Chapter 7 presents a series of behavioural experiments that investigate how the degree of perceptual similarity between consecutive experiences affects transfer. The majority of this work is conducted within the domain of video games because they involve sequential decision making based on reward and represent a more naturalistic setting compared to classical experimental paradigms. The results of this work suggest that the degree of perceptual similarity between experiences may have little impact upon transfer performance. However, the empirical studies do highlight an interaction between perceptual similarity and the ability of people to utilise explicit rules for transfer. We use the results of these empirical studies to discuss the implications for the analogy between the brain and Deep RL algorithms. Our hope is that these results can help to constrain theories of transfer in humans and the assumptions made by Deep RL models.

In the next chapter we begin by covering the background material required to understand the other chapters of this thesis. This chapter particularly focuses on the mathematical foundations of RL and Deep RL, which are important for the analogy between Deep RL and the brain that we will form throughout this thesis.

Chapter 2

Background

Overview

In this chapter we provide the necessary background material to understand Deep Reinforcement Learning (RL). We start by outlining the mathematical foundations of RL as described by Sutton and Barto (2018) (Section 2.1). We then provide a brief overview of the basic principles of Deep Learning and some common network architectures that will appear throughout this thesis (Section 2.2). Finally we demonstrate how the two methods can be combined to produce Deep RL (Section 2.3).

2.1 Reinforcement Learning

The following description of Reinforcement Learning (RL) draws from the work of Sutton and Barto (2018).

2.1.1 The Reinforcement Learning Problem

RL is a computational framework that attempts to explain how an agent can act in its environment in order to maximise reward. According to RL theory, the interaction between the agent and its environment can be characterised by three main signals:

1. $s_t \rightarrow$ The state of the environment at time t ($s_t \in S$)

2. $a_t \rightarrow$ The action chosen by the agent at time t ($a_t \in A$)
3. $r_{t+1} \rightarrow$ The reward obtained by the agent at time $t + 1$ ($r_{t+1} \in R$)

Where S is the set of all possible states, A is the set of all possible actions and R is the set of all possible reward values. The contributions of these three signals are commonly depicted by the diagram shown in Figure 2.1

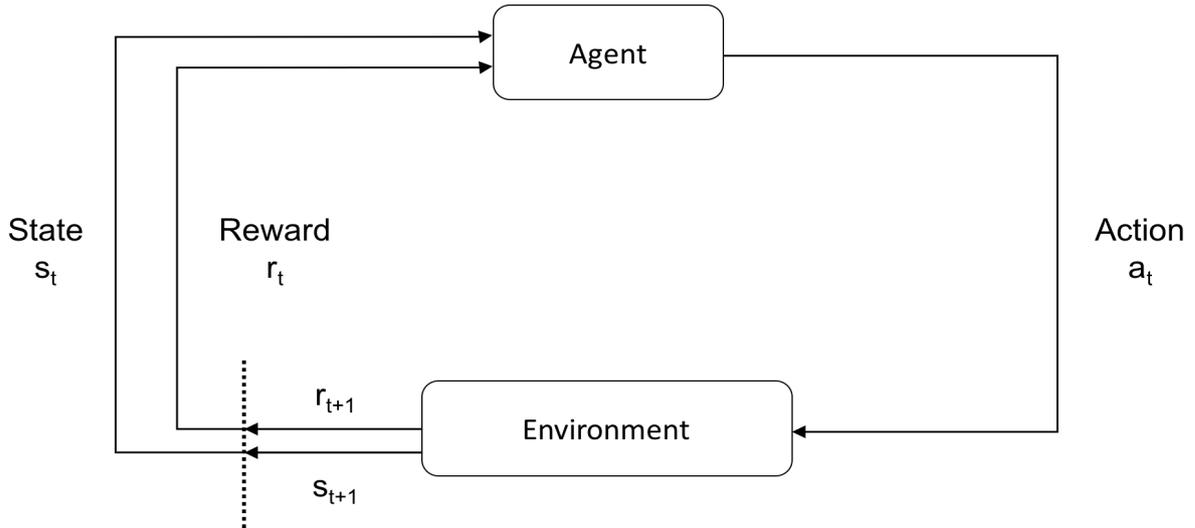


Figure 2.1: *The Reinforcement Learning (RL) loop. The agent observes the state of the environment at time t (s_t) and selects an action (a_t). The environment responds to this action and produces a new state (s_{t+1}) and a scalar reward value (r_{t+1}).*

As the diagram shows, the interaction between the agent and its environment can be described as a constant loop. The agent chooses an action (a_t) based on the current environmental state (s_t) and the environment responds to this action returning the next state (s_{t+1}) and associated reward (r_{t+1}). This process can proceed indefinitely as the agent sequentially chooses actions in response to the environment. The central goal in RL is to maximise the amount of reward obtained by the agent. This is often referred to as maximising the expected *return*, where the return R_t is an arbitrary function of the reward sequence experienced by the agent from time t onwards. One of the most common definitions of the return is the discounted sum of future rewards:

$$R_t = \sum_{t'=t}^T \gamma^{t'-t} r_{t'} \quad (2.1)$$

The discount factor $\gamma \in (0, 1)$ is applied at each time step so that immediate rewards are worth more than distant rewards. γ is required for non-episodic environments i.e. environments where there is no clear end state, because it ensures that the sum will converge to a finite value as k approaches ∞ .

Since the primary goal of the agent is to maximise reward, the reward signal serves as the primary feedback signal for evaluating actions. This evaluation process can be challenging because reward signals may be sparse and delayed with respect to the action(s) that caused them. This is often referred to as the *credit assignment problem*; deciding which actions are responsible for a given outcome, and is central to the Reinforcement Learning Problem.

2.1.1.1 Markov Decision Processes

In RL the state at any given time point (s_t) is commonly treated as a Markov state. This means that the state satisfies the Markov property: the probability of future states given past and present states only depends on the current state. In other words, future outcomes are only a function of the current state and not of previous states. Mathematically we can write the Markov property as follows:

$$\begin{aligned} P(s_{t+1} = s', r_{t+1} = r \mid s_t, a_t, r_t, s_{t-1}, a_{t-1}, r_{t-1}, \dots) = \\ P(s_{t+1} = s', r_{t+1} = r \mid s_t, a_t) \forall s', r, s_t, a_t \end{aligned} \tag{2.2}$$

That is the probability of the next state being equal to s' and the next reward being equal to r given all previous states and actions is equal to the probability of the next state being equal to s' and the next reward being equal to r given the current state and action. This basic assumption greatly simplifies the RL problem because now the agent only needs to consider it's current state in order to select an action and maximise the expected return.

In the real-world biological agents are often faced with problems where the Markov property does not hold and past states do have an impact upon the probability of future states. Many RL algorithms will perform well even when the states it encounters are not strictly Markov. However the closer the states are to satisfying the Markov property the better the RL algorithm will perform. While this may seem

highly prohibitive, often the representation of the state signal can decide whether a state satisfies the Markov property or not. For example, to train an agent to play a video game using RL one could provide an agent with the last four game frames as the state signal. This provides the agent with enough information to infer what objects are present on the screen, what direction they are moving in and how fast they are moving. These key variables should provide enough information for the agent to choose optimal actions, therefore satisfying the Markov property.

Assuming the states in our RL problem satisfy the Markov property, we can frame the problem as a Markov Decision Process (MDP). There are five main components of a MDP:

1. S - The set of all possible states ($s_t \in S$)
2. A - The set of all possible actions ($a_t \in A$)
3. $P_{ss'}^a$ - The transition function
4. $R_{ss'}^a$ - The reward function
5. γ - The discount factor ($0 \leq \gamma \leq 1$)

$P_{ss'}^a$ is called the transition function and it defines the probability distribution over the next state given the current state and action. Mathematically we can define this as:

$$P_{ss'}^a = P(s_{t+1} = s' \mid s_t = s, a_t = a) \quad (2.3)$$

$R_{ss'}^a$ specifies the expected reward value for transitioning from state s to state s' via action a . Mathematically we can define this as:

$$R_{ss'}^a = \mathbb{E}[r_{t+1} \mid s_t = s, a_t = a, s_{t+1} = s'] \quad (2.4)$$

Importantly, $P_{ss'}^a$ and $R_{ss'}^a$ provide all the information needed to describe the dynamics of the environment and to make decisions that maximise the expected return. Both $P_{ss'}^a$ and $R_{ss'}^a$ rely on the Markov property of the state signal in order to simplify the probability distribution.

2.1.1.2 Value Functions and Policies

At the heart of most solution methods in RL you will find either a value function and/or a policy. In simple terms, value functions estimate the expected return from a given state with respect to a particular policy. Policies, commonly denoted π , are mappings from states to actions and therefore specify the action to be taken by the agent for any given state:

$$\pi : s \mapsto a \tag{2.5}$$

π may be a deterministic or stochastic mapping. Value functions are dependent on the policy π because the expected return from a given state will be highly dependent on the action taken by the agent both in that state and also future states. Typically we can define two different types of value function, either a state-value function $V^\pi(s)$ or an action-value function $Q^\pi(s, a)$. A state-value function estimates the expected return of a given state s , whereas an action-value function estimates the expected return of an action a given state s . Using the discounted sum of future rewards as our return we can define these value functions as follows:

$$V^\pi(s) = \mathbb{E}_\pi[R_t \mid s_t = s] \tag{2.6}$$

$$= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\right] \tag{2.7}$$

$$Q^\pi(s, a) = \mathbb{E}_\pi[R_t \mid s_t = s, a_t = a] \tag{2.8}$$

$$= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a\right] \tag{2.9}$$

Using the components of an MDP (see section 2.1.1.1), one can show that these value functions satisfy particular recursive relationships. Using the state-value function as an example, one can define it as a function of itself using $P_{ss'}^a$, and $R_{ss'}^a$:

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s \right] \quad (2.10)$$

$$= \mathbb{E}_\pi \left[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_t = s \right] \quad (2.11)$$

$$= \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_{t+1} = s' \right]] \quad (2.12)$$

$$= \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V^\pi(s')] \quad (2.13)$$

The value of any given state is simply the reward obtained immediately from that state plus the value (expected return) of the next state, averaged over the policy and one-step dynamics of the environment. A similar recursive relationship for the action-value function can be defined as follows:

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a \right] \quad (2.14)$$

$$= \mathbb{E}_\pi \left[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_t = s, a_t = a \right] \quad (2.15)$$

$$= \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_{t+1} = s' \right]] \quad (2.16)$$

$$= \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma \sum_{a'} \pi(s', a') Q^\pi(s', a')] \quad (2.17)$$

Here the action-value function is defined as the immediate reward given a specific action in the current state, averaged over the environment dynamics, plus the value of actions in the successor state, averaged over the environment dynamics and the agents policy. These two recursive equations are known as the Bellman equations and form the basis of many RL solution methods including dynamic programming and temporal difference learning:

$$V^\pi(s) = \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V^\pi(s')] \quad (2.18)$$

$$Q^\pi(s, a) = \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma \sum_{a'} \pi(s', a') Q^\pi(s', a')] \quad (2.19)$$

2.1.2 Solution Methods

2.1.2.1 Optimal Policies

In general, solving the RL problem equates to finding a policy π (a mapping from states to actions) that achieves the maximum amount of reward over time. An optimal policy, denoted π^* , is one that achieves an expected return that is greater than or equal to all other policies for all states (and actions):

$$V^*(s) = \max_{\pi} V^\pi(s) \quad \text{for all } s \in S \quad (2.20)$$

$$Q^*(s, a) = \max_{\pi} Q^\pi(s, a) \quad \text{for all } s \in S \text{ and } a \in A(s) \quad (2.21)$$

Finding the optimal policy (there may be more than one) is equivalent to achieving the largest expected return and therefore solving the RL problem. There are four primary methods for finding the optimal policy:

1. Value-Based Methods \rightarrow Learn value functions to infer the optimal policy
2. Policy-Based Methods \rightarrow Learn the optimal policy directly
3. Actor-Critic Methods \rightarrow Learn both a value function and a policy
4. Model-Based Methods \rightarrow Use knowledge of the environment's dynamics to infer the optimal policy

The next section describes each one of these methods in more detail and covers some of the canonical algorithms for solving the RL problem.

2.1.2.2 Value-Based Solution Methods

Value-based methods rely on value functions to infer the optimal policy. Since value functions are dependent on a particular policy, there exists an optimal state value function $V^*(s)$ and action value function $Q^*(s, a)$, which define the expected return of a state or state-action pair given the agent follows the optimal policy:

$$V^*(s) = \max_{\pi} V^{\pi}(s) \quad \text{for all } s \in S \quad (2.22)$$

$$Q^*(s, a) = \max_{\pi} Q^{\pi}(s, a) \quad \text{for all } s \in S \text{ and } a \in A(s) \quad (2.23)$$

As in Section 2.1.1.2, these value functions can be written as recursive equations, which are known as the Bellman optimality equations. The Bellman optimality equations are written without reference to a specific policy because they rely on the fact that the value of a state with respect to the optimal policy is the same as the expected return for the best action from that state. The Bellman optimality equations for $V^*(s)$ and $Q^*(s, a)$ are therefore as follows:

$$V^*(s) = \max_{a \in A(s)} Q^{\pi^*}(s, a) \quad (2.24)$$

$$= \max_a \mathbb{E}_{\pi^*}[R_t \mid s_t = s, a_t = a] \quad (2.25)$$

$$= \max_a \mathbb{E}_{\pi^*}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a\right] \quad (2.26)$$

$$= \max_a \mathbb{E}_{\pi^*}\left[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_t = s, a_t = a\right] \quad (2.27)$$

$$= \max_a \mathbb{E}[r_{t+1} + \gamma V^*(s_{t+1}) \mid s_t = s, a_t = a] \quad (2.28)$$

$$= \max_a \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V^*(s')] \quad (2.29)$$

$$(2.30)$$

$$Q^*(s, a) = \mathbb{E}[r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') \mid s_t = s, a_t = a] \quad (2.31)$$

$$= \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma \max_{a'} Q^*(s', a')] \quad (2.32)$$

$$(2.33)$$

Importantly, given the optimal value function (V^* or $Q^*(s, a)$) for an MDP, the optimal policy π^* is represented implicitly as the policy that acts greedily with respect to the optimal value function. In the case of the optimal state value function, the best action in any given state is the one that maximises the sum of the immediate reward and the discounted value of the next state:

$$\pi^*(s) = \arg \max_a \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V^*(s')] \quad (2.34)$$

This is akin to making a one-step lookahead search and then acting greedily. Taking greedy actions works because the optimal state value function takes into account the reward consequences of all possible future behaviours. This is powerful because it means the agent does not need to evaluate lots of future actions in order to make an optimal decision. For the action-value function, the best action is simply the one that has the largest value from the current state:

$$\pi^*(s) = \arg \max_a Q^*(s, a) \quad (2.35)$$

In the case of the optimal action value function there is no need to perform a one-step lookahead search. $Q^*(s, a)$ stores the results of all one-step lookahead searches and provides the optimal expected long-term return as a locally and immediately available value. So by representing the value function as a function of states and actions, rather than just states, optimal actions can be chosen without having to know anything about the environment's dynamics.

The central goal of value-based methods is to learn the optimal value function and therefore obtain the optimal policy. Given the Bellman optimality equations above, one can in fact solve for the optimal value function as a series of N nonlinear equations in N unknowns, where N is the number of states, as long as $P_{ss'}^a$ and $R_{ss'}^a$

are known. Unfortunately, this exhaustive search approach is rarely possible due to three main reasons:

1. The environment's dynamics are often unknown i.e. $P_{ss'}^a$ and $R_{ss'}^a$
2. The amount of computational resources required becomes infeasible as N grows
3. The Markov property is often not satisfied

For any given RL problem it is usually the case that one, or a combination, of these problems arises. We therefore need other methods to solve for the optimal value function.

2.1.2.3 Temporal Difference Learning and Q-Learning

Temporal Difference (TD) learning refers to a group of value-based solution methods that circumvent the problems mentioned above. Most importantly TD learning is model-free, meaning that it does not require knowledge of the environments dynamics ($P_{ss'}^a$ and $R_{ss'}^a$). It can also be applied at every time step meaning that it does not require episodic environments with a clear terminal state.

At its core TD learning samples from the environment and utilises the recursive nature of the Bellman equations to solve for the optimal value function. TD learning involves two key steps in order to solve for the optimal value function; policy evaluation and policy improvement. Policy evaluation involves calculating the value function for a given policy whereas policy improvement involves changing the current policy to obtain more reward given the current value function. TD learning alternates between these two steps in order to converge to the optimal policy.

Recall that the recursive definitions for each of the value functions are as follows:

$$V^\pi(s) = \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V^\pi(s')] \quad (2.36)$$

$$Q^\pi(s, a) = \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma \sum_{a'} \pi(s', a') Q^\pi(s', a')] \quad (2.37)$$

These can be re-written as expectations with respect to the environment's dynamics and the agent's policy to produce:

$$V^\pi(s) = \mathbb{E}[r_{t+1} + \gamma V^\pi(s_{t+1}) \mid s_t = s] \quad (2.38)$$

$$Q^\pi(s, a) = \mathbb{E}[r_{t+1} + \gamma Q^\pi(s_{t+1}, a') \mid s_t = s, a_t = a] \quad (2.39)$$

$$(2.40)$$

TD learning exploits the facts that these are expectations by sampling one step of the environment many times in order to obtain an average of the above expression and evaluate the current policy. The only quantities needed are the rewards experienced by the agent and the value estimate of the resulting state or state-action pair. One possibility would be to keep a true average of $r_{t+1} + \gamma V^\pi(s_{t+1})$ or $r_{t+1} + \gamma Q^\pi(s_{t+1}, a')$ for each state or state-action pair. However in practice this is a poor approximation because the agents policy is changing during learning and so the value estimates will also change. Instead TD learning relies on taking an exponentially weighted average, so that more recent values carry higher weights in the average calculation. If V_k is our estimate at time point k and R_k is our observed value at time k then the exponentially weighted average can be derived as follows:

$$V_k = V_{k-1} + \alpha[R_k - V_{k-1}] \quad (2.41)$$

$$= \alpha R_k + (1 - \alpha)V_{k-1} \quad (2.42)$$

$$= \alpha R_k + (1 - \alpha)\alpha R_{k-1} + (1 - \alpha)^2 V_{k-2} \quad (2.43)$$

$$= \alpha R_k + (1 - \alpha)\alpha R_{k-1} + (1 - \alpha)^2 R_{k-2} + \dots + (1 - \alpha)^{k-1} R_1 + (1 - \alpha)^k V_0 \quad (2.44)$$

$$= (1 - \alpha)^k V_0 + \sum_{i=1}^k \alpha (1 - \alpha)^{k-i} R_i \quad (2.45)$$

The TD learning update rules to evaluate the current policy therefore become:

$$V(s_t) = V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \quad (2.46)$$

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (2.47)$$

This allows for evaluation of the current policy after just a single step of the environment by observing outcomes and bootstrapping value estimates via the bellman equations.

Since TD learning is a value-based method, the policy is not represented directly. Instead policy improvement is typically achieved by taking actions that are chosen using an ϵ -greedy approach. The agent selects actions that are greedy with respect to the current value function but also takes a random action with probability ϵ in order to explore other options.

Using the update rule in Equation 2.47 to perform policy evaluation and an ϵ -greedy action selection method for policy improvement is commonly referred to as the SARSA algorithm. This method gets its name from the fact that it uses the tuple $(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1})$ to drive learning. The full algorithm can be seen in Algorithm 1.

Algorithm 1 SARSA

```

Initialize  $Q(s, a)$  at random
for each episode do
  Initialise starting state  $s_1$ 
  Choose  $a_1$  from  $s_1$  using policy based on  $Q$  e.g.  $\epsilon$ -greedy
  for  $t = 1$  to  $T - 1$  do
    Take action  $a_t$ , observe  $r_{t+1}$  and  $s_{t+1}$ 
    Choose  $a_{t+1}$  from  $s_{t+1}$  using policy based on  $Q$  e.g.  $\epsilon$ -greedy
     $Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$ 
  end for
end for

```

Another popular TD learning method is Q-learning, which only differs from SARSA by a single term in the policy evaluation step. Instead of using the action chosen by the agent a_{t+1} for the target value $Q(s_{t+1}, a_{t+1})$, Q-learning uses the best possible action $\max_a Q(s_{t+1}, a)$. The full update rule therefore becomes:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (2.48)$$

This update equation directly turns the equation for the optimal action-value function (Equation 2.31) into an update rule. Q-learning is known as an ‘off-policy’ learning method because the policy used by the agent to select actions (e.g. ϵ -greedy) is different from the one being evaluated (greedy). In comparison, SARSA

is an ‘on-policy’ method because both policies are the same (ϵ -greedy).

It is worth noting that Temporal Difference methods do not have to exclusively rely on sampling from a single step of the environment before updating their value estimates. The approach can be extended to sample many steps of the environment before the value estimate is bootstrapped and the update rule applied. How many steps to take is usually a bias-variance trade-off. Fewer steps introduces bias from the regular bootstrapping of values but many steps introduces variance from the wide range of outcomes experienced.

2.1.2.4 Policy-Based Methods

Policy-based methods focus on learning the optimal policy directly without representing a value function. Typically this requires that the policy is parameterised by some learn-able parameters θ . One major advantage of this approach is that the agent can learn policies that are stochastic and/or involve continuous action spaces. In order to update the parameters θ and learn the optimal value function, policy-based methods require an objective function that describes how good a given policy π_θ is i.e. how much reward it achieves. Depending on whether the environment is episodic or continuous there are several options for such an objective function. For example, in an episodic environment the objective function may be the average reward achieved from the starting state to the end of the episode. In contrast, in continuous environments the objective function may be the average value of all states or the average reward obtained per time-step. Armed with an appropriate objective function, finding the optimal policy is a standard optimisation problem that is amenable to both gradient-based (e.g. gradient descent) and non-gradient-based (e.g. genetic algorithms) approaches.

2.1.2.5 Monte-Carlo Policy Gradient

Monte-Carlo Policy Gradient, otherwise known as REINFORCE, is an example of a policy-based method that relies upon stochastic gradient descent to find the best policy. Stochastic gradient descent calculates the partial derivative of the objective function ($J(\theta)$) with respect to each of the parameters θ of the parameterised policy π_θ . It then moves each of the parameters a small amount in the direction of the

gradient in order to find a local maximum of $J(\theta)$:

$$\nabla_{\theta} J(\theta) = \begin{pmatrix} \frac{\partial J(\theta)}{\partial \theta_1} \\ \vdots \\ \frac{\partial J(\theta)}{\partial \theta_n} \end{pmatrix} \quad (2.49)$$

$$\theta = \theta + \alpha \nabla_{\theta} J(\theta) \quad (2.50)$$

Where α is known as the step-size parameter or learning rate and dictates the magnitude of the parameter update. This process of calculating the partial derivatives and updating the policy parameters is repeated until a local maximum is reached. As mentioned previously, the goal of RL is to maximise the expected return R_t for all possible states. We can therefore write our objective function as the average reward obtained from the MDP given our policy π :

$$J(\theta) = \sum_{s \in S} d^{\pi}(s) V^{\pi}(s) \quad (2.51)$$

$$= \sum_{s \in S} d^{\pi}(s) \sum_{a \in A} \pi_{\theta}(a | s) Q^{\pi}(s, a) \quad (2.52)$$

Where $d^{\pi}(s)$ is known as a stationary distribution of the MDP given our policy π . One way to intuitively think about this stationary distribution is if you were to travel through the MDP forever using the policy π then eventually the probability of ending up in a given state becomes unchanged.

The key component of a policy-gradient method (i.e. a method that uses the gradient to optimise $J(\theta)$) is calculating the vector of partial derivatives of the objective function with respect to the policy's parameters ($\nabla_{\theta} J(\theta)$). At first look this appears difficult because the derivative depends on the stationary distribution ($d^{\pi}(s)$):

$$\nabla_{\theta} J(\theta) = \nabla_{\theta} \sum_{s \in S} d^{\pi}(s) \sum_{a \in A} Q^{\pi}(s, a) \pi_{\theta}(a | s) \quad (2.53)$$

However it can be shown that using a theorem known as the ‘policy gradient theorem’ that:

$$\nabla_{\theta} J(\theta) \propto \sum_{s \in \mathcal{S}} d^{\pi}(s) \sum_{a \in \mathcal{A}} Q^{\pi}(s, a) \nabla_{\theta} \pi_{\theta}(a | s) \quad (2.54)$$

This simplification allows the update to be expressed as an expectation and means that we can take samples from the environment in order to gain an approximation of the expectation:

$$\nabla_{\theta} J(\theta) \propto \sum_{s \in \mathcal{S}} d^{\pi}(s) \sum_{a \in \mathcal{A}} Q^{\pi}(s, a) \nabla_{\theta} \pi_{\theta}(a | s) \quad (2.55)$$

$$= \sum_{s \in \mathcal{S}} d^{\pi}(s) \sum_{a \in \mathcal{A}} \pi_{\theta}(a | s) Q^{\pi}(s, a) \frac{\nabla_{\theta} \pi_{\theta}(a | s)}{\pi_{\theta}(a | s)} \quad (2.56)$$

$$= \mathbb{E}_{\pi} \left[Q^{\pi}(s, a) \frac{\nabla_{\theta} \pi_{\theta}(a | s)}{\pi_{\theta}(a | s)} \right] \quad (2.57)$$

$$= \mathbb{E}_{\pi} [Q^{\pi}(s, a) \nabla_{\theta} \ln \pi_{\theta}(a | s)] \quad (2.58)$$

Our final parameter update therefore becomes:

$$\theta = \theta + \alpha \nabla_{\theta} J(\theta) \quad (2.59)$$

$$= \theta + \alpha \mathbb{E}_{\pi} [Q^{\pi}(s, a) \nabla_{\theta} \ln \pi_{\theta}(a | s)] \quad (2.60)$$

Intuitively this update rule can be seen as changing the policy's parameters so that actions with a higher value become more likely than those with a lower value. The $\ln \pi_{\theta}(a | s)$ term is important because sub-optimal actions may be more likely given the current policy and so we need to control for the fact that they will be updated more often than other actions.

The Monte-Carlo component of Monte-Carlo Policy Gradient comes from the fact that $Q^{\pi}(s, a)$ can be estimated using Monte-Carlo methods. More specifically Monte-Carlo methods only work in episodic environments and rely on sampling a whole episode to obtain an estimate of the return and an unbiased sample of the value of each action taken. A general outline of the Monte-Carlo Policy Gradient algorithm can be seen in Algorithm 2.

Algorithm 2 Monte-Carlo Policy Gradient (REINFORCE)

Initialize parameters of policy θ at random
for each episode $s_1, a_1, r_2, \dots, s_{T-1}, a_{T-1}, r_T$ sampled using π_θ **do**
 for $t = 1$ to $T - 1$ **do**
 Calculate return R_t
 $\theta = \theta + \alpha R_t \nabla_\theta \ln \pi_\theta(a_t | s_t)$
 end for
end for

Monte-Carlo Policy Gradient is just one example of a policy gradient method. Many different techniques exist for calculating estimates of $Q^\pi(s, a)$. These other techniques typically involve learning the value function directly as well so that it can be used in the policy update. These techniques are known as actor-critic methods and will be covered in the next section. The main benefit to using actor-critic methods is that the policy-gradient method can be used in non-episodic environments where a Monte-Carlo approach is not possible.

2.1.2.6 Actor-Critic Methods

Actor-critic methods combine value-based and policy-based approaches by learning both a value function and a policy directly. This allows for the benefits of policy-based approaches, such as stochastic or continuous policies, while using the learnt value function to reduce the variance of the policy updates. The policy is typically referred to as the actor because it is used to select actions whereas the value function is referred to as the critic because it is used to critique the actions taken by the actor.

In Section 2.1.2.5 we saw that the policy could be updated using the following update rule:

$$\theta = \theta + \alpha \nabla_\theta J(\theta) \tag{2.61}$$

$$= \theta + \alpha \mathbb{E}_\pi [Q^\pi(s, a) \nabla_\theta \ln \pi_\theta(a | s)] \tag{2.62}$$

In the case of Monte-Carlo Policy Gradient or REINFORCE, $Q^\pi(s, a)$ is calculated as the full return from state s and action a for a given episode. This estimate of the value of state s and action a typically has high variance because the results of different episodes can be wildly different due to the stochastic nature of the MDP

and potentially the policy. There is also the issue that the full return can only be sampled when there are clear episodes and longer episodes will lead to higher variance. Actor-Critic methods circumvent this problem by using the value estimates from the critic to approximate $Q^\pi(s, a)$, thereby greatly reducing the variance. It is often common to parameterise both the value function and the policy in actor-critic methods. Algorithm 3 provides a general outline of such an approach.

Algorithm 3 Actor-Critic

Initialize parameters of policy θ and value-function w at random
 Sample starting state s_1 and action $a_1 \sim \pi_\theta(a | s)$
for $t = 1$ to $T - 1$ **do**
 Sample reward r_{t+1} and next state s_{t+1}
 Sample next action from policy $a_{t+1} \sim \pi_\theta(a | s_{t+1})$
 Update policy parameters
 $\theta = \theta + \alpha_\theta Q_w(s_t, a_t) \nabla_\theta \ln \pi_\theta(a_t | s_t)$
 Compute TD error
 $\delta_t = r_t + \gamma Q_w(s_{t+1}, a_{t+1}) - Q_w(s_t, a_t)$
 Update value function parameters
 $w = w + \alpha_w \delta_t \nabla_w Q_w(s_t, a_t)$
end for

While using a value function to estimate $Q^\pi(s, a)$ can greatly reduce variance in the parameter update it can also be reduced further. For example all actions from a given state may lead to a high amount of reward and so the parameter update will be relatively uninformative. However one action may lead to slightly more reward than the others and so we want this information to be captured in the weight update. For this reason it is often better to use the quantity $Q(s, a) - V(s)$ rather than just $Q(s, a)$ to perform the weight update. $Q(s, a) - V(s)$ tells us how much better an action is compared to the current policy. This serves to provide greater signal to noise ratio and captures how good an action is relative to the others. $Q(s, a) - V(s)$ is often called the advantage function and is represented as $A(s, a)$. Importantly this still works from a theoretical standpoint because $V(s)$ does not depend on the action being taken.

2.1.2.7 Model-Based Methods

The solution methods mentioned so far are known as model-free methods because they do not rely on knowledge of the environments dynamics. More specifically,

they do not explicitly use knowledge of the transition function $P_{ss'}^a$ or the reward function $R_{ss'}^a$. Model-based methods however, either have access to these functions or they attempt to learn them in order to solve the RL problem. Parallels are often drawn between model-based methods and the act of ‘planning’ because the agent has explicit knowledge of the consequences of its actions and can use this to select actions. Model-based RL algorithms therefore rely on the fact that they can simulate experience of the environment. A naive approach to planning would be to perform an exhaustive search over all possible states and actions from the agents current state using its model of the environment. This requires no value function to be stored. However for most interesting RL problems this is infeasible as the number of possible branches can be vast and no terminal state may exist.

For this reason, model-based RL algorithms generally revolve around using simulated experience to compute value functions, which can then be used to improve the current policy in an iterative manner. Some of the first model-based RL algorithms proposed, known as policy and value iteration, relied on dynamic programming to iteratively solve the MDP based on the recursive nature of the Bellman equations. By repeatedly sweeping over all the states in the MDP these methods could use the bellman equations to calculate the value of each state given the current policy. The estimated value function would then be used to improve the policy in a greedy manner. This process would be repeated until convergence. However, one problem with this approach is that it involves iterating over all states, which can be computationally infeasible if the state space is large. One alternative is to just sample from the agents model of the environment in order to improve the estimate of the agents current value function. An example of this is the Dyna-Q architecture which uses real experience in the environment to learn a value function and a model of the environment. The model of the environment is then sampled from periodically to produce simulated experience in order to further improve the estimate of the value function. For example, one way to sample the simulated experience is to use the current policy to generate trajectories likely to be experienced by the agent and then update the associated value estimates.

Both dynamic programming and Dyna-Q are referred to as *background planning* algorithms because they use a model of the environment to improve value estimates,

which are then ultimately used to make a decision. In comparison, planning can also be used directly to choose an action a_t at the current state s_t . This is known as *decision-time planning* and involves using the model of the environment to perform a look-ahead search in order to choose the best action. One example of this would be a heuristic search to find the best action from a given state. The agent can perform a heuristic search to some target depth k and then use the resulting rewards experienced to choose the best action. Importantly background planning and decision-time planning algorithms do not have to be mutually exclusive. For example, heuristic search can use the rewards obtained up to k and the approximate value function at the leaf nodes to select an action. The results of the heuristic search can then also be used to improve the underlying value function. These examples of model-based RL demonstrate how knowledge of the environment can be useful for both learning value functions and for considering the effects of one's actions based on the current state.

2.2 Deep Learning

The following description of Deep Learning (DL) draws from the work of Goodfellow et al. (2016).

2.2.1 Basic Principles of Neural Networks

Neural networks are a computational model for processing information based on a network of interconnected units. Each unit in the network has a value that represents its level of activation and a set of values that correspond to the weights of the connections between itself and other units. The activation value of a unit is calculated by applying a non-linear function to the dot product between the activation values of the units it is connected to and the associated weights:

$$a_j = h\left(\sum_{i=1}^N w_{ji}a_i + b_j\right) \quad (2.63)$$

Where a_j is the activation value of unit j , $h(z)$ is a non-linear function, w_{ji} is the connection from unit j to unit i and b_j is the bias of unit j . The bias represents

a scalar value that allows for shifts in the activation value which are independent of the input. importantly, each time this equation is performed the unit applies a non-linear transformation to the values of the units it is connected to, which allows the neural network to learn a non-linear mapping between its input and output.

In their simplest form, neural networks are organised into a series of layers, whereby each unit is only connected to units in the previous layer (Figure 2.2). Input is presented to the network by setting the values of the first layer to the values of the input. The activation values of the next layer are then calculated using Equation 2.63 for each unit in the layer. This process is repeated sequentially until the activation values of the final layer have been calculated and this is taken as the output of the network. Such an architecture is referred to as a feed-forward neural network because information is passed through the network layer-by-layer with no backwards information flow. Deep Learning (DL) refers to the use of neural networks that consist of several hidden layers. A hidden layer is defined as a layer of units that is in between the input and output layers. The term ‘hidden’ is used because the values of these layers are not used when reading the output of the network. By using multiple hidden layers, Deep Neural Networks (DNNs) are able to learn hierarchical representations that become increasingly abstract. Most importantly, DNNs are often referred to as universal function approximators because they are able to learn any continuous function between two Euclidean Spaces. This means that they can learn smooth non-linear functions, which imbues them with strong generalization capabilities even for highly non-linear relationships between variables.

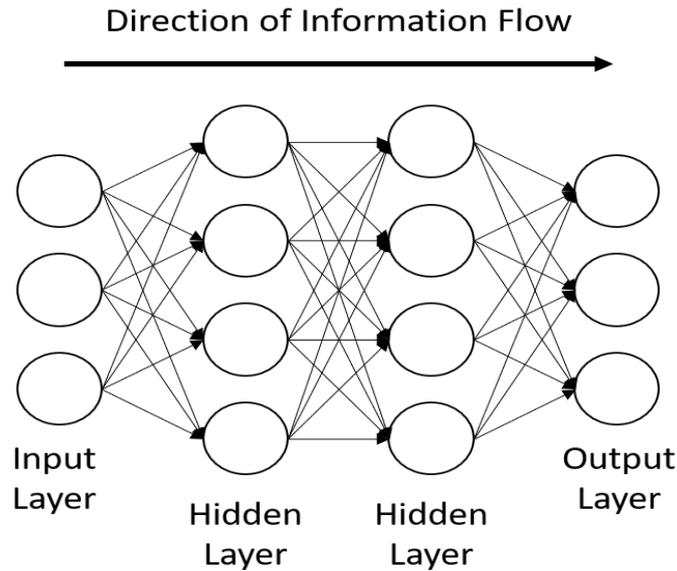


Figure 2.2: *Diagram of a feed-forward deep neural network. The network is described as feed-forward because each unit is only connected to units from the previous layer. This ensures that information flows from left to right. The network is also described as deep because it consists of more than one hidden layer.*

2.2.2 Learning in Deep Neural Networks

Learning is performed in neural networks by updating the values of the weights between units. This ultimately leads to a change in the input-output mapping of the network. A plethora of methods exist for updating the weights of a neural network. However, by far the most common in the field of Deep Learning (DL) is an approach known as backpropagation. Backpropagation works by incrementally updating the weights of a network in order to minimise or maximise an objective function. The overall effect of this is that the network gradually learns to represent a desirable input-output mapping. The objective function can be seen as a way of evaluating how well the neural network is performing on a given task.

2.2.2.1 Objective Functions

The first step in training a Deep Neural Network (DNN) via backpropagation is to decide on an objective function. As is commonly the case in machine learning, we want to maximise the likelihood of the data given our model. A common objective function is therefore the negative log-likelihood, which needs to be minimised in order to maximise the likelihood of the data. In terms of training a neural network

to predict an output variable \mathbf{y} given an input variable \mathbf{x} , the objective function can be written as follows:

$$J(\theta) = -\mathbb{E}_{x,y \sim \hat{p}_{data}} \ln p_{model}(\mathbf{y} \mid \mathbf{x}) \quad (2.64)$$

Where θ is the network weights, $\mathbb{E}_{x,y \sim \hat{p}_{data}}$ is the expectation with respect to the data distribution \hat{p}_{data} , and p_{model} is the distribution defined by our neural network. From this expression we can see that finding the minimum of the negative log-likelihood is the same as minimizing the cross entropy between \hat{p}_{data} and p_{model} . Armed with this general expression for the objective function, a more specific form can be devised depending on how we represent $p_{model}(\mathbf{y} \mid \mathbf{x})$, which will be influenced by the nature of y . For example, for a regression problem, $p_{model}(\mathbf{y} \mid \mathbf{x})$ can be represented as a gaussian distribution with the mean being the output of our network given x , and the variance being an arbitrary value σ^2 :

$$p_{model}(\mathbf{y} \mid \mathbf{x}) \sim \mathcal{N}(\mathbf{y}; f(\mathbf{x}, \theta), \sigma^2) \quad (2.65)$$

Given N input ($\mathbf{X} = \{x_1, \dots, x_N\}$) and output ($\mathbf{Y} = \{y_1, \dots, y_N\}$) pairs and assuming all data points are independent and identically distributed, minimisation of the negative log-likelihood therefore becomes:

$$\begin{aligned} J(\theta) &= -\ln p_{model}(\mathbf{Y} \mid \mathbf{X}, \theta, \sigma^2) \\ &= -\ln \prod_{i=1}^N \mathcal{N}(t_i \mid y(x_i, \theta), \sigma^2) \\ &= -\ln \prod_{i=1}^N \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(t_i - y(x_i, \theta))^2} \\ &= -\sum_{i=1}^N \ln \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) - \frac{1}{2\sigma^2} (t_i - y(x_i, \theta))^2 \\ &= -\sum_{i=1}^N -\frac{1}{2\sigma^2} (t_i - y(x_i, \theta))^2 \\ &= \sum_{i=1}^N (y_i - f(x_i, \theta))^2 \end{aligned} \quad (2.66)$$

The objective function therefore becomes the squared difference between the

data and the output of the network summed over all data points. Importantly, for a regression problem the output of the network needs to be able to cover the sequence of all possible real numbers. For this reason the output units just output the weighted sum of their inputs without applying a non-linear function. This is the same as setting the function $h(z)$ in Equation 2.63 to the identity function i.e. $y = x$.

The same logic can be applied to other problems such as classification, where $p_{model}(\mathbf{y} \mid \mathbf{x})$ is represented as a Bernoulli distribution and the activation function of the output units is set to the logistic function ($\frac{1}{1+e^{-x}}$) to correspond to $p(y = 1 \mid x)$. Ultimately, the nature of the output variable \mathbf{y} dictates the form of the objective function and also the activation function of the output units.

2.2.2.2 Backpropagation

With the objective function identified, backpropagation can update the weights of the network in order to minimise or maximise it and therefore improve performance. Backpropagation does this via gradient-based optimisation. Closed form solutions for the network weights are not possible because neural networks generally produce non-convex objective functions. This means that there may be several minima, maxima or saddle points. For example, neural networks can produce equivalent models by swapping or scaling parameters, meaning a specific input-output mapping can be represented by multiple sets of network weights.

Due to the non-convex optimisation problem posed by neural networks, there is no guarantee that backpropagation will find the global minima/maxima. Any solution will be sensitive to the values that the network weights are initialised to. In practice this is not a major problem because even a local minima represents a solution that performs well for a given task. A lot of research has gone into improving backpropagation in an attempt to overcome the problems associated with non-convex optimisation. However, for the purposes of this thesis, we shall cover the standard back propagation algorithm, which provides the main intuition for training DNNs.

The key quantity for gradient-based optimisation is the partial derivative of the objective function with respect to a model parameter i.e. $\frac{\partial J(\theta)}{\partial \theta_i}$. In the case of

neural networks we need to calculate the partial derivative of the objective function with respect to each of the weights in the network. Backpropagation allows us to sequentially calculate the partial derivatives for each layer of weights, starting from the output layer and moving back to the input layer.

Lets start by calculating the partial derivative of the object function with respect to the weights of the output units. We shall assume that we are using the objective function in Equation 2.66 and we have a three layer neural network. The activation function of each unit will be a logistic function except from the single output unit, which uses the identity function. To keep things clean we shall omit the sum over data points and assume we are using a single training example. We shall use \mathbf{W} to denote all the weights in the network, and $w_{ji}^{(3)}$ to denote the weight of the i^{th} input to the j^{th} unit in layer 3. Finally we will use $net_j^{(3)}$ to denote the net input into the j^{th} unit before the activation function is applied (i.e. the dot product between the inputs and the weights). Using the chain rule from calculus we get the following:

$$\begin{aligned} \frac{\partial J(\mathbf{W})}{\partial w_{ji}^{(3)}} &= \frac{\partial J(\mathbf{W})}{\partial net_j^{(3)}} \frac{\partial net_j^{(3)}}{\partial w_{ji}^{(3)}} \\ &= \frac{\partial J(\mathbf{W})}{\partial net_j^{(3)}} x_{ji}^{(3)} \\ &= \frac{\partial J(\mathbf{W})}{\partial net_j^{(3)}} a_i^{(2)} \end{aligned} \tag{2.67}$$

Where $x_{ji}^{(3)}$ is the i^{th} input into the j^{th} unit in layer 3, which is equivalent to the activation value of the i^{th} unit in layer 2. Now we need to evaluate the partial derivate of our objective function with respect to the net input to the j^{th} unit in layer 3. Lets call this quantity $\delta_j^{(3)}$ and again use the chain rule to evaluate it:

$$\begin{aligned} \delta_j^{(3)} &= \frac{\partial J(\mathbf{W})}{\partial net_j^{(3)}} \\ &= \frac{\partial J(\mathbf{W})}{\partial a_j^{(3)}} \frac{\partial a_j^{(3)}}{\partial net_j^{(3)}} \\ &= a_j^{(3)} - y_j \end{aligned} \tag{2.68}$$

Putting this all together we now have an expression for the partial derivative of our objective function with respect to the weights of the final output layer:

$$\begin{aligned}
\frac{\partial J(\mathbf{W})}{\partial w_{ji}^{(3)}} &= \frac{\partial J(\mathbf{W})}{\partial net_j^{(3)}} \frac{\partial net_j^{(3)}}{\partial w_{ji}^{(3)}} \\
&= \delta_j^{(3)} a_i^{(2)} \\
&= (a_j^{(3)} - y_j) a_i^{(2)}
\end{aligned} \tag{2.69}$$

To perform gradient descent on the weights of the final output layer and incrementally minimise the objective function we apply the following update rule:

$$\begin{aligned}
w_{ji}^{(3)} &= w_{ji}^{(3)} - \Delta w_{ji}^{(3)} \\
&= w_{ji}^{(3)} - \alpha \frac{\partial J(\mathbf{W})}{\partial w_{ji}^{(3)}} \\
&= w_{ji}^{(3)} - \alpha ((a_j^{(3)} - y_j) a_i^{(2)})
\end{aligned} \tag{2.70}$$

Next we need to calculate the partial derivatives for the earlier layers. Notice how changing the values of weights in the earlier layers will impact all subsequent values. Backpropagation accounts for this by propagating the error values back through the network. To see how this works we shall calculate the partial derivative for a weight of a unit in layer 2:

$$\begin{aligned}
\frac{\partial J(\mathbf{W})}{\partial w_{ik}^{(2)}} &= \frac{\partial J(\mathbf{W})}{\partial net_i^{(2)}} \frac{\partial net_i^{(2)}}{\partial w_{ik}^{(2)}} \\
&= \frac{\partial J(\mathbf{W})}{\partial net_i^{(2)}} x_{ik}^{(2)} \\
&= \frac{\partial J(\mathbf{W})}{\partial net_i^{(2)}} a_k^{(1)} \\
&= \delta_i^{(2)} a_k^{(1)}
\end{aligned} \tag{2.71}$$

This is exactly the same approach as before, however the key difference is in the evaluation of $\frac{\partial J(\mathbf{W})}{\partial net_i^{(2)}}$ or $\delta_i^{(2)}$. $\delta_i^{(2)}$ will be different to $\delta_j^{(3)}$ because it describes how the objective function changes as a result of changing the net input into the i^{th} unit in

layer 2 rather than the j^{th} unit in layer 3. To evaluate $\delta_i^{(2)}$ we do the following:

$$\begin{aligned}\delta_i^{(2)} &= \frac{\partial J(\mathbf{W})}{\partial net_i^{(2)}} \\ &= \sum_{j \in \text{downstream}(i)} \frac{\partial J(\mathbf{W})}{\partial net_j^{(3)}} \frac{\partial net_j^{(3)}}{\partial net_i^{(2)}}\end{aligned}\tag{2.72}$$

Where $\text{downstream}(i)$ is the set of all units that are connected immediately downstream of the i^{th} unit in layer 2. If layer 2 and layer 3 were fully connected then this would contain all the units in layer 3. We have already calculated $\frac{\partial J(\mathbf{W})}{\partial net_j^{(3)}}$ for all $j \in \text{downstream}(i)$ because these are just the $\delta_j^{(3)}$ values from the final output layer. Simplifying the above expression we eventually get:

$$\begin{aligned}\delta_i^{(2)} &= \sum_{j \in \text{downstream}(i)} \frac{\partial J(\mathbf{W})}{\partial net_j^{(3)}} \frac{\partial net_j^{(3)}}{\partial net_i^{(2)}} \\ &= \sum_{j \in \text{downstream}(i)} \delta_j^{(3)} \frac{\partial net_j^{(3)}}{\partial net_i^{(2)}} \\ &= \sum_{j \in \text{downstream}(i)} \delta_j^{(3)} \frac{\partial net_j^{(3)}}{\partial a_i^{(2)}} \frac{\partial a_i^{(2)}}{\partial net_i^{(2)}} \\ &= \sum_{j \in \text{downstream}(i)} \delta_j^{(3)} w_{ji}^{(3)} a_i^{(2)} (1 - a_i^{(2)}) \\ &= a_i^{(2)} (1 - a_i^{(2)}) \sum_{j \in \text{downstream}(i)} \delta_j^{(3)} w_{ji}^{(3)}\end{aligned}\tag{2.73}$$

Where $a_i^{(2)}(1 - a_i^{(2)})$ is the derivative of the logistic activation function of unit i . These equations demonstrate how the δ values can be propagated backwards from the output layer to the input layer in order to calculate the partial derivatives for gradient-based optimisation. In general, the backpropagation rule can be written as follows:

$$\delta_u^{(n)} = \begin{cases} (a_u^{(n)} - y_u) & n = \text{number of layers} \\ \frac{\partial a_u^{(n)}}{\partial net_u^{(n)}} \sum_{d \in \text{downstream}(u)} \delta_d^{(n+1)} w_{du}^{(n+1)} & \text{otherwise} \end{cases}\tag{2.74}$$

$$\frac{\partial J(\mathbf{W})}{\partial w_{um}^{(n)}} = a_m^{n-1} \delta_u^{(n)} \quad (2.75)$$

For the first layer i.e. when $n = 1$, $a_m^{(0)}$ is just taken to be the m^{th} entry of our input variable \mathbf{x} . These partial derivatives can then be used, as in Equation 2.70, to incrementally update all the weights of the network and therefore minimise the objective function.

2.2.3 Deep Learning Architectures

Within the domain of Deep Learning (DL) a range of different neural network architectures exist beyond just feed-forward networks. The choice of network architecture typically depends on the nature of the problem that the neural network is trying to solve. For the purposes of this thesis we shall briefly cover three main architectures; deep convolutional neural networks, long short-term memory networks and autoencoders. These architectures form the backbone of much of the work in this thesis and each fulfill a distinct role.

2.2.3.1 Deep Convolutional Neural Networks

The term Deep Convolutional Neural Network (DCNN) is commonly used to refer to a neural network that uses a number of convolutional layers to process its input, which is typically in the form of an image (Figure 2.3). Convolutional layers are loosely modelled on the visual cortex of the brain, in that they rely on spatial invariance to learn features of images. They achieve this by using feature maps, where all the units share the same weights but process a different spatial region of the input.

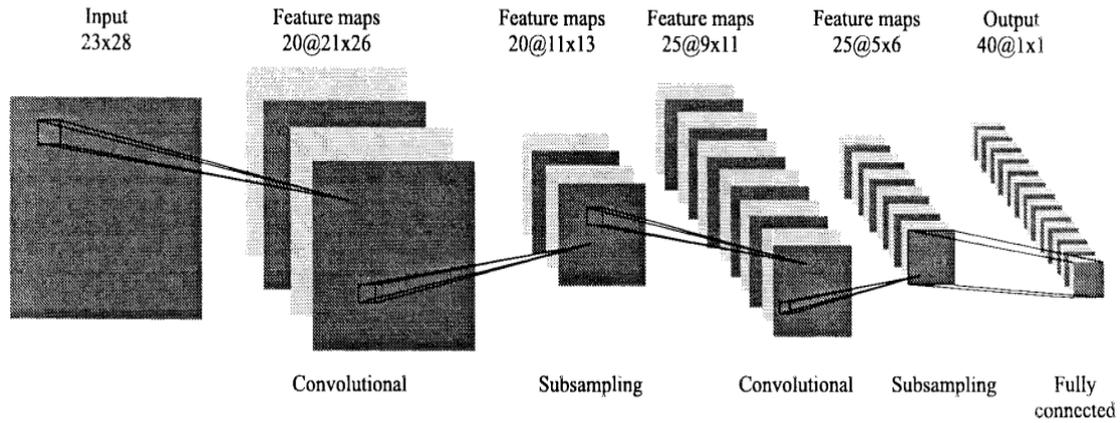


Figure 2.3: *Depiction of a Deep Convolutional Neural Network (DCNN). The input is commonly an image, which goes through a series of convolutional layers and subsampling layers (e.g. max-pooling). The output of the network is produced by a final fully connected layer. Figure adapted from Lawrence et al. (1997)*

Figure 2.4 shows a depiction of a single convolutional layer. The layer consists of N feature maps, with each map being made up of a 2D array of units. Within a single feature map all the units share the same weight values, which means that all the units learn to represent the same feature. However, each unit applies these weight values to a different region of the input, which results in a convolution. The region that a unit processes corresponds to the unit's receptive field and the receptive fields are staggered to produce a tiling of the input. The network can therefore detect a feature regardless of its spatial location making it spatially invariant.

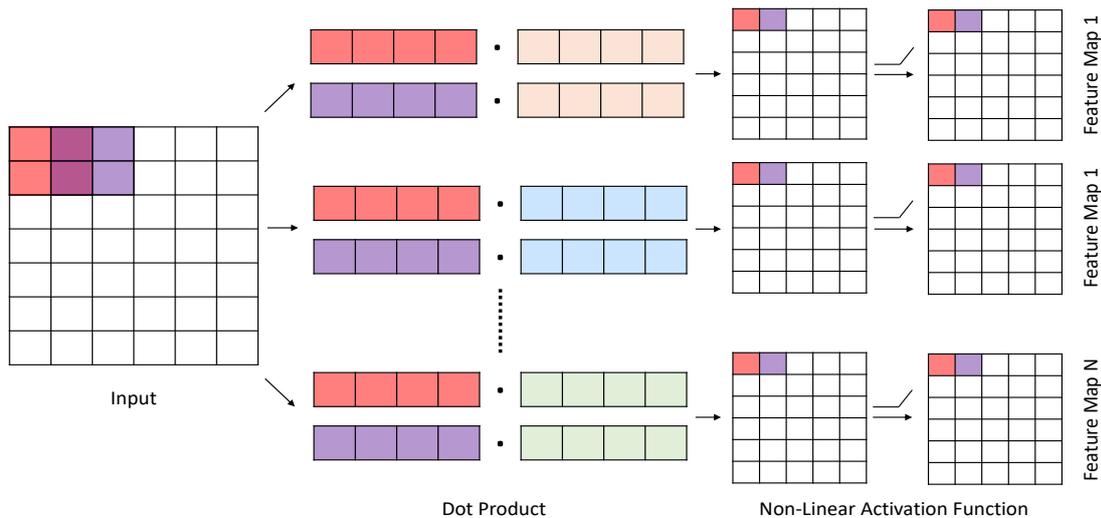


Figure 2.4: *Depiction of a convolutional layer with an input that only has one channel. Each feature map consists of a 2D array of units that all share the same weight values (shown as orange, blue or green vectors). Each unit within a feature map has a different receptive field (two examples are shown in red and purple). Activation values are calculated by applying a non-linear activation function (e.g. $\max(0, x)$) to the dot product between the unit's weights and receptive field.*

The weights of each unit are 3-dimensional because they have a height, width and depth. The width and height are hyper-parameters that define the size of the unit's receptive field, while the depth matches the number of channels in the input. If the input to the convolutional layer was an image, then the depth would be 3 because there is a red, green and blue channel. In contrast, if the input was another convolutional layer, then the depth would be the number of feature maps in the previous layer. Another important hyper-parameter is the 'stride' length, which determines how finely the input is tiled by the convolutional layer. For instance, a larger stride length would result in the receptive fields of each unit being farther apart.

Typically a DCNN while stack many convolutional layers on top of each other to produce increasingly abstract features of the input while maintaining information about spatial location. The results of the last convolutional layer are then reshaped into a single vector and passed to a standard neural network layer to produce the overall output of the network. Another common approach is to perform sub-sampling by sporadically including max-pooling layers between convolutional

layers (Figure 2.3). Max-pooling layers partition the feature maps of the convolutional layer into non-overlapping regions and then output the maximum value for each region. This serves to decrease the width and height of the feature maps in the convolutional layer, which can speed up learning by reducing the dimensionality of the problem.

In summary, DCNNs represent a powerful approach for learning spatial invariant features of their input. Each feature map corresponds to a different feature of the input and they maintain information about where in the input the feature may be present. By sharing weights across units within a feature map, the number of learnable weights is greatly reduced, which can greatly improve the speed and quality of the learnt solution.

2.2.3.2 Long Short-Term Memory Networks

Long Short-Term Memory networks (LSTMs) are a specific form of recurrent neural network. A recurrent neural network is defined as a network where connections exist within layers and/or to previous layers. This is important because it provides the network with a form of memory as values from the previous time step interact with the input presented on the current time step. As a result the input is processed in a contextual manner, which can be important for problems that involve temporal dependencies.

LSTMs are a specific form of recurrent neural network that rely on the gating of information between successive time steps. The general idea of gating is that a set of learnable parameters are responsible for whether a unit's activation value is maintained or altered at each time step. This allows the recurrent neural network to remember aspects of the input for long time-scales, which is particularly important for problems involving long-range dependencies.

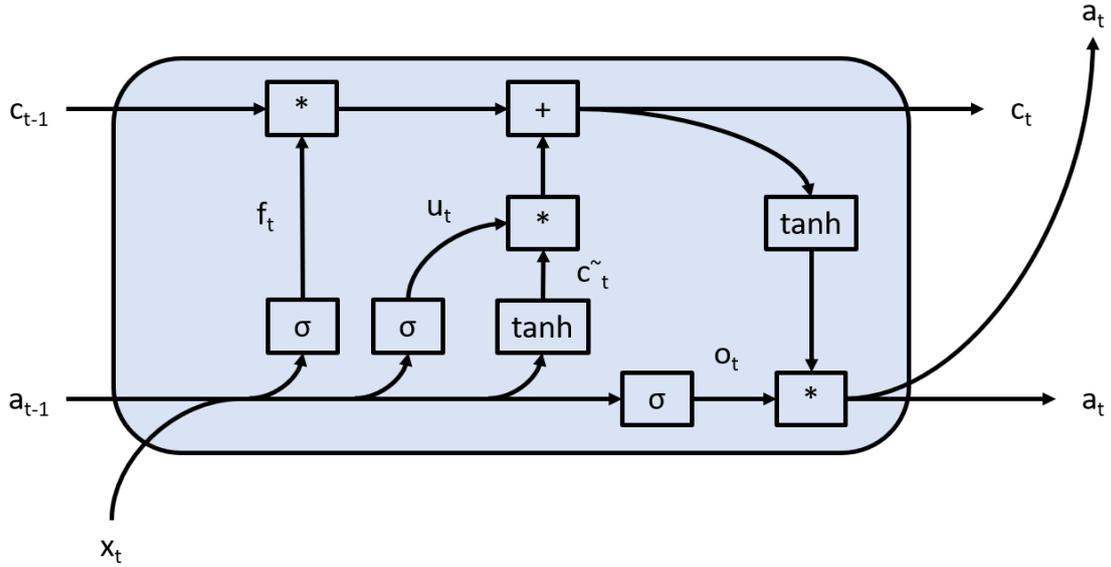


Figure 2.5: *Depiction of a Long Short-Term Memory (LSTM) network. Each unit has a hidden state value (c_t) and an activation value (a_t). The hidden state is updated based on a weighted sum of the previous hidden state (c_{t-1}) and a new candidate value (\tilde{c}_t). The weights (f_t and u_t) and the new candidate value (\tilde{c}_t) are produced by applying separate learnable weights to the previous activation values (a_{t-1}) and the current input values (x_t). The updated hidden state is used to calculate a new activation value. The output of this new activation value is gated by another weight (o_t), which is also learnt by applying learnable weights to the previous activation values (a_{t-1}) and the current input values (x_t).*

Figure 2.5 shows the general architecture of an LSTM. Each unit of the LSTM has a hidden state value (c_t) and an activation value (a_t). The hidden states of the LSTM network are updated by taking the weighted sum of the previous hidden states (c_{t-1}) and a new set of candidate values based on the current input (\tilde{c}_t):

$$\mathbf{c}_t = \mathbf{f}_t * \mathbf{c}_{t-1} + \mathbf{u}_t * \tilde{\mathbf{c}}_t \quad (2.76)$$

Where \mathbf{c} , \mathbf{f} , \mathbf{u} and $\tilde{\mathbf{c}}$ are all vectors with each entry corresponding to a unit in the network. The weights \mathbf{f}_t and \mathbf{u}_t are seen as gates that control how much of the previous hidden state value and the new candidate value are used respectively. The weights \mathbf{f}_t and \mathbf{u}_t , as well as the new candidate value $\tilde{\mathbf{c}}_t$, are all calculated by applying their own set of learnable weights to the previous steps activation values (\mathbf{a}_{t-1}) and the current input values (\mathbf{x}_t):

$$\begin{aligned}
\tilde{\mathbf{c}}_t &= g(\mathbf{W}_c \begin{bmatrix} \mathbf{a}_{t-1} \\ \mathbf{x}_t \end{bmatrix} + \mathbf{b}_c) \\
\mathbf{u}_t &= \sigma(\mathbf{W}_u \begin{bmatrix} \mathbf{a}_{t-1} \\ \mathbf{x}_t \end{bmatrix} + \mathbf{b}_u) \\
\mathbf{f}_t &= \sigma(\mathbf{W}_f \begin{bmatrix} \mathbf{a}_{t-1} \\ \mathbf{x}_t \end{bmatrix} + \mathbf{b}_f)
\end{aligned} \tag{2.77}$$

Where the function $g(z)$ is the *tanh* function and $\sigma(z)$ is the *logistic* function. \mathbf{W}_c , \mathbf{W}_u and \mathbf{W}_f are matrices that represent the weights of each unit for the hidden states and inputs. $\begin{bmatrix} \mathbf{a}_{t-1} \\ \mathbf{x}_t \end{bmatrix}$ is a vector containing the hidden states of the previous time step and the inputs of the current time step. \mathbf{b}_c , \mathbf{b}_u and \mathbf{b}_f are vectors containing the bias values for each unit. Once the new hidden state values (\mathbf{c}_t) have been calculated using Equation 2.76, the new activation values are calculated by applying a non-linear function to the new hidden state values and multiplying the result by another set of gating values (\mathbf{o}_t):

$$\mathbf{a}_t = \mathbf{o}_t * g(\mathbf{c}_t) \tag{2.78}$$

\mathbf{o}_t therefore represents an ‘output’ gate, which dictates how much of the new hidden state is output by the unit. \mathbf{o}_t is calculated in the same way as the previous gates but with its own set of learnable parameters:

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \begin{bmatrix} \mathbf{a}_{t-1} \\ \mathbf{x}_t \end{bmatrix} + \mathbf{b}_o) \tag{2.79}$$

By using these equations, an LSTM can learn to use information from many time steps ago to produce predictions. When it comes to training an LSTM, backpropagation can still be used in the normal way. The only difference is that the errors are backpropagated to the previous time steps and the partial derivatives for each time step are summed to produce the final weight update. This is often referred to as backpropagation through time and is usually fixed to a certain number of time steps if the number of previous time steps is extremely large.

2.2.3.3 Autoencoders

Autoencoders are different to LSTMs and DCNNs in that they do not correspond to a particular neural network architecture. Instead an autoencoder refers to a method of training whereby the neural network is trained to re-create its input. This is a form of unsupervised learning because no labels are required for the input data. In the case of an autoencoder, the training objective from Equation 2.64 becomes:

$$J(\theta) = -\mathbb{E}_{x \sim \hat{p}_{data}} \ln p_{model}(\mathbf{x} | \mathbf{x}) \quad (2.80)$$

This means that an autoencoder can consist of DCNNs or LSTMs so long as the input variable is the same as the output variable. Due to this property, autoencoders are of interest not because of their output predictions, but because of their hidden layer representations. By re-creating the input data, the activation values of the hidden layers represent a latent encoding of the input data, which can often be designed to have desirable properties. Typically the latent representation is read from the middle hidden layer with the preceding portion of the network referred to as the ‘encoder’ and the subsequent portion referred to as the ‘decoder’ (Figure 2.6).

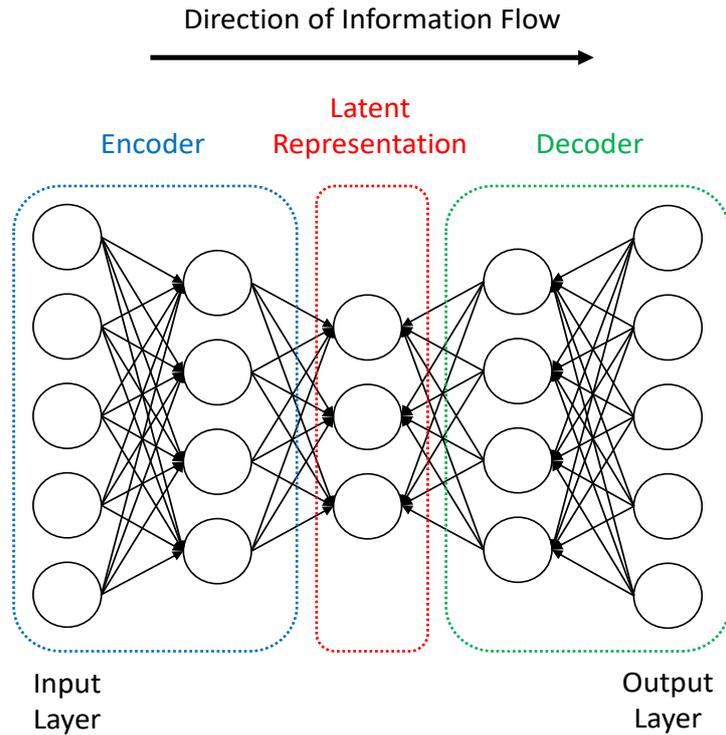


Figure 2.6: *Depiction of an autoencoder. The network is trained to re-create its input at the output layer. The middle hidden layer is taken as the latent representation of the input. The first half of the network is referred to as the encoder and the second half is referred to as the decoder. The number of units in the middle layer is typically less than the number of input units to enforce dimensionality reduction.*

The most common desirable property of the learnt latent representation is to have less dimensions than the input data. Dimensionality reduction is often desirable because it helps to alleviate the curse of dimensionality and remove spurious noise. This is achieved in an autoencoder by setting the number of units in the middle hidden layer to be less than the number of dimensions of the input data (i.e. the number of input units) (Figure 2.6). This also helps to prevent the network from just remembering individual exemplars because there is an information bottleneck and so the network needs to learn the underlying factors of variation.

2.3 Deep Reinforcement Learning

Originally Reinforcement Learning (RL) solution methods relied upon tabular approaches i.e. a table of values for each state or state-action pair. More sophisticated approaches used fixed basis functions or linear functions to approximate the value function and/or policy. However none of these approaches were able to deal with

high-dimensional states and the *curse of dimensionality*. Recently approaches have started to use Deep Learning (DL) to approximate the key functions required in RL, a technique known as Deep Reinforcement Learning (Deep RL) (François-Lavet et al., 2018). Deep RL is particularly powerful because it utilises the universal function approximation abilities of deep neural networks to either represent the value function and/or policy. This allows the value function and/or policy to be a complex non-linear function of the states and actions. This non-linear function also has generalization properties because the deep neural networks are able to interpolate between data points in a smooth manner. In addition, Deep neural networks learn hierarchical representations with increasing levels of abstraction, which helps to overcome the *cure of dimensionality*. The fact that deep neural networks *learn* these representations is particularly important because it means they can be tailored to the task at hand without the need for being hand-designed. The deep neural networks used in Deep RL are typically trained using the backpropagation algorithm and often rely on the Temporal Difference (TD) error if learning a value function or the policy gradient if learning a policy.

2.3.1 Deep Q-Learning

One of the seminal studies demonstrating the capabilities of Deep Reinforcement Learning (Deep RL) to deal with complex high-dimensional states was conducted by Mnih et al. (2015). In the study Mnih et al. (2015) applied an approach known as a Deep Q-Network (DQN) to an array of Atari 2600 video games. Strikingly DQN was able to reach or exceed human level performance on 29 of the 46 Atari games. DQN was able to achieve this by combining Q-Learning with deep convolutional neural networks to convert raw pixel values into meaningful actions. Most importantly, the same network architecture and hyper-parameter values were used for each video game demonstrating the the DQN could learn useful state representations on its own without the need for game-specific information.

Mnih et al. (2015) made several implementation decisions that ensured the success of the DQN approach. Firstly each frame of the game was resized to 84 X 84 pixels to reduce computational costs. DQN was then provided the last four frames as the current state of the environment so that the total input was 84 X 84 X 4.

This was important because it provided DQN with information about the direction that objects were moving in and helped to make the state more Markovian. From an architectural standpoint, DQN used three convolutional layers followed by one fully connected hidden layer and finally an output layer (Figure 2.7). As DQN relied upon Q-Learning and therefore the calculation of state-action values, the output layer consisted of 18 units which corresponded to one unit for each action in the Atari games. The output of each of these units was taken as the state-action value or Q-Value for the corresponding action. With the state and action space defined, the final step was to define the reward function. Fortunately the Atari game environment has a relatively natural reward function; the score achieved between two time-steps. This also allowed the reward signal to be standardised across games.

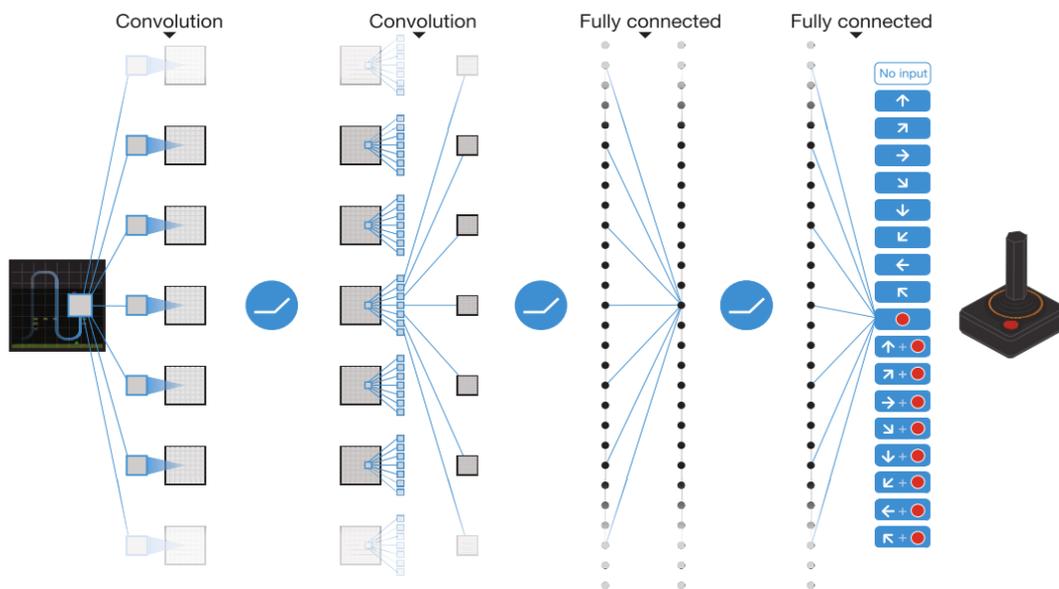


Figure 2.7: Architecture of the Deep Q-Network (DQN) (Mnih et al., 2015). The last four game frames were provided as input to the network. The network consisted of three convolutional layers followed by two fully connected layers. The output layer contained 18 units, which output the value of each of the possible actions available in the video games. All units apart from the output units used a Rectified Linear Unit (ReLU) activation function, whereby the input value was set to 0 if it was less than 0. The output units used a linear activation function. Figure adapted from Mnih et al. (2015).

For training DQN used an ϵ -greedy approach, with the value of ϵ decreasing linearly over the first one million game frames. This helped to encourage exploration early on and then settle on a policy later. DQN was allowed to interact with each game for 50 million frame, which equates to around 38 days of continuous game-

play. Throughout this time DQN would observe the last four frames, select an action, receive a reward signal and the next game frame, and repeat. The weights of the deep neural network were updated using backpropagation and the Q-Learning TD error as the objective function:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha(r_{t+1} + \gamma \max_a \hat{Q}(s_{t+1}, a; \mathbf{w}_t) - \hat{Q}(s_t, a_t; \mathbf{w}_t)) \nabla \hat{Q}(s_t, a_t; \mathbf{w}_t) \quad (2.81)$$

Where \mathbf{w}_t is the vector of weights of the deep neural network at time t and $\nabla \hat{Q}(s_t, a_t; \mathbf{w}_t)$ is the partial derivative as calculated by backpropagation. Rather than applying these updates after each action Mnih et al. (2015) accumulated a history of one-step transitions $(s_t, a_t, r_{t+1}, s_{t+1})$ and applied the updates intermittently using batches of data randomly sampled from this history. This approach was known as an *experience replay* and was beneficial for several reasons. Firstly, because the updates were applied in a batch they helped to provide a lower-variance sample of the actual gradients for training. Secondly, the transitions were sampled randomly, which removed any correlations between the updates. Finally, it made more efficient use of the transitions experienced by the agent.

Aside from experience replay, Mnih et al. (2015) used one final trick to help improve the stability of the algorithm. TD methods such as Q-Learning rely on bootstrapping value estimates in order to generate target values to be used in the update rule. This can be seen in Equation 2.81 where the target value is the sum of the reward experienced (r_{t+1}) and the value of the next state-action pair as predicted by the deep neural network ($\max_a \hat{Q}(s_{t+1}, a; \mathbf{w}_t)$). The problem with this is that the target value is a function of the weights being updated, which can lead to instability and divergence. To solve this problem Mnih et al. (2015) would take a copy of the deep neural network every C weight updates and freeze the values of the weights in that network. This ‘target’ network was then used exclusively to calculate the target value for the Q-Learning weight updates. This approach allowed for bootstrapping of values while keep the target values relatively stable during training. With \tilde{Q} denoting the target network, the final weight update used by Mnih et al. (2015) became:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha(r_{t+1} + \gamma \max_a \tilde{Q}(s_{t+1}, a; \mathbf{w}_t) - \hat{Q}(s_t, a_t; \mathbf{w}_t)) \nabla \hat{Q}(s_t, a_t; \mathbf{w}_t) \quad (2.82)$$

With these straight forward modifications Mnih et al. (2015) was able to demonstrate for the first time the power of combining deep neural networks with reinforcement learning techniques. For the first time, hand-crafted state features were not required as DQN could learn from scratch useful representations of high-dimensional inputs. Despite this achievement DQN still required that the weights of the network were reset before learning a new game. This highlights that DQN learns very task-specific representations and that further work is required to design agents that can learn a range of tasks simultaneously.

2.3.2 Advantage Actor-Critic (A2C)

Arguably the most popular family of Deep RL algorithms are the advantage actor-critic (A2C) algorithms (Mnih et al., 2016). As mentioned in Section XXX, these algorithms learn both a value function and policy, either using two separate Deep Neural Networks (DNNs) or a single one with two output layers. The critic typically learns the state value function using the Temporal Difference (TD) error:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha(r_{t+1} + \gamma V(s_{t+1}; \mathbf{w}_t) - V(s_t; \mathbf{w}_t)) \nabla_{\mathbf{w}} V(s_t; \mathbf{w}_t) \quad (2.83)$$

This approximation is then used to calculate the advantage of an action a_t from a given state s_t . After an action a_t is taken from state s_t the resulting reward r_{t+1} and state s_{t+1} is observed. These values can then be used to calculate the ‘advantage’ of the action using the following formula:

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t) \quad (2.84)$$

$$= (r_{t+1} + \gamma V(s_{t+1}; \mathbf{w}_t)) - V(s_t; \mathbf{w}_t) \quad (2.85)$$

This advantage value is subsequently used to update the policy network via the policy gradient:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha A(s_t, a_t) \nabla_{\boldsymbol{\theta}} \ln \pi(a_t | s_t; \boldsymbol{\theta}_t) \quad (2.86)$$

The main advantage of these methods is that they can be used to learn continuous and stochastic policies. In comparison approaches such as DQN simply learn action values for discrete actions. The development of A2C algorithms therefore opened up the use of Deep RL to more domains and applications.

2.4 Summary

In this chapter we have reviewed the necessary background material to understand the basic mechanisms underlying Deep Reinforcement Learning (RL). RL provides a framework for learning how to select actions that maximise reward based on the state of the environment. Meanwhile Deep Learning (DL) provides a mechanism for learning hierarchical representations that can support the approximation of continuous non-linear functions. By combining RL with DL, Deep RL agents are able to learn complex tasks involving high-dimensional inputs, such as playing video games from raw pixels. In the next chapter we shall explore how Complementary Learning Systems (CLS) theory from cognitive science can be used to understand efficient RL in the brain and simultaneously improve the efficiency of Deep RL systems.

Chapter 3

CLS Theory as the Basis for Efficient RL

Overview

The purpose of this chapter is to review past literature in order to reconcile Deep Reinforcement Learning (RL) with Complementary Learning Systems (CLS) theory (McClelland et al., 1995). This will allow us to identify the learning systems that support efficient RL in the brain and that are lacking in Deep RL algorithms. In particular, we focus on the computational properties of these learning systems and how they interact with each other to support the rapid learning of new information and the transfer of past information. We start by using the analogy between Deep RL and CLS theory to identify three key pathways in the brain that may contribute to its ability to perform rapid learning and transfer (Section 3.1). These pathways include connections between (1) the neocortex and the striatum, (2) the hippocampus and the striatum and (3) the neocortex and the hippocampus. For each pathway we review recent advancements in Deep RL and computational modelling to highlight how these pathways may support efficient RL (Sections 3.2 - 3.4). Our hope is that this chapter demonstrates the utility of an analogy between Deep RL and CLS theory for understanding rapid learning and transfer in the human brain.

3.1 CLS Theory and Deep RL

The human brain is a complex network of different learning systems and it is their interactions that support human level intelligence. One theory that is central to this thesis is Complementary Learning Systems (CLS) theory (McClelland et al., 1995; O’Reilly and Norman, 2002; Kumaran et al., 2016). The premise of CLS theory is that the brain relies on two different forms of memory to learn and make decisions (Figure 3.1). The first form of memory, often referred to as semantic memory, is thought to occur in neocortical areas and relies upon the slow learning of regularities across multiple experiences. These regularities are thought to be encoded by overlapping representations, which provides semantic memory with good generalisation abilities. In comparison, the second form of memory, often referred to as episodic memory, is thought to reside in the hippocampus and is responsible for rapidly remembering the specifics of individual experiences. These specifics are encoded via pattern-separated representations, which ensures that episodic memory can recall very similar experiences without interference. It has been proposed that one of the functions of episodic memory in the hippocampus is to replay individual experiences to the neocortex in an interleaved fashion to facilitate the neocortex’s gradual semantic learning (McClelland et al., 1995).

These two forms of memory are therefore complementary; the neocortex slowly learns distributed representations of multiple experiences while the hippocampus rapidly learns pattern-separated representations of individual experiences. With respect to Reinforcement Learning (RL), both the neocortex and the hippocampus send projections to the striatum (Pennartz et al., 2011), which is thought to evaluate states and/or actions (Schultz, 1998; Houk et al., 1995; Joel et al., 2002; Maia, 2009; Setlow et al., 2003). It is therefore plausible that the RL machinery uses both semantic and episodic memory to inform decisions. With this in mind, we identify three key pathways within the CLS framework that may support efficient RL in the brain (Figure 3.1). Pathways 1 and 2 in Figure 3.1 correspond to the direct projections of the neocortex and hippocampus to the striatum respectively. These two pathways allow each learning system to directly influence the evaluation of states and actions. Pathway 3 corresponds to the connections between the neocortex and the hippocampus. This pathway allows the two learning systems to have an indirect

effect on RL by influencing the behaviour of the other system. We strongly believe that it is the interactions between these different pathways that allow for efficient RL through rapid learning and transfer.

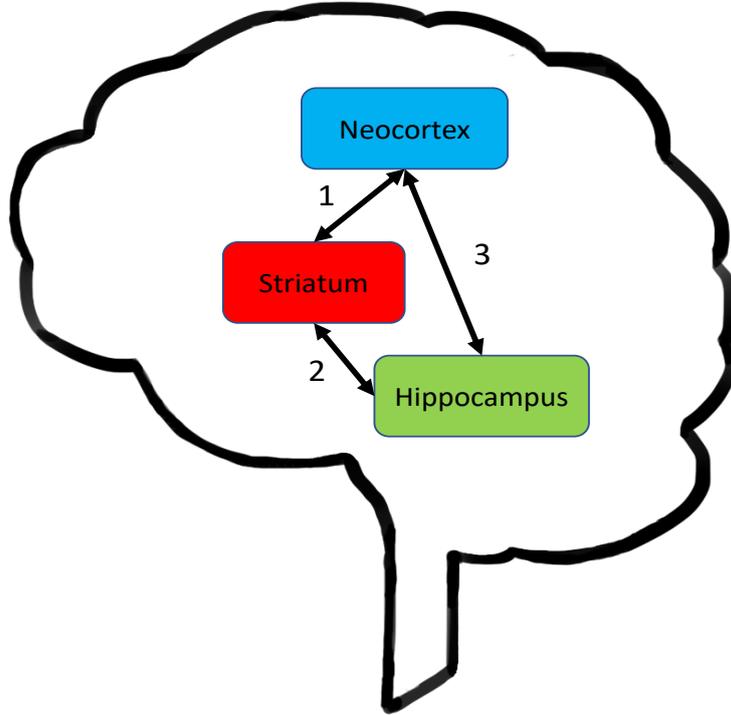


Figure 3.1: *Diagram reconciling Complementary Learning Systems (CLS) theory with Reinforcement Learning (RL). Three main pathways are identified; (1) projections from the neocortex to the striatum, (2) projections from the hippocampus to the striatum and (3) projections between the neocortex and the hippocampus.*

How does Deep RL fit into CLS theory and the pathways outlined in Figure 3.1? Deep RL (François-Lavet et al., 2018) uses Deep Neural Networks (DNNs) to learn continuous functions of the state space so that predictions can generalise to large and high-dimensional state spaces. The distributed and overlapping representations learnt by DNNs are crucial to their ability to generalise. Interestingly this property of DNNs appears to mimic those of the neocortex, which also relies on the slow learning of overlapping representations for generalisation. Indeed, the representations learnt by DNNs have been found to have a close similarity to those learnt by the neocortex (Güçlü and van Gerven, 2015; Yamins and DiCarlo, 2016). This suggests that classic Deep RL algorithms may be viewed as analogous to Pathway 1 in Figure 3.1, whereby overlapping distributed representations are used to evaluate states and actions. This analogy is central to the current thesis and suggests that Deep RL

algorithms should benefit from the addition of learning systems that correspond to the other pathways highlighted in Figure 3.1.

The remainder of this chapter will review recent advances in Deep RL and computational modelling that attempt to understand how the brain achieves efficient RL. These advances will be categorised based on which of the three pathways they relate to in Figure 3.1. This will help to emphasise the utility of a CLS framework and how efficient RL is likely to be supported by a plethora of interacting systems. We will start by exploring advancements that do not suggest the need for additional learning systems in Deep RL. These approaches can be seen as trying to improve the ability of Deep RL to replicate the interactions between the neocortex and the striatum and therefore the contributions of semantic memory to RL (Pathway 1 in Figure 3.1).

3.2 1. Connections Between the Neocortex and Striatum

The striatum is thought to be responsible for evaluating states and/or actions during Reinforcement Learning (RL) (Schultz, 1998; Houk et al., 1995; Joel et al., 2002; Maia, 2009; Setlow et al., 2003). Importantly, the striatum receives a multitude of inputs from the neocortex (Haber, 2016), which is thought to provide overlapping representations containing semantic knowledge for evaluation (McClelland et al., 1995). From a Complementary Learning Systems (CLS) perspective, it is logical that this pathway should be the primary candidate for transfer as it provides knowledge that is general and abstracted across many experiences. As mentioned in Section 3.1, the Deep Neural Networks (DNNs) used in Deep RL appear to share similar properties with the neocortex in that they slowly learn overlapping representations. The representations learnt by these DNNs are used to calculate the values of states or state-action pairs. This therefore seems to mimic the cortical-striatal pathway in the brain (Pathway 1 in Figure 3.1).

Despite this seemingly natural comparison between the use of DNNs for approximating value-functions and the cortical-striatal pathway in the brain, the resulting behaviour of the two is strikingly different. Typically Deep RL approaches that

utilise DNNs, such as DQN (Mnih et al., 2015), lack any ability to transfer or generalise as soon as the task changes. In comparison semantic knowledge in the cortex appears to be able to support behaviour in a wealth of different scenarios throughout life (Quillan, 1966). The question arises then, why are DNNs in Deep RL such a poor model of semantic memory and how can they be altered in order to better understand the properties of semantic memory and move closer to the efficiency of human RL? Ultimately this question reduces to investigating the nature and content of the representations learnt by DNNs and how they differ from those learnt by the neocortex.

3.2.1 Catastrophic Forgetting

One obvious difference between the DNNs used in Deep RL and semantic memory in the neocortex is that humans start a task with a wealth of prior experience. However, this wealth of past experience alone is not enough to explain the disparity between the efficiency of Deep RL and human RL. This is because prior experience is only useful if one has the ability to process those experiences into a form of semantic knowledge that can be re-used. One of the problems that stops DNNs from being able to do this is known as *catastrophic forgetting* (French, 1999). The term catastrophic forgetting is used to describe the fact that DNNs suffer from interference so that the learning of new information can disrupt previously learnt information. For this reason DNNs have to be repeatedly trained on a mixture of old and new information so that they do not forget the old information. As a result, for an approach such as DQN to capitalise on the years of experience that humans have and form representations that generalise across them, it would need to record all of the experiences ever encountered, which is computationally infeasible.

With this in mind, a wealth of approaches have been proposed to reduce the prevalence of catastrophic forgetting in DNNs. One particular group of approaches rely on applying constraints to the updates of the weights between units. More specifically, these approaches work by reducing the plasticity of weights i.e. the magnitude of updates, based on how important they were for previous tasks (Kirkpatrick et al., 2017; Zenke et al., 2017) (Figure 3.2). These approaches are of interest because they are inspired by theoretical neuroscience and work on synaptic plasticity

(Benna and Fusi, 2016), making them potentially biologically plausible candidates for reducing catastrophic forgetting. Another group of approaches takes inspiration from Complementary Learning Systems (CLS) theory to replay past experiences back to the network during training using generative models (Mocanu et al., 2016; Shin et al., 2017; van de Ven and Tolias, 2018). We shall cover these approaches in more detail in Section 3.4.2 as they involve Pathway 3 in Figure 3.1. In summary, by alleviating catastrophic forgetting, DNNs can be trained on multiple tasks without over-writing previous knowledge. This can lead to the learning of representations that are potentially useful across a range of tasks and that generalise well to new unseen tasks.

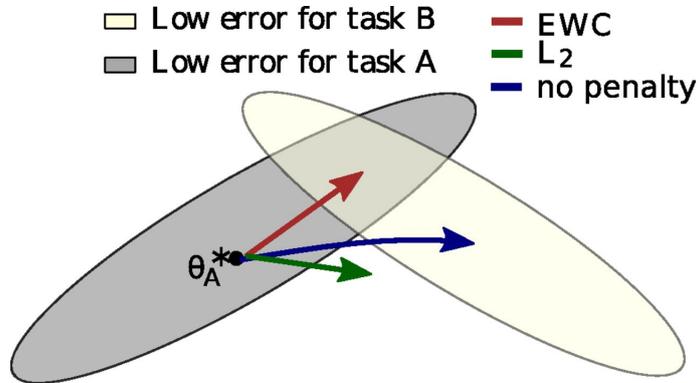


Figure 3.2: *Depiction of how weights are updated in Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017). Updating the weights based on the gradients calculated on task B will erase the learning on task A (blue arrow). In contrast, constraining the updates using a coefficient will potentially degrade performance on both tasks (green arrow). In comparison, EWC attempts to update weights in a way that improves performance on task B while maintaining performance on task A. Figure adapted from Kirkpatrick et al. (2017).*

3.2.2 Objective Functions

Even with the elimination of catastrophic forgetting it is still an open question how well the representations learnt in Deep RL generalise to new scenarios and whether they mimic the kinds of transfer displayed by humans. Testing such a hypothesis is difficult for two main reasons. Firstly, the algorithm would need to be exposed to a stream of multi-sensory data that is similar to the one experienced by a developing child in order to make a fair comparison. Secondly, and perhaps even more difficult, the algorithm would need to be trained to optimise similar objectives to that of a human being. The representations in a DNN are naturally affected by the task that

is being optimised for and so this has to be taken into account when comparing Deep RL models to human cognition. For the DNNs used in Deep RL to be a reliable model of semantic memory we need to understand what the main drivers are for learning representations in the neocortex.

Research in this area often trains DNNs on a specific objective function and then compares representations with known brain regions to quantify how likely that objective function is in the brain (Khaligh-Razavi and Kriegeskorte, 2014; Cadieu et al., 2014). Most of the success of this approach has come from training convolutional DNNs to classify natural images and comparing the learnt representations to those of the visual stream in the neocortex. The fact that there appears to be a similarity between them demonstrates that the brain may be optimising objective functions other than just the difference between predicted and actual reward, as is the case in Deep RL. Indeed recent modelling work in Deep RL has shown that imbuing a Deep RL agent with additional objective functions, such as learning how to maximally change pixels in the input, can greatly improve performance on single video games (Jaderberg et al., 2016).

From a theoretical point of view, it can be argued that if the number of tasks a person is faced with can potentially be infinite and ever changing, then the best approach would be for semantic memory to learn general features of the perceptual input that are task-agnostic. Indeed, this is the goal of unsupervised learning, which finds patterns in the input data without the explicit use of labels for a specific task. This is potentially useful because it allows one to learn the representations of underlying variations in the environment, which can then be picked and processed according to the task at hand (Bengio, 2012), rather than being tuned to any particular task. Such representations can be re-used across tasks and constitute transferrable knowledge. In the case of Deep RL, unsupervised learning is able to learn representations that may appear extraneous to achieving reward for the current task but that may be useful for obtaining reward in future tasks without the need for explicit labels. Interestingly, developmental research suggests that infants and young children rely largely on unsupervised learning in order to learn representations of their environment because they experience relatively few labels compared to the amount of data they are exposed to (Lake et al., 2009; Rosenthal et al., 2001).

However, we have already seen that representations in the visual stream appear to resemble the representations learnt by supervised DNNs categorizing natural images as opposed to unsupervised DNNs (Khaligh-Razavi and Kriegeskorte, 2014; Cadieu et al., 2014). In addition, adult perception can change with sufficient training on a specific task such as the sexing of chicks (Biederman and Shiffrar, 1987; Schyns et al., 1998), suggesting that our learnt perceptual representations can be altered by a specific task or goal. Finally, task-specific modulation of activity can be found in the neocortex during behavioural experiments (Woolgar et al., 2011; Kouider et al., 2016) as well as choice-related activity (Yang et al., 2016). It therefore seems unlikely that semantic knowledge is completely driven by unsupervised learning in a task agnostic manner. In reality semantic knowledge is likely the result of both supervised and unsupervised learning working in tandem to generate features that are both useful for the current task and that retain some degree of generality. However, it remains an open question which objective functions best capture the representations found in semantic memory and that lead to efficient RL.

As a final point on objective functions, it is worth noting that the problems faced by the neocortex and semantic memory are not static and consistent throughout life. A child is not immediately faced with learning to play video games but instead acquires simple motor, perceptual and language skills in the first few years of life (Gallahue et al., 2006; Gibson, 1969; Bloom and Lahey, 1978). These simple tasks, such as learning to segment objects (Spelke, 1990), are likely to serve as the basis for more complex behaviour and allow for the learning of representations that can be re-used throughout adult life. This developmental bootstrapping of knowledge is a powerful demonstration of transfer itself and it is this ability that serves as the basis for further more powerful transfer later in development. For example, in the field of analogy a theory known as progressive alignment (Gentner and Hoyos, 2017) states that children tend to start with relatively simple generalisations and this allows them to understand increasingly complex generalisations as they grow older. It is therefore not just the fact that humans encounter multiple tasks that encourage the learning of general representations but that these tasks are usually encountered in a specific order based on difficulty. Indeed even Skinner’s seminal studies on animal learning rely on increasing the difficulty of the learning problem in order to achieve the desired

performance on a final target task (Peterson, 2004), a process referred to as ‘shaping’. This concept is particularly pertinent when considering DNNs as a possible model of semantic memory because they also appear to benefit from shaping during training (Elman, 1993; Bengio et al., 2009; Krueger and Dayan, 2009). Most importantly the benefit appears to manifest itself as a form of regularisation, whereby performance is improved on the test set (Bengio et al., 2009) i.e. it improves transfer in DNNs. The benefits have also been demonstrated in RL problems, where the phenomenon is often referred to as ‘curriculum learning’ (Narvekar, 2017; Narvekar et al., 2017; Narvekar and Stone, 2018). It is therefore important that the correct objective functions are not only identified, but that they are also presented in the correct order to faithfully mimic human development.

If the DNNs in Deep RL are to capture the capabilities of semantic memory and represent the cortical-striatal pathway in the brain it is critical that they attempt to optimise similar problems to those faced by the brain. This likely involves an array of supervised and unsupervised learning tasks that lead to representations that generalise well to a range of tasks. In addition, the ordering of these tasks must be considered, so that semantic knowledge can be efficiently boot-strapped to enable increasingly complex behaviour and generalisation.

3.2.3 Disentangled Representations

The question of objective functions aside, the representations learnt by DNNs have intrinsic properties that can either help or hinder transfer. Work conducted by Higgins et al. (2016) suggests that representations learnt by DNNs should be ‘disentangled’ in order to promote transfer. The term ‘disentangled’ refers to the idea that the individual units of a neural network should encode independent factors of variation. For example one unit may encode the size of an object while another may encode its colour. Higgins et al. (2016) achieve disentangled representations by using a slightly modified Variational AutoEncoder (VAE).

An autoencoder is an unsupervised neural network that has to recreate its input at its output via successive layers of units. Typically an autoencoder contains a bottleneck, where the number of units in the middle layer is smaller than the dimensionality of the input data. This creates a latent representation with lower dimen-

the representations learnt by VAE's. $\beta - VAE$ was able to discover underlying latent factors of the input data that were understandable from a human perspective. However, more pertinent to this thesis was the fact that this disentanglement of representations also appeared to improve the flexibility of Deep RL approaches so that they could display the kinds of transfer often exhibited by humans.

In a further paper by Higgins et al. (2017), the authors combined the $\beta - VAE$ model with RL algorithms in order to explore the benefits of disentanglement for RL. In this study the $\beta - VAE$ was first used to generate representations of a virtual 3D environment. This virtual 3D environment consisted of a basic room that could have different combinations of objects (two of either a hat, cake, can or balloon) and coloured walls. The representations from $\beta - VAE$ were then used with a variety of different model-free RL algorithms to assess how useful they were for transfer, independent of the exact RL algorithm implementation. Higgins et al. (2017) called this combination of $\beta - VAE$ and RL algorithm a DisentAngled Representation Learning Agent (DARLA). The task of the agent in the 3D environment was to avoid hat and cake objects and to collect can and balloon objects. Importantly the RL algorithms were only trained on a specific subset of the object and wall combinations (Figure 3.4). This allowed them to be tested on the held out subset to see whether they could transfer their knowledge from the training set to the new combinations. DARLA was able to transfer its knowledge and could successfully collect either the can or balloon and avoid either the hat and cake for combinations of objects and wall colour that it had not been trained on. This is an impressive demonstration of transfer as DARLA is able to reason about which objects are rewarding even in contexts that it has not learnt from before e.g. a can is rewarding regardless of the room that it is in.

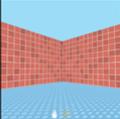
object room						
						
	D_S	D_S			D_U	D_U
	D_T	D_S	D_U	D_U		

Figure 3.4: The rooms and objects used to train and test Disentangled Representation Learning Agent (DARLA). D_U represents combinations that were used for visual pre-training to obtain disentangled representations from $\beta - VAE$. D_S represents combinations that were used to train the policy of the agent using reward feedback. D_T represents the test task, which involved a combination of objects and room that the agent had no prior experience of with respect to reward feedback. Figure adapted from Higgins et al. (2017).

One could argue that this example is not true transfer because the $\beta - VAE$ component of DARLA has seen all the objects and wall colours at some point. However this would be a harsh criticism as it is very rare that we as humans have a perceptual experience that is truly novel, apart from very soon after birth. In fact when we talk about perceptual novelty we are usually talking about novel combinations of perceptual features that we have little knowledge of how to act upon. Higgins et al. (2017) describe DARLA as first using the $\beta - VAE$ to learn how to ‘see’ and represent the world, and then using the RL algorithm to learn how to ‘act’ in the world. One can see how this might parallel the learning of an infant soon after birth, as it first has to learn how to represent the world before it can act effectively. The disentangled representations used by DARLA appear to be useful for transfer for two main reasons. The first reason is because, as with other unsupervised techniques, they provide a constant latent space for RL algorithms to learn a policy from. The second reason is because the main factors of variation are independently encoded in single units. This means that interference is greatly reduced during learning. For example, to evaluate the value of colour the network only has to learn the weights for a single unit rather than fitting weights across many units in order to learn a useful policy.

Quite how far the approach characterised by DARLA can go to replicating human like transfer in more complicated tasks, such as the video games often used to test DQN (Mnih et al., 2015), remains to be seen. One potential criticism that might be hard to address is the biological plausibility of disentangled representations. The key property of $\beta - VAE$ is that units encode independent factors of variation such as colour or size in a mutually exclusive fashion. If disentangled representations are utilised in semantic memory for transfer, then this would suggest that different regions of the cortex should be responsible for encoding independent factors of variation. Neuroscience research suggests that this may be the case, for example the classic model of primate visual cortex states that colour and shape are encoded separately (Livingstone and Hubel, 1988). However these findings are still contentious (Garg et al., 2019) and it is not straight forward to define what the independent factors of variation should be for the rich stream of perceptual data that is experienced by human beings. This being said, a recent study by Higgins et al. (2020) has shown a good correspondence between the representations learnt by $\beta - VAE$ when trained on images of faces and the response of single inferotemporal neurons in macaques. Furthermore Higgins et al. (2020) found that just from a handful of cell recordings they were able to use the decoding portion of $\beta - VAE$ to reconstruct images of the faces. $\beta - VAE$ therefore appears to represent a promising model of representations in the neocortex.

One criticism that could be made of $\beta - VAE$ as a model of representations in the neocortex is its extremely low robustness to damage. For instance, losing the unit that encodes colour would effectively result in us being unable to represent or ‘see’ colour. This is in contrast to numerous findings in neuroscience that suggest a high robustness to damage in many brain areas (Aerts et al., 2016). One potential counter argument to this could be that groups of units in the latent space of $\beta - VAE$ could encode each factor of variation rather than a single unit, which would greatly improve robustness to damage.

3.3 2. Connections Between the Hippocampus and Striatum

In Section 3.2 we saw how the Deep Neural Networks (DNNs) used in Deep Reinforcement Learning (RL) share similar properties to the cortical-striatal pathway found in the brain (Pathway 1) and that they can tentatively be compared to semantic memory. However, CLS theory states that the presence of episodic memory is also an important component for intelligent behaviour (McClelland et al., 1995). Episodic memory in the brain is thought to be the responsibility of the hippocampus (Burgess et al., 2002). Interestingly, the hippocampus projects to the striatum (Pennartz et al., 2011) indicating that RL in the brain relies on both semantic (Pathway 1) and episodic (Pathway 2) memory. Therefore in order to capture the complex RL behaviour demonstrated by humans, CLS theory suggests that Deep RL approaches should benefit from the addition of a ‘hippocampal’ learning system. Indeed, many theoretical advantages have been proposed for the use of hippocampal episodic information in RL. In particular, it has been suggested that episodic information can be used to approximate value functions, increase data efficiency and reconcile long-range dependencies (Gershman and Daw, 2017).

When the first rigorous mathematical treatment of RL was proposed by Sutton and Barto (1998), many of the solution methods relied upon learning tables of values, otherwise known as tabular approaches. Tabular approaches store a separate value for every state or state-action pair, which eliminates the potential for interference and allows for updates to be performed instantly. They therefore appear to share properties with a hippocampal learning system; both systems quickly learn pattern-separated values of individual states and/or actions. The main disadvantage of tabular approaches is that as the number of states and/or actions increases, they require more experience to encounter each action-value and more computational resources to store the associated values. The distributed representations of DNNs then become advantageous because they allow for efficient generalisation over the state space. In an ideal scenario a DNN would be responsible for generalisation over the state space while a tabular method would store pattern-separated memories that are crucial to behavior and that violate the generalisations of the network.

How does the brain make use of a hippocampal learning system for efficient RL? The remainder of this section will attempt to answer this question by reviewing recent advancements in Deep RL and computational modelling that explore the computational properties of the hippocampus and how they can support rapid learning and transfer in humans.

3.3.1 Fast Learning in the Hippocampus Supports Efficient Reinforcement Learning

One of the primary issues with relying on semantic knowledge for Reinforcement Learning (RL) is that updating of the knowledge is typically slow. As with Deep Neural Networks (DNNs), learning is supposedly slow and incremental in the neocortex, abstracting similarities over many experiences. For these reasons it is thought that the brain relies upon the fast learning of the hippocampus to quickly drive learning in novel situations. Recent work in Deep RL has supported this hypothesis by demonstrating the power of combining tabular approaches with DNNs (Botvinick et al., 2019).

Most notably Blundell et al. (2016) proposed an algorithm called ‘model-free episodic control’, which consisted of a table containing the maximum return (sum of discounted rewards) for each state-action pair experienced (Figure 3.5). Values in this table could be updated instantly supporting fast learning in new environments. The memory requirements for this table were kept constant by removing the least recently updated table entry once the size limit had been reached. Each observation from the environment was projected by a DNN-based embedding function (either a random projection or a variational autoencoder) to a state value and actions were selected based on a k-nearest neighbours method, which allowed for some degree of generalisation to novel states. Blundell et al. (2016) tested this approach on the Arcade Learning Environment (Atari) (Bellemare et al., 2013) and Labyrinth (Mnih et al., 2016), which both require the use of visual information to learn an optimal policy. The results of these simulations showed that model-free episodic control was significantly more data efficient than other classical Deep RL approaches, suggesting that episodic information is indeed important for fast learning.

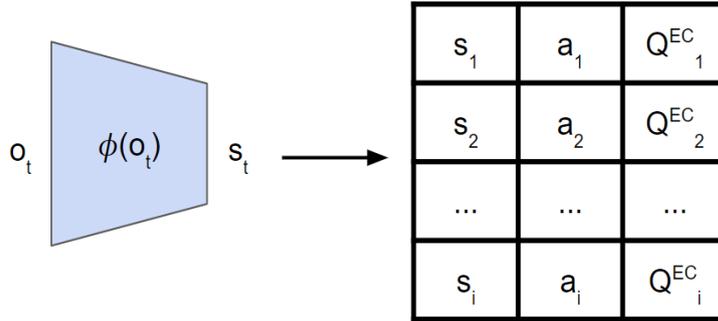


Figure 3.5: *Depiction of model-free episodic control (Blundell et al., 2016). The agent observes the environment at time t and transforms the observation (o_t) into a latent state (s_t) using a deep neural network. A table is used to store the maximum return experienced from separate state-action pairings. Actions are then selected using a k -nearest neighbours approach based on the state values in the table.*

While taking a first step towards highlighting the benefits of a ‘hippocampal’ learning system that utilises fast learning of pattern-separated information, the work of Blundell et al. (2016) has several notable drawbacks. Firstly, the table recorded the maximum return from any given episode and used this to inform the policy of the agent. This naturally cannot handle stochastic environments, where the expected return is the important quantity and not the maximum return of an individual episode. Secondly, this approach is likely to be highly inflexible. For example if a state-action pair suddenly becomes highly aversive then the entry in the table will not be updated because only the maximum value is stored. A third criticism is that the approach relies on the full return for each state-action pair and this is only possible when the task has distinct finite episodes. Some of these criticisms have been addressed in subsequent work, for example Pritzel et al. (2017) propose a fully differentiable version of ‘model-free episodic control’ that learns the embedding function in an online fashion using N -step Q-learning.

The above issues notwithstanding, what is most pertinent to this thesis is that ‘model-free episodic control’, and its various derivatives, do not rely on two complementary learning systems that operate in parallel to evaluate actions. The DNN-based embedding function may be tentatively compared to a neocortical learning system (as seen in Section 3.2) but it operates before the hippocampal learning system and as a result only the output of the hippocampal learning system is used to evaluate action values. This means that any advantages that may be conferred from

the additional predictions of a neocortical learning system are lost. In essence, the aforementioned approaches cannot arbitrate between the predictions of a neocortical and a ‘hippocampal’ learning system, but are instead restricted to using episodic predictions. This is inconsistent with the finding that the striatum receives inputs from both cortical areas and the hippocampus and needs to arbitrate between the two (Pennartz et al., 2011).

3.3.2 Recurrent Similarity Computation

From a Complementary Learning Systems (CLS) perspective, generalisation is the primary function of the neocortex due to its slow learning of overlapping representations that form over multiple experiences (McClelland et al., 1995). However, this line of reasoning struggles to explain how people are able to generalise a newly learnt piece of information in a matter of minutes. We are therefore left with the hippocampal system as a potential source of generalisation and transfer for shorter time scales. This proposal is hard to reconcile with CLS theory because the hippocampus is thought to rely on non-overlapping representations, which try to separate memories as much as possible in order to avoid interference. However, work by Kumaran and McClelland (2012) has provided insights into how non-overlapping representations may still be able to support generalisation over short time scales.

Kumaran and McClelland (2012) proposed a computational model called Recurrency and Episodic Memory Results in Generalization (REMERGE), which implemented a potential mechanism for generalisation in the hippocampus using non-overlapping hippocampal representations (Figure 3.6). The model took a connectionist approach and had two key architectural components: a ‘feature’ layer and a ‘conjunctive’ layer that were connected to each other using bi-directional connections. The ‘feature’ layer encoded the input to the model as a distributed representation across units while the ‘conjunctive’ layer used individual units to encode conjunctions of features as non-overlapping episodic representations. The strength of the connections between the units of these two layers were set according to the features present in a given conjunction e.g. a conjunction unit for features A, B and D would have strong connections to the corresponding feature units A, B and D. Importantly, this architecture allowed REMERGE to perform an operation Kumaran

and McClelland (2012) termed ‘recurrent similarity computation’. During recurrent similarity computation, input is presented to the feature layer and propagated to units in the conjunctive layer, which in turn activate other units in the feature layer. This process is repeated for a set number of cycles and the overall result is that the activity of the conjunctive units reflect their similarity to both externally presented input and input reconstructed by the network.

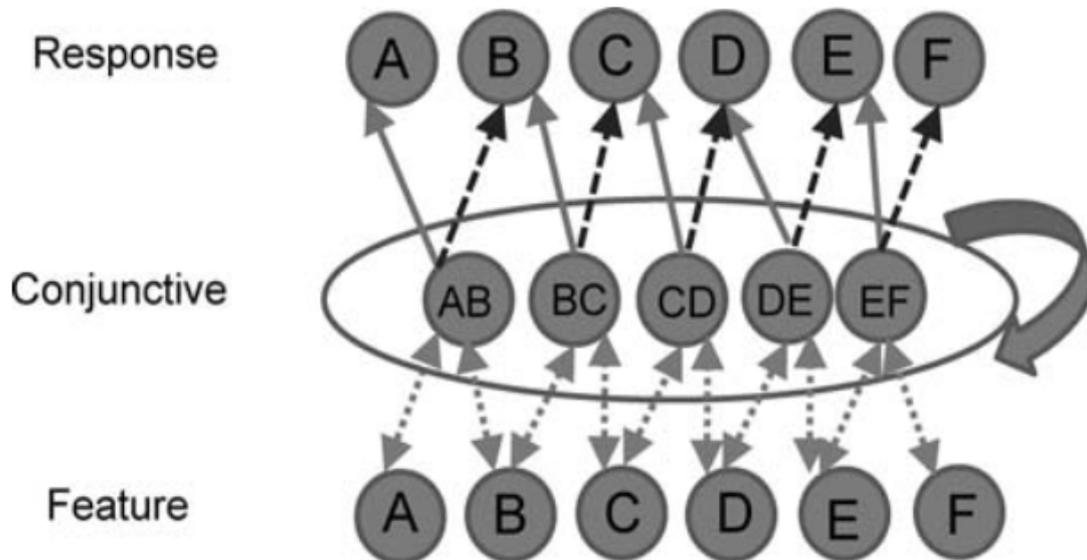


Figure 3.6: *Architecture of Recurrency and Episodic Memory Results in Generalization (REMERGE) for the transitive inference task. Input is presented at the feature layer, where each unit corresponds to a different stimulus (A-F). Each unit in the conjunctive layer corresponds to a stimulus pairing that was used during training (e.g. AB). Each unit in the response layer corresponds to a different choice made by the network (A-F). Connections between the feature and conjunctive layer are excitatory and bi-directional. Connections between the conjunctive layer and the response layer are uni-directional and either excitatory (solid arrow) or inhibitory (dashed arrow) to represent the learnt transitive relationship. The curved arrow next to the conjunctive layer denotes a softmax operation that causes competitive inhibition between units in the conjunctive layer. Figure adapted from Kumaran and McClelland (2012).*

Kumaran and McClelland (2012) showed that REMERGE is able to solve an array of classical generalisation tasks such as Transitive Inference, Paired Associative Inference and Acquired Equivalence tasks. It therefore appears that the hippocampal system may be a good candidate for fast, flexible transfer as it is not restricted by the slow interleaved learning of the neocortex and can acquire episodic memories quickly for recurrent similarity computation. However, while REMERGE appears

to be able to account for transfer behaviour on Transitive Inference, Paired Associative Inference and Acquired Equivalence tasks, it is unclear how its mechanism of recurrent similarity computation would apply to the types of RL problems typically tackled by Deep RL systems.

3.3.3 Relational Representations and Cognitive Maps

The work by Kumaran and McClelland (2012) demonstrates how the computational properties of the hippocampus can allow for specific forms of quick inference that the RL machinery can then act upon. However, it remains an open question how ‘recurrent similarity computation’ could be extended to tasks other than Transitive Inference, Paired Associative Inference and Acquired Equivalence tasks. One property these tasks have in common is that they involve relationships between items. For example, the Transitive Inference task involves a linear relational structure between items; A is greater than B, which is greater than C, which is greater than D, etc. It is largely thought that the ability to reason about such relationships is fundamental to making useful inferences in novel situations (Gentner, 1988; Wilson et al., 1985; Cook and Wasserman, 2007; Torrey, 2009; Holyoak, 2012). Indeed, an entire field of reinforcement learning known as relational RL has been suggested as a potential solution to the transfer problem (Van Otterlo, 2005; Džeroski et al., 2001). The basic premise behind relational RL is that if incoming perceptual information is represented in terms of abstract relationships between objects then reinforcement learning should be highly flexible because the representations are not tied to particular perceptual instances.

The idea of using relational representations for transfer has some interesting connections to developmental work on transfer. In particular, the fact that children shift from perceptual matches to relational matches and that transfer appears to be dependent on domain knowledge suggests that the learning of such representations may be a key component of adult level transfer (Gentner and Hoyos, 2017). In fact, the theory of progressive alignment suggests that this may be a reciprocal relationship, whereby the learning of relational representations supports transfer, which then bootstraps the learning of further relational representations.

Interestingly, recent work has highlighted the hippocampus as a potential source

of such relational representations in the brain. For many years neuroscientists have focused on the role of the hippocampus in spatial reasoning. As a result, researchers have described a multitude of different cell types in the hippocampus and entorhinal cortex that are involved in spatial reasoning. The most famous of these cells are place cells in the hippocampus that only fire in one location in an environment (O'keefe and Nadel, 1978) and grid cells in the entorhinal cortex that fire at multiple locations on a triangular grid (Hafting et al., 2005). Along with these two seminal cell types a multitude of other cells have been discovered such as head-direction cells (Taube et al., 1990), object-vector cells (Høydal et al., 2019), reward cells (Gauthier and Tank, 2018), boundary-vector cells (Lever et al., 2009) and goal-direction cells (Sarel et al., 2017). Collectively it is thought that these different cell types are the substrate for a cognitive map in the spatial domain (Tolman, 1948), which can be used for flexible inferences such as path integration (McNaughton et al., 2006). However, increasing evidence is showing that these cell types are involved in non-spatial tasks such as the manipulation of sound by rodents (Aronov et al., 2017) and the navigation of abstract conceptual representations by humans (Constantinescu et al., 2016). This raises an important question, do these different cell types support a more general ability to form non-spatial cognitive maps and how might the hippocampus construct such maps for flexible behaviour outside of the spatial domain?

To address these questions Whittington et al. (2019) have proposed a computational model known as the Tolman-Eichenbaum Machine (TEM) (Figure 3.7) that captures many of the spatial and non-spatial findings in the hippocampus and entorhinal cortex. At its core, TEM proposes that entorhinal cells represent a basis for structural knowledge while hippocampal cells link this basis with sensory representations also from the entorhinal cortex. This factorises structural regularities or relationships from the sensory content of an experience so that the learnt structure can easily be re-used in novel situations. In the spatial domain, structural regularities are based on the rules of Euclidian space. In comparison, in the non-spatial domain of the Transitive Inference Task, the structural regularity is that of an ordered line. These structural regularities are what allow for flexible inference as they can be re-combined with different sensory stimuli in order to guide decision-making.

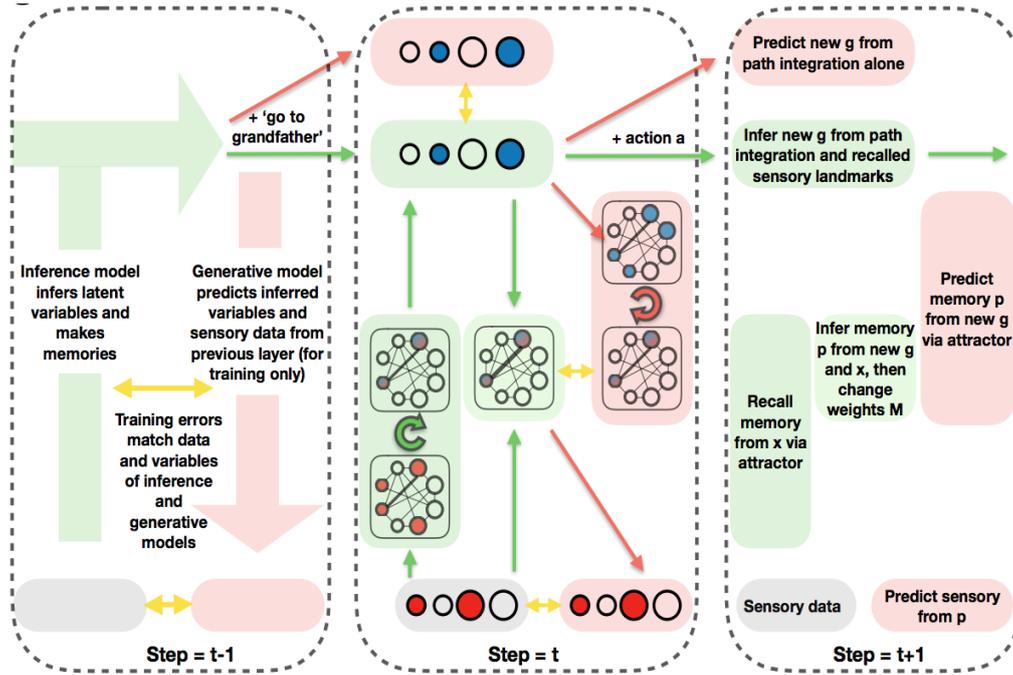


Figure 3.7: *General architecture of Tolman-Eichenbaum Machine (TEM). Step $t-1$ describes the architecture in terms of Bayesian logic, step t describes the network implementation, and step $t+1$ describes the computations in words. TEM consists of two models; a generative model (red) and an inference model (green). The inference model is used to predict the latent variables \mathbf{g} (blue units) and \mathbf{p} (blue/red units) based on the observed variable \mathbf{x} (red units). The generative model then uses the results to predict the values of all the variables, which generates training errors for each of them (yellow arrows). Circular arrows represent attractor dynamics. Red and green arrows are learnable weights. Black lines between units represent weights that are updated using Hebbian learning. Figure adapted from Whittington et al. (2019).*

From an implementational point of view, all variables in TEM are encoded as vectors. TEM represents the abstract relational structure as a graph, where a latent variable \mathbf{g} is used to represent the location in the graph. Learning of the abstract relational structure corresponds to learning predictive representations that represent the graph transitions from \mathbf{g} to \mathbf{g}' based on different actions. TEM achieves this by using generative and inference models, along with a training scheme similar to the Wake Sleep algorithm or a Helmholtz machine (Hinton et al., 1995; Dayan et al., 1995) (Figure 3.7). Importantly, TEM factorises this structural knowledge from sensory representations via conjunctive representations. TEM uses a variable \mathbf{p} to represent the grounded conjunction of \mathbf{g} and a sensory experience \mathbf{x} . \mathbf{p} is learnt using Hebbian learning, which causes the sensory stimulus and the abstract location to be tied together in a one-shot fashion when both of them are active. One useful property

of this form of learning is that given either \mathbf{g} or \mathbf{x} , the Hebbian memory system can use pattern completion to complete the conjunctive representation \mathbf{p} . This allows TEM to either infer the sensory representation from the abstract location or vice versa, all via the conjunctive representations stored in hebbian memory. Indeed, this may be where recurrent similarity computation proves particularly useful as it can fetch related conjunctive representations.

From the perspective of biological plausibility, Whittington et al. (2019) propose that abstract locations (\mathbf{g}) are stored in the Medial Entorhinal Cortex (MEC), sensory representations (\mathbf{x}) in the Lateral Entorhinal Cortex (LEC) and conjunctions of the two (\mathbf{p}) in the hippocampus. In addition, the weights between the units of \mathbf{p} are learnt using Hebbian learning, which mirrors the very fast learning abilities of the hippocampus (O’Reilly and Rudy, 2000). Similarly, \mathbf{p} is retrieved using attractor dynamics and pattern completion, which are known properties of the hippocampus (Wills et al., 2005; Rolls, 2007). Finally, the use of a generative model during learning is of interest as it has been suggested that hippocampal replay is sampled from a generative model (Foster and Wilson, 2006; Igata et al., 2020; O’Neill et al., 2017) (see Section 3.4.2).

Whittington et al. (2019) tested TEM on transitive inference, social hierarchy and 2D spatial tasks in order to demonstrate its ability to use relational knowledge for transfer in novel situations. All of these tasks are similar in that they can be represented as a relational graph with edges and nodes. In all cases, after training on a variety of environments TEM is able to infer the correct sensory experience on the second visit to a node even if the edge/route has never been taken before. This demonstrates that TEM is able to learn the relational structure and use it to make novel inferences about transitions it has never experienced.

Having demonstrated the ability of TEM to perform relational inferences, Whittington et al. (2019) also investigated the representations learnt by TEM while it randomly traversed a 2D graph. Starting with the representations of the abstract graph locations \mathbf{g} , Whittington et al. (2019) found that TEM learnt representations akin to grid cells found in the MEC. The learnt representations formed a grid-like code at different spatial frequencies and phases, as is commonly found in the MEC. In addition, as with biological grid cells, these representations remained consistent

across environments making them a useful basis for generalising structure as they are not dependent on a single environment. Moving on to the conjunctive representations \mathbf{p} learnt by TEM, they appeared to reflect hippocampal place cells whereby they only fired in a single location. This is due to the fact that they only fire when both the abstract location \mathbf{g} and the sensory experience \mathbf{x} is present. This also meant that they changed between environments because the configuration of sensory experiences changed, which is also widely described in hippocampal place cells and referred to as re-mapping. Generally rodents do not perform random walks in an environment but prefer to approach objects and remain near boundaries. With this in mind Whittington et al. (2019) made TEM follow such a policy and found that \mathbf{g} representations mimicked border and object vector cells found in the entorhinal cortex, while \mathbf{p} representations mimicked landmark cells found in the hippocampus. These results suggest that the wide array of different cells types found in the hippocampus, which are commonly associated with spatial reasoning, may actually be the result of a more general relational reasoning mechanism.

The above findings were based on 2D graphs with a spatial interpretation, however if the entorhinal cortex truly does learn abstract relational structure then this should apply to non-spatial tasks. To test this hypothesis Whittington et al. (2019) gave TEM a circular track whereby it would receive reward every 3 laps. Such a task has recently been given to rodents and it was found that hippocampal cells encoded track location, track location and a specific lap number, or track location and a count of the number of laps. The \mathbf{p} representations learnt by TEM mirrored these three hippocampal cell types. In addition, TEM learnt \mathbf{g} representations that counted laps suggesting that they may form the basis for hippocampal representations which are a conjunction of abstract task space i.e. lap count, and sensory experience i.e. spatial position. These representations are likely to arise from the fact that they need to encode states that predict both the current sensory experience and different future states (i.e. reward). It is therefore insufficient to just encode spatial location, they also need to encode task location. The prediction that entorhinal cells should encode lap number is open to empirical investigation.

Taking all of the aforementioned results into consideration, it appears that the hippocampus and entorhinal cortex may be a potential locus for relational reason-

ing in the brain. In particular, the entorhinal cortex appears to factorise abstract relational knowledge from sensory experiences, while the hippocampus forms conjunctive representations of the two. This allows for inferences about state transitions in novel environments based on the common relationships between environments. It also demonstrates that spatial reasoning, and the plethora of cells thought to support it, may be the result of a general relational reasoning mechanism.

3.3.4 Model-Based Reinforcement Learning and the Successor Representation

The previous section highlighted how the hippocampus may be involved in forming a cognitive map that allows for relational inferences about state transitions. The ability to reason about state transitions is extremely useful when faced with a novel situation because it allows for planning by considering the consequences of one's actions. In the field of Reinforcement Learning (RL) this is known as Model-Based RL, which utilises (see Section 2.1.2.7) a representation of the environment's dynamics to calculate values and/or a policy. A representation of the environment's dynamics is viewed as the most flexible representation for RL because it allows for the updating of values with just a small amount of local experience rather than having to experience a whole trajectory.

The disadvantage of Model-Based RL, or planning through mental simulation, is that it is computationally expensive. An alternative approach to Model-Based RL, is Model-Free RL which learns cached values using trial and error. While computationally lightweight, Model-Free RL is inflexible in response to changes in the environment because it requires direct experience of all the state transitions and rewards in order to update the value function. In recent years a third approach has been proposed that sits between Model-Based and Model-Free RL in terms of computational cost and flexibility. This approach is called the Successor Representation (SR) (Dayan, 1993) and works by calculating how likely a future goal state is given the current state. Access to such a probability can inform decisions without needing to mentally simulate all the states and actions that lead to the goal state. For example, when deciding to move to France or not it may be useful to know that there is a higher probability of successfully moving to France if you are

currently taking French lessons. Interestingly, the SR has been suggested to reside in the hippocampus (Stachenfeld et al., 2014; Momennejad et al., 2017; Stachenfeld et al., 2017), which supports the idea from the previous section that the hippocampus is responsible for forming predictive representations that can be used to make inferences about novel events.

The SR relies on forming a ‘predictive map’ of a given environment, which contains the predictive relationships between the different states of an environment. Mathematically the SR can be represented as a simple function that takes the current state s and some future goal state s' and outputs the expected time spent in s' given the agent is currently at state s :

$$M^\pi(s, s') = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbb{I}(s_t = s') \mid s_0 = s\right] \quad (3.1)$$

$M^\pi(s, s')$ is dependent on the current policy and the ‘expected future occupancy’ of s' from s is typically discounted so that occupancies far into the future are down-weighted. Armed with the SR, the value function can be decomposed into the inner product of the SR and the environment’s reward function:

$$V(s) = \sum_{s'} M^\pi(s, s') R(s') \quad (3.2)$$

The value of a state is the amount of time spent in a future state multiplied by the amount of reward obtained in that future state, summed over all possible future states. The primary benefit of the SR is that it can be learned in a Model-Free manner at a low computational cost. The SR has the same definition as a value function but it calculates expected discounted occupancies rather than rewards. As a result, algorithms such as temporal difference learning can be used to update the SR, using the prediction error between observed and expected state occupancies:

$$M_{t+1}^\pi(s_t, s') = M_t^\pi(s_t, s') + \eta[\mathbb{I}(s_t = s') + \gamma M_t^\pi(s_{t+1}, s') - M_t^\pi(s_t, s')] \quad (3.3)$$

While the computational cost of calculating and using the SR is low, it is significantly more flexible than Model-Free RL in circumstances where the environment’s

reward function changes. This is due to the fact that when calculating values the reward function is separate from the SR and so changes to the reward function can be used to update values without any re-learning of the state dynamics. In comparison, changes to the state dynamics will require complete re-learning of the SR via temporal difference learning, rendering the SR inflexible to such changes.

The predominant behavioural evidence for the SR in animals comes from reward revaluation studies such as Adams (1982). In this study rats were trained to press a lever for sucrose and then ceased to do so when the sucrose was paired with illness outside of the context of lever pressing. This behaviour demonstrates that changes to the reward contingencies can be accounted for by rats value estimates without the need for direct experience of pressing the lever and getting the sucrose paired with illness. Such behaviour cannot be accounted for by Model-Free RL because it requires direct experience of both the state transitions and the new reward in order to update value estimates. In comparison, both Model-Based RL and the SR can account for the behaviour because they can use their knowledge of the state dynamics to incorporate the change in reward into their value estimates immediately. Similar conclusions can be drawn from experiments investigating latent learning (Tolman, 1948), whereby rodents learn to obtain reward faster in a maze if they are allowed to explore the maze reward-free before hand.

Both reward revaluation and latent learning can be accounted for by either the SR or Model-Based RL, which raises the question of whether one or both are responsible for the observed flexible behaviour. Recent work by Momennejad et al. (2017) has attempted to dissociate between the two approaches in humans by exploiting the fact that the SR should be flexible to changes in the reward structure and inflexible to changes in the transition structure, where as Model-based RL should be flexible in response to both. Momennejad et al. (2017) provided participants with two different non-overlapping sequences of states that led to different monetary values. During learning participants had to choose which starting state they wanted in order to obtain the most reward. After this initial learning phase, participants learnt that the end of each sequence of states was associated with either (1) different reward values or (2) different state transitions. Participants then had to use this information to reverse their initial starting state preference. Momennejad et al.

(2017) hypothesised that if participants utilise the SR then they should reverse their starting state preference more frequently in the reward revaluation case compared to the transition revaluation case. This was indeed what the authors found, suggesting that the SR may be a plausible representation for RL in the brain.

From a neural perspective, the SR has been heavily implicated with the hippocampus and in particular place cells. Place cells have long been thought to encode an animal's current spatial location, however it has been proposed that instead they may encode an animal's future locations. If true, then in the spatial domain, future locations are directly related to future state occupancies and place cells may therefore be a neural correlate of the SR (Stachenfeld et al., 2017). A key prediction of this hypothesis is that place cells should be dependent on the animal's policy and should therefore be affected by the environment's dynamics. Recent findings are consistent with this prediction; the firing of hippocampal place cells appear to be distorted by spatial barriers (Alvernhe et al., 2011) and cluster around reward locations (Hollup et al., 2001) as expected if place cells encode states that are likely to be visited based on the animal's policy. While place cells have been heavily linked to spatial cognition, a growing body of evidence is suggesting that place cells are also involved in non-spatial processing (Tolman, 1948; Constantinescu et al., 2016; Aronov et al., 2017), which is important as the SR is applicable to many domains. This is pivotal to the current thesis as it suggests that the hippocampus may subserve broader forms of flexible behaviour via a general SR mechanism.

While growing evidence suggests that the SR is encoded by hippocampal place cells, the neural implementation of how the SR is learnt is still under debate. One interesting suggestion is that phasic midbrain dopamine neurons, commonly associated with value learning in temporal difference learning, may also be responsible for providing the error signal needed to learn the SR. The main evidence for this suggestion comes from the fact that these dopamine neurons appear to respond to many elements of the environment that are not reward-related. In particular, the firing of phasic midbrain dopamine neurons appear to respond to sensory prediction errors (Takahashi et al., 2017) and drive learning from these errors (Chang et al., 2017). Similarly, they appear to be responsible for learning stimulus-stimulus associations (Sharpe et al., 2017). These findings that phasic midbrain dopamine neurons are

implicated in sensory and stimulus-driven learning via prediction errors makes them a plausible candidate for the learning of future state occupancies required by the SR.

3.4 3. Connections Between the Neocortex and Hippocampus

Sections 3.2 and 3.3 explored how the computational properties of either a cortical or hippocampal learning system could enable humans to perform efficient Reinforcement Learning (RL). However, the final pathway in Figure 3.1 indicates that these two systems do not operate in isolation but communicate with each other to further support intelligent behaviour. This communication is key as it allows the two systems to complement each other during learning and decision-making. The remainder of this section explores how this communication may be enacted in the brain and how it contributes to efficient RL. Critically, we believe that a holistic explanation of rapid learning and transfer should address the interaction of these different learning systems.

3.4.1 Re-play

Interactions between the hippocampus and neocortex have been a topic of interest for neuroscientists for many years. One of the most well known interactions is that of ‘replay’ (Skaggs and McNaughton, 1996; Nádasdy et al., 1999; Ji and Wilson, 2007; Karlsson and Frank, 2009). The term replay corresponds to the finding that experiences stored in the hippocampus appear to be replayed during periods of rest in biological agents. Importantly this replaying of experiences has been found to coincide with replay of the same experiences in the neocortex. For example, Ji and Wilson (2007) recorded the activity of neurons in the visual cortex and hippocampus of sleeping rodents. They found that spiking patterns corresponding to the same awake experience occurred in both areas in a coordinated manner during sleep. Subsequently, similar findings have also been found in awake rodents during periods of rest (Karlsson and Frank, 2009). Overall, these findings suggest that during periods of rest both areas work in synchrony to consolidate memory through

synchronised re-activation.

How could this synchronous activity support memory consolidation? From a theoretical standpoint, McClelland et al. (1995) have proposed that replay is used to sample a range of individual memories from the hippocampus. These samples are then used to train the neocortex to abstract generalities from the sampled experiences. With this view in mind, replay is important for two reasons. Firstly, it allows for additional training offline, which can speed up the rate of learning. Secondly, the random sampling of experiences allows for interleaved training that removes spurious temporal correlations. Ultimately this process leads to the consolidation of knowledge from the hippocampus to the neocortex, freeing up resources in the hippocampus. Evidence for this process has come from the fact that damage to the hippocampus appears to affect memory for recent events but not distant ones (Scoville and Milner, 1957; Morris, 2006; Tse et al., 2011). This suggests that as time progresses memories become less and less dependent on the hippocampus.

The theoretical predictions of McClelland et al. (1995) are particularly interesting from a Reinforcement Learning (RL) point of view because similar techniques have been used in Deep RL. For example, the seminal Deep Q Network (DQN) (Mnih et al., 2015) relied upon a mechanism similar to that of biological replay. In the case of DQN, the neocortical learning system was represented by a Deep Neural Network (DNN), with its slow learning of distributed representations over many training examples. Conversely, the hippocampal learning system was represented by a table of past experiences $(s_t, a_t, s_{t+1}, a_{t+1})$ known as an experience replay buffer. Crucially, this table was used to help train the DNN by randomly sampling experiences for gradient descent updates.

This method of communication between the two systems therefore appears to mirror the theoretical motivations for biological replay. More specifically, it allowed for offline interleaved training of DQN, which sped up learning and removed temporal correlations. However, despite DQN having a mechanism that appears to parallel biological replay, this does not seem to be sufficient to capture the level of efficiency demonstrated by humans. This raises the question of whether additional mechanisms of communication between the two systems can get us closer to an understanding of how the brain achieves efficient RL.

3.4.2 Pre-play

The classic account of replay is that the hippocampus and neocortex simultaneously re-activate cells associated with past experiences. However, recent evidence has suggested that not only are past experiences replayed but so are novel experiences. For example, a study by Gupta et al. (2010) explored the content of replayed memories in the hippocampus while rodents participated in a simple maze task with a food reward. Gupta et al. (2010) found that some of the hippocampal replay events encoded routes in the maze that had not been experienced before but that provided short-cuts to the food goal. Similarly, Ólafsdóttir et al. (2015) conducted another maze task, which involved rats exploring a T-maze. The rats were allowed to approach the junction of the T-maze but both of the arms were blocked off. Subsequently, the rats observed food being placed into one of the arms. After a rest period the rats were then allowed to re-enter the maze and the arms were no longer blocked. Interestingly, replay events in the hippocampus during the rest phase of the experiment encoded a route along the arm of the maze containing the food but not the other arm. This was taken as evidence that the hippocampus can replay future possible experiences and that this replay is modulated by the presence of reward. This phenomenon is often known as ‘pre-play’ (Dragoi and Tonegawa, 2011, 2013) and refers to the offline activation of hippocampal cells that encode events that may happen in the future.

From an RL point of view, pre-play has several interesting implications that may help to promote transfer and flexible behaviour. If the hippocampus is generating novel experiences to train the neocortex then this would help prepare the neocortex for action selection in novel environments even before they have been experienced. This is similar to the Dyna Q architecture proposed by Sutton and Barto (1998), whereby a transition model of the world is used to update the value function by generating simulated trajectories in the current environment. The only difference here is that the trajectories are not simulations of the current environment but of potential new environments or unexplored state transitions. It is therefore tempting to view the hippocampus as a generative model that uses past experiences to generate plausible new ones. Indeed, human patients with hippocampal amnesia are severely impaired when it comes to imagining new experiences (Hassabis et al., 2007)

and functional magnetic resonance imaging studies suggest that the hippocampus is involved in both remembering and imagining events (Addis et al., 2007). This suggests that the hippocampus in both humans and rodents may play a key role in generating new experiences from imagination.

From a computational perspective, a multitude of modelling studies have used the idea of the hippocampus as a generative model to improve the capabilities of connectionist approaches. Most notably, Mocanu et al. (2016) demonstrated how generative replay could be used to reduce catastrophic forgetting in a connectionist network (Figure 3.8) (see also Shin et al., 2017; van de Ven et al., 2020). Typically to avoid catastrophic forgetting a history of all past examples must be kept in order to train a network offline, as we have seen in DQN (Mnih et al., 2015). However Mocanu et al. (2016) showed that you do not need to store past examples but instead you can use them in an online manner to train a generative model. In their case, Mocanu et al. (2016) used a Restricted Boltzmann Machine (RBM) to model the input data and on every time step it was trained to re-create new incoming data points as well as a set of data points that were sampled from the RBM itself. In this way the RBM was trained to incorporate new information and retain old information without the need to store explicit examples. This is important because it allows for continual lifelong learning and the incorporation of old information with new information. Subsequent research has shown that training a generative model in this way can then be used to replay simulated data to networks that are used for prediction (Shin et al., 2017; van de Ven and Tolias, 2018). This allows the networks used for prediction to also avoid catastrophic forgetting and abstract commonalities across all previous experiences. It is distinctly possible that the hippocampus is employing a similar mechanism to learn a generative model of the world, which can then be used to train the neocortex to abstract similarities across a wide range of experiences.

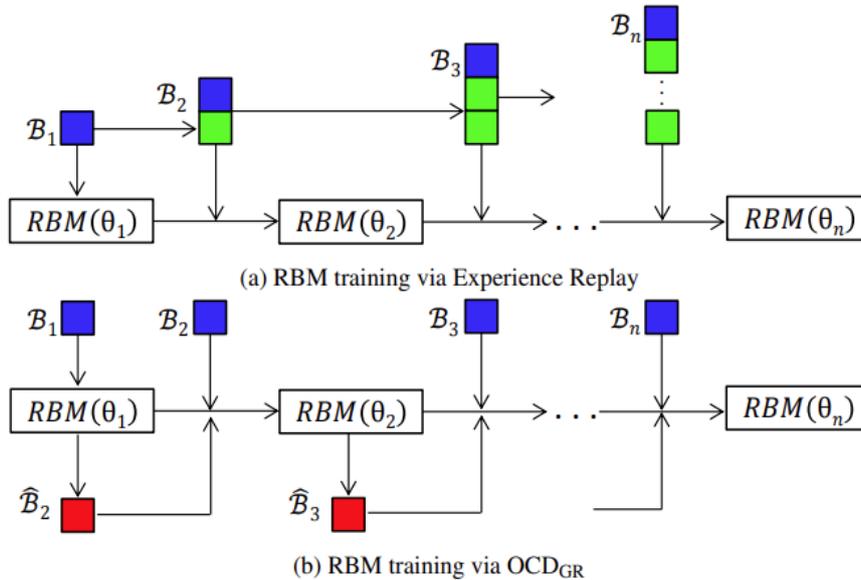


Figure 3.8: *Depiction of using generative replay to overcome catastrophic forgetting. (A) Typically connectionist models are trained by sampling a batch of data (B_t) from a history of previous data at each time step. This leads to a large memory cost for storing past data. (B) Online Contrastive Divergence with Generative Replay (OCD_{GR}) (Mocanu et al., 2016) forms batches at each time step by combining new data with data sampled from a generative model (\hat{B}_t), such as a Restricted Boltzmann Machine (RBM). This removes the need for a large history of past data and can be used to train on data that has never been experienced before. Figure adapted from Mocanu et al. (2016)*

The fact that the hippocampus may represent a generative model that can reduce catastrophic forgetting is important for transfer because the representations learnt by the neocortex will not over-fit to the most recent experiences. However, the aforementioned results from rodent (Gupta et al., 2010; Ólafsdóttir et al., 2015) and human studies (Hassabis et al., 2007; Addis et al., 2007) suggest that the real promise of the hippocampus as a generative model is that it can imagine experiences that may be useful for future tasks i.e. pre-play. This can be seen as a form of forward planning in anticipation for new tasks rather than just maintaining knowledge of previous ones. The ability of the hippocampus to generate such pre-emptive experiences is likely to be dependent on the knowledge being used to formulate the generative model. For example, if knowledge of the relationships between objects or the outcomes of ones actions is incorporated into the generative model then this should increase the capabilities of the hippocampus to generate examples that are highly dissimilar from past experiences and that support pre-play. It remains an

open question how a computational implementation of pre-play could be formulated in order to improve the transfer ability of RL algorithms.

3.5 Conclusions

In this chapter we have explored Complementary Learning Systems (CLS) theory as a guiding framework for how humans are able to demonstrate efficient RL in terms of rapid learning and transfer. A natural analogy appears to exist between the use of a neocortical learning system in CLS theory, and the use of Deep Neural Networks (DNNs) in Deep RL. However, this analogy falls short when trying to replicate the efficiency exhibited by human RL. This failure is likely due to the differences between the neocortex and DNNs as well as the lack of additional learning systems in Deep RL.

Several avenues of research are helping to improve the use of DNNs to approximate a neocortical learning system that contains semantic knowledge for RL. One critical avenue is preventing catastrophic forgetting in DNNs so that they can learn representations that generalise across multiple tasks (French, 1999; Kirkpatrick et al., 2017). In addition, researchers are investigating the objective functions utilised by the brain and the order in which they occur, in an effort to improve our understanding of DNNs as a model of semantic memory (Khaligh-Razavi and Kriegeskorte, 2014; Cadieu et al., 2014). Finally the use of inductive biases in the training of DNNs appears to be a promising avenue for moving the efficiency of Deep RL towards human levels. In particular, the learning of disentangled representations appears to have a positive effect on transfer in DNNs that are used for RL (Higgins et al., 2016, 2017). Improving DNNs as a model of semantic memory is likely to have a huge benefit in capturing human RL behaviour, particularly because CLS theory predicts that it should be the primary source of generalisation in the brain.

Central to CLS theory is the idea that the hippocampus is important for complex behaviour because of its complementary properties with the neocortex. The hippocampus is thought to rely on the rapid learning of pattern separated representations that encode individual experiences. This is critical for efficient RL because it allows for faster learning and reduced interference compared to a neocortical learn-

ing system. As we have seen, these benefits have been demonstrated by the use of tabular approaches in Deep RL (Blundell et al., 2016). It has long been believed that because the hippocampus stores individual experiences in a pattern-separated manner it therefore lacks any generalisation properties. However, ongoing research suggests that this assumption may be ill-founded and that it does in fact possess properties that allow for transfer. For example, the hippocampus appears to be able to perform some forms of generalisation through recurrent similarity computation, allowing it to solve relational tasks such as Transitive Inference (Kumaran and McClelland, 2012). Some lines of research are beginning to take this one step further and suggest that the hippocampus forms the basis of a cognitive map that can support general relational reasoning in novel environments (Whittington et al., 2019). Finally, the hippocampus may play a critical role in model-based RL by encoding the probability of occupying a successor state given the current state (Momennejad et al., 2017). All of these bodies of research highlight that in addition to the fast learning of pattern-separated representations, the hippocampus has other useful properties such as recurrency, conjunctive representations and predictive representations. Deep RL models should look to these additional properties if they hope to capture and explain the efficiency of human RL.

The properties of each system aside, there also appear to be crucial forms of interaction between the two systems. Most notably, it has been suggested that the hippocampus may represent a generative model of the world, capable of generating examples of previous and novel environments (Gupta et al., 2010; Ólafsdóttir et al., 2015; Hassabis et al., 2007; Addis et al., 2007). This serves two key purposes, firstly it reduces catastrophic forgetting in the neocortex without the need for vast memory resources. Secondly, it allows for the hippocampus to preemptively train the neocortex on potential future environments allowing for pro-active transfer. This highlights the importance of considering the brain as multiple learning systems that communicate with each other to support learning.

In the next chapter we will present a novel algorithm that attempts to model how the benefits of both a neocortical and hippocampal learning system can be used for efficient RL. This approach incorporates all three pathways in Figure 3.1 by allowing both systems to contribute to the evaluation of states and/or actions

(Pathways 1 and 2), and allowing communication between the two systems in the form of Temporal Difference (TD) errors (Pathway 3). The result of this architecture is that the neocortical system is used to generalise over the state space while the hippocampus encodes salient experiences that violate these generalisations. The algorithm demonstrates the utility of CLS theory for understanding efficient RL in the brain and future work should investigate how its core principles can be combined with the advancements outlined in this chapter.

Chapter 4

Complementary Temporal Difference Learning

Overview

In this chapter we present a novel algorithm for efficient Reinforcement Learning (RL) called Complementary Temporal Difference Learning (CTDL). As predicted by Complementary Learning Systems (CLS) theory, CTDL combines a neocortical and hippocampal learning system to exploit the benefits of both systems: the generalization properties of the neocortex and the fast, interference-free, learning of the hippocampus. CTDL represents the neocortical learning system as a Deep Neural Network (DNN) and the hippocampal learning system as a Self-Organising Map (SOM) (Section 4.2). Key features of CTDL include using both the SOM and DNN to evaluate actions and states, and using the Temporal Difference (TD) error from the DNN to update the SOM. We evaluate CTDL on Grid World, Cart–Pole and Continuous Mountain Car tasks and show several benefits over classic Deep RL approaches (Section 4.3). Our results demonstrate (1) the utility of complementary learning systems for the evaluation of actions and states, (2) that the TD error signal is a useful form of communication between the two systems and (3) that our approach extends to both discrete and continuous state and action spaces.¹

¹The work in this chapter has been published in *Neural Networks*: Blakeman, S., & Mareschal, D. (2020). A complementary learning systems approach to temporal difference learning. *Neural Networks*, 122, 218-230.

4.1 Introduction

Complementary Learning Systems (CLS) theory posits that the neocortex and hippocampus have complementary properties that allow for complex behavior (McClelland et al., 1995; Kumaran et al., 2016). More specifically, the hippocampus relies on the fast learning of conjunctive, pattern-separated representations of individual memories. These memories then support the learning of a second system, the neocortex, which slowly learns overlapping representations that support generalisation across features and experiences. The Deep Neural Networks (DNNs) used in Deep Reinforcement Learning (RL) share similar properties with the neocortex in that they also rely on the slow learning of overlapping representations. CLS theory therefore predicts that Deep RL algorithms could benefit from the addition of a hippocampal learning system (Gershman and Daw, 2017).

In Chapter 3 we outlined three distinct pathways that describe how the hippocampus, sensory cortex and striatum may interact to achieve rapid learning and transfer in RL (Figure 3.1). Previous computational work has focused on subsets of these pathways and has not attempted to model a system that utilises all three. In this chapter we focus on how the rapid learning of individual experiences by a hippocampal system can be used in combination with a neocortical system to improve the efficiency of RL. As mentioned in Chapter 3, previous work by Blundell et al. (2016) and Pritzel et al. (2017) has already shown that the use of episodic memory to evaluate states and actions can improve the efficiency of Deep RL systems. This highlights the importance of pathway 2 (Figure 3.1) in RL and improves the analogy between Deep RL systems and the brain. However, these approaches still have several fundamental differences to the architecture of the brain. Most notably, these approaches rely on a Deep Neural Network (DNN) followed by a tabular method. This is akin to using a neocortical learning system followed by a hippocampal learning system, which is at odds with the parallel projections of the neocortex and the hippocampus to the striatum in the brain. To faithfully replicate CLS theory, a system would need to use both a hippocampal and neocortical learning system in parallel to evaluate states and actions. Furthermore, the two systems should communicate with each other in order to support each others learning. It therefore appears that the work of Blundell et al. (2016) and Pritzel et al. (2017) lack mecha-

nisms that correspond to pathways 1 and 3 in Figure 3.1, which may be crucial for understanding the importance of complementary learning systems in the brain.

With this in mind, we present a novel method for imbuing a Deep RL agent with both a ‘neocortical’ and a ‘hippocampal’ learning system so that it utilises all the pathways in Figure 3.1 (Blakeman and Mareschal, 2020a). Most importantly these two systems: (1) learn in parallel, (2) communicate with each other using a biologically plausible signal, and (3) both make value predictions. From a computational perspective, this allows for the fast learning of pattern-separated representations that reflect salient individual events and the slow learning of overlapping representations that generalise across experiences. We represent the neocortical system as a DNN and the hippocampal system as a Self-Organising Map (SOM). We use a SOM to represent the hippocampal learning system because it can utilise large learning rates for fast learning and each unit stores its own weight vector making the representations pattern-separated. In addition, the size of the SOM is significantly smaller than the state space experienced by the agent, which replicates the restricted computational resources of episodic memory. Importantly, the SOM is tasked with storing pattern-separated memories of states that the DNN is poor at evaluating. This allows the DNN to generalise over the state space while the SOM quickly encodes important violations of these generalisations. To achieve this interaction we use the TD error from the DNN to train the SOM. This approach demonstrates how the TD error of a ‘cortical’ system can be used to inform a ‘hippocampal’ system about when and what memories should be stored. Both systems contribute to the evaluation of action-values, which allows the agent to utilize the benefits of both a neocortical and hippocampal learning system for action selection. We call our novel algorithm *Complementary Temporal Difference Learning (CTDL)* and demonstrate that it can improve the performance and robustness of a Deep RL agent on the Grid World, Cart-Pole and Continuous Mountain Car tasks.

4.2 Methods

4.2.1 Complementary Temporal Difference Learning (CTDL)

Our novel approach combines a DNN with a SOM to imbue an agent with the benefits of both a ‘neocortical’ and ‘hippocampal’ learning system. The DNN is a simple feed-forward network that takes the current state as input and outputs the predicted action values for each action. The network is trained using the same training objective as Mnih et al. (2015) and a copy of the network is made every C time steps in order to improve training stability. The optimiser used was RMSProp and the hyper-parameter values can be seen in Table 4.2. Importantly, unlike in Mnih et al. (2015), no memory buffer is used to record past experiences, which saves considerable memory resources. The SOM component is represented as a square grid of units, with each unit having a corresponding action-value $Q(u, a)$ and weights β_u that represent a particular state.

A general outline of the algorithm detailing how the DNN and SOM interact can be seen in Algorithm 4. In simple terms, the TD error produced by the DNN is used to update the SOM and both systems are used to calculate Q values for action selection. When the agent observes the state s_t , the closest matching unit in the SOM u_t is calculated based on the euclidean distance between the units weights β_u and s_t . This distance is also used to calculate a weighting parameter $\eta \in \{0, 1\}$, which is used to calculate a weighted average of the action values from the SOM and the DNN. If the best matching unit is close to the current state then a larger weighting will be applied to the Q value produced by the SOM. A free parameter τ_η acts as a temperature parameter to scale the euclidean distance between β_u and s_t when calculating the weighted average.

For learning in both the DNN and the SOM, the TD error is calculated using the difference between the target value and the predicted Q value of the DNN. The TD error is used to perform a gradient descent step with respect to the parameters θ of the DNN, which ensures that the predictions of the DNN move towards the weighted average of the SOM and DNN predictions. After updating the DNN, the TD error is also used to update the SOM. More specifically, the TD error is used to create an exponentially increasing value $\delta \in \{0, 1\}$, which scales the standard

Algorithm 4 - CTDL. Highlighted lines are unique to CTDL when compared to DQN

Initialize probability of selecting a random action $\epsilon = 1$
Initialize SOM weights β to random locations in the grid world
Initialize SOM action-values $Q^{SOM} = 0$
Initialize action-value function Q^{DNN} with random weights θ
Initialize target action-value function \tilde{Q}^{DNN} with weights $\theta^- = \theta$

for $e = 1, E$ **do**

If $\epsilon > \epsilon^{end}$ then decrease ϵ by $\epsilon^{end}/E^\epsilon$

for $t = 1, T$ **do**

Observe current state s_t and reward r_t

Retrieve SOM unit u_t that is closest to s_t

$u_t = \arg \min_u \|\beta_u - s_t\|^2$

Calculate weighting η based on distance

$\eta = \exp(-\|\beta_{u_t} - s_t\|^2/\tau_\eta)$

Calculate $Q(s_t, a')$ as weighted average of SOM and DNN values

$Q(s_t, a') = \eta Q^{SOM}(u_t, a') + (1 - \eta) \tilde{Q}^{DNN}(s_t, a'; \theta^-)$

$$\text{set } y_t = \begin{cases} r_t & \text{if episode is over} \\ r_t + \gamma \max_{a'} Q(s_t, a') & \text{otherwise} \end{cases}$$

Perform gradient descent step on $(y_t - Q^{DNN}(s_{t-1}, a_{t-1}; \theta))^2$ with respect to the network parameters θ

Calculate δ based on the TD error produced by the DNN

$\delta = \exp(|y_t - Q^{DNN}(s_{t-1}, a_{t-1}; \theta)|/\tau_\delta) - 1$

Calculate the neighbourhood function based on u_{t-1}

$T_{u_j, u_{t-1}} = \exp(-\|l_{u_j} - l_{u_{t-1}}\|^2/2(\sigma_c + (\delta * \sigma)))$

Update the weights β of SOM

$\Delta\beta_{ji} = \alpha * \delta * T_{u_j, u_{t-1}}(s_{t-1, i} - \beta_{ji})$

Update the action value $\Delta Q^{SOM}(u_{t-1}, a_{t-1}) =$

$\rho * \eta_{t-1} * (y_t - Q^{SOM}(u_{t-1}, a_{t-1}))$

Replay contents of SOM to DNN using a random sample of actions

a_t and unit weights β_u . y_t is set to $Q^{SOM}(\beta_u, a_t)$

Select random action with probability ϵ , else $a_t =$

$\arg \max_{a'} \eta Q^{SOM}(u_t, a') + (1 - \eta) Q^{DNN}(s_t, a'; \theta)$

Every C steps reset $\tilde{Q}^{DNN} = Q^{DNN}$

If the goal has been reached then **break** and end episode

end for

end for

deviation of the SOM’s neighbourhood function and the learning rate of the SOM’s weight update rule. Again a temperature parameter τ_δ is used to scale the TD error. Next, the action value of the closest matching unit from the previous time step $Q^{SOM}(u_{t-1}, a_{t-1})$ is updated using the learning rate ρ , the weighting from the previous time step η_{t-1} and the difference between $Q^{SOM}(u_{t-1}, a_{t-1})$ and the target value y_t . The inclusion of η_{t-1} ensures that the action value only receives a large update if the closest matching unit is similar to the state value.

To aid in the training of the DNN and to mimic biological ‘replay’, the contents of the SOM are replayed to the DNN as a training batch for gradient descent. To construct the training batch the actions a_t are sampled randomly, the states s_t are set to a random sample of the SOM weights β_u and the target values y_t are set to the corresponding Q values stored in the SOM. Finally, the agent’s actual action is chosen in an ϵ -greedy manner with respect to the weighted average of the predicted DQN and SOM Q values.

The aforementioned algorithm has several interesting properties. Firstly, the calculation of Q values involves the contribution of both the DNN and the SOM. The size of their respective contributions are controlled by the parameter η , which ensures that if the current state is close to one stored in SOM memory then the Q value predicted by the SOM will have a larger contribution. This is akin to retrieving a closely matching episodic memory and using its associated value for action selection. Secondly, because the SOM is updated using the TD error produced by the DNN, it is biased towards storing memories of states that the DNN is poor at evaluating. Theoretically this should allow the DNN to learn generalisations across states, while the SOM picks up on violations or exceptions to these generalisations and stores them in memory along with a record of their action values. If after many learning iterations the DNN converges to a good approximation of the optimal action-value function then no TD error will be produced and the SOM will be free to use its resources for other tasks. Finally, the SOM can use much larger learning rates than the DNN because it relies on a tabular approximation of the action-value function, which should improve data efficiency.

4.2.2 Simulated Environments

4.2.2.1 Grid World Task

The grid world task consists of procedurally generated 2D grid worlds (Figure 4.1). Each cell in the grid world represents a state $s \in \mathbf{R}^2$ that is described by its x and y position. If N is the number of cells in the grid world, then $\frac{N}{5}$ negative rewards (-1) are randomly placed in the grid world along with a single positive reward (+1) and the agents starting position. The agent’s task is to reach the positive reward, at which point the episode is over and a new episode begins. The agent’s action space is defined by four possible actions (up, down, left and right), each of which moves the agent one cell in the corresponding direction with probability 1. If the agent chooses an action that would move it out of the grid world then it remains where it is for that time step. Table 4.1 shows the hyper-parameter values used in all grid world simulations.

Table 4.1: Grid world hyper-parameter values used for all simulations.

Parameter	Value	Description
W	10	Width of grid world
H	10	Height of grid world
E	1,000	Number of episodes for learning
T	1,000	Maximum number of time steps per episode

4.2.2.2 Cart-Pole

The Cart-Pole problem, as provided by the OpenAI Gym (Brockman et al., 2016), consists of a cart with a pole attached by a single un-actuated joint. The goal of the agent is to control the velocity of the cart on a linear friction-less track so that the pole stays up-right. The state observed by the agent is made up of four values which correspond to the position of the cart $[-4.8, 4.8]$, the velocity of the cart $[-\infty, \infty]$, the angle of the pole $[\sim -41.8, \sim 41.8]$ and the velocity of the end of the pole $[-\infty, \infty]$. Two discrete actions are available to the agent; push the cart left and push the cart right. The agent receives a reward of +1 at every time step and an episode ends either when the angle of the pole is greater than 15 degrees, the cart moves off the screen or the episode length is greater than 500.

4.2.2.3 Continuous Mountain Car

The continuous mountain car environment, as provided by the OpenAI Gym (Brockman et al., 2016), is a 2D problem consisting of a car that starts in-between two hills. The goal of the agent is to drive the car to the top of the right-hand hill. This problem is complicated by the fact that the cars engine has insufficient power to drive straight up the hill. The agent therefore needs to learn to drive forwards and backwards in order to gain momentum and traverse the hill. The state observed by the agent is defined by the cars position $[-1.2, 0.6]$ and velocity $[-0.07, 0.07]$. Importantly the action space is continuous; the agent must choose to apply a force between 1 and -1 to the car at each time step. The agent receives a reward of +100 for reaching the target location but also receives a negative reward that is equal to the squared sum of the actions it has chosen. An episode terminates either when the car reaches the target location or the episode length is greater than 1000.

4.3 Results

In our first simulation we compare CTDL to the standard DQN described by Mnih et al. (2015) (see Section 2.3.1 for more details) on a range of grid worlds. Both CTDL and DQN utilise the same DNN architecture (see Table 4.2 for hyper-parameter values) but there are two key differences between the two approaches. Firstly, a standard DQN stores a memory buffer of size N that is used to replay past experiences whereas CTDL relies on the contents of a SOM for replay. For our simulations we set the memory buffer size M of the DQN to 100,000 while the size of the SOM was set to 36 units. This represents a significant decrease in memory resources between the two approaches. The second key difference is that a standard DQN only uses the DNN for calculation of Q values whereas CTDL also incorporates the predictions of a SOM. This allows CTDL to utilise the benefits of a ‘hippocampal’ learning system during decision making, namely pattern-separated memories and larger learning rates. Hyper-parameter values specific to CTDL can be seen in Table 4.3. Both models learned from 1000 episodes, with a maximum episode length of 1000. The probability of randomly selecting an action ϵ was linearly decreased from 1.0 to 0.1 over the first 200 episodes. The discount factor for future rewards

was set to 0.99 for all simulations.

Table 4.2: Hyper-parameter values used for the DNN component of DQN and CTDL in the grid world simulations.

Parameter	Value	Description
L	3	Number of layers
U	[128, 128, 4]	Number of units
C	10,000	Number of steps before updating the target network
B	32	Batch Size for training
λ	.00025	Learning rate for RMSProp
κ	.95	Momentum for RMSProp
ϕ	.01	Constant for denominator in RMSProp

Table 4.3: Hyper-parameter values unique to CTDL. τ_η , τ_δ , σ , σ_c , α and ρ were selected by using a random grid search on a single grid world.

Parameter	Value	Description
U	36	Number of units in SOM
τ_η	10	Temperature for calculating η
τ_δ	1	Temperature for calculating δ
σ	.1	Standard deviation of the SOM neighbourhood function
σ_c	.1	Constant for denominator in SOM neighbourhood function
α	.01	Learning rate for updating the weights of the SOM
ρ	.9	Learning rate for updating the Q values of the SOM

Figure 4.1 demonstrates the results of the two approaches on a random selection of grid worlds. CTDL outperforms the DQN in terms of cumulative reward and the cumulative number of ‘ideal’ episodes. An ideal episode is classified as an episode where the agent avoids all negative rewards and reaches the positive reward. These findings suggest that the inclusion of a second ‘hippocampal’ system, which explicitly contributes to the calculation of Q values, is beneficial in our simple grid world task. This gain in performance is achieved at a much lower cost in terms of memory resources. Figure 4.2 shows an example maze along with the weights of each unit in the SOM and the location they represent in the maze at the end of learning.

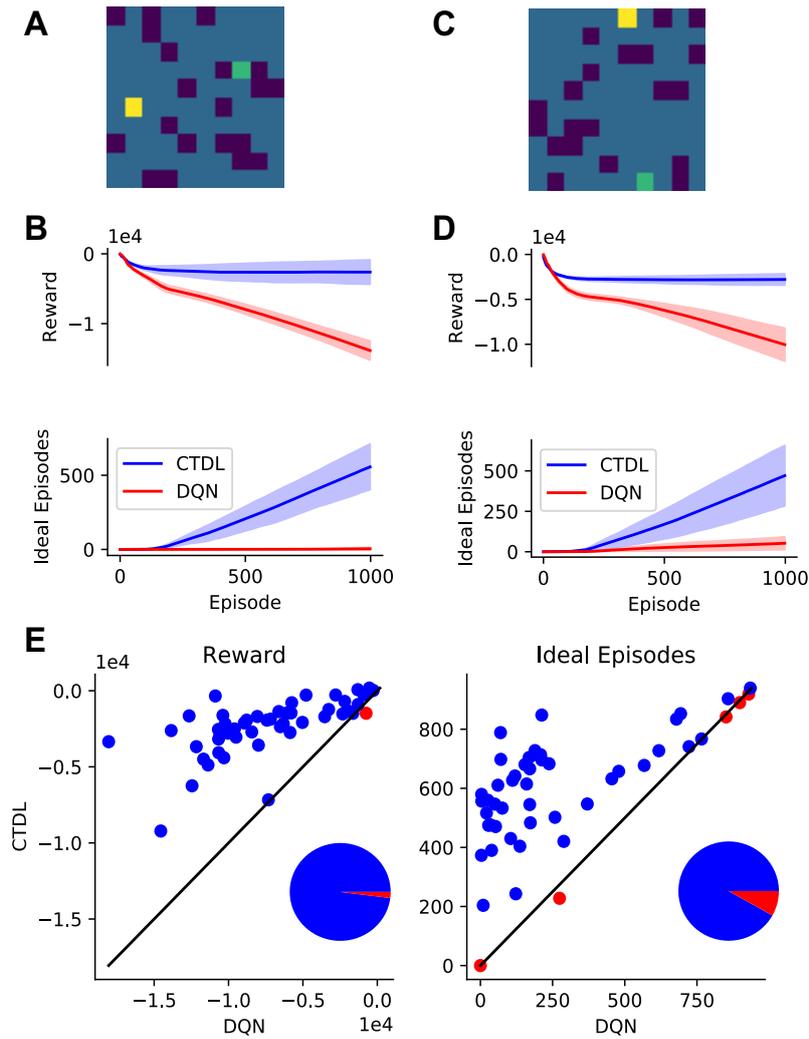


Figure 4.1: **A:** First example grid world, dark blue cells represent negative rewards (-1), the green cell represents the goal (+1) and the yellow cell represents the agents starting position. **B:** Performance of CTDL and DQN on the first example grid world in terms of cumulative reward and ‘ideal’ episodes over the course of learning. An ‘ideal’ episode is an episode where the agent reached the goal location without encountering a negative reward. Both CTDL and DQN were run 30 times on each maze. **C:** Second example grid world. **D:** Performance of CTDL and DQN on the second example grid world. **E:** Scatter plots comparing the performance of CTDL and DQN on 50 different randomly generated grid worlds. Both CTDL and DQN were run 30 times on each maze and the mean value at the end of learning was calculated. Blue points indicate grid worlds where CTDL out-performed DQN and red points indicate grid worlds where DQN out-performed CTDL. The pie charts to the lower right indicate the proportions of blue and red points.

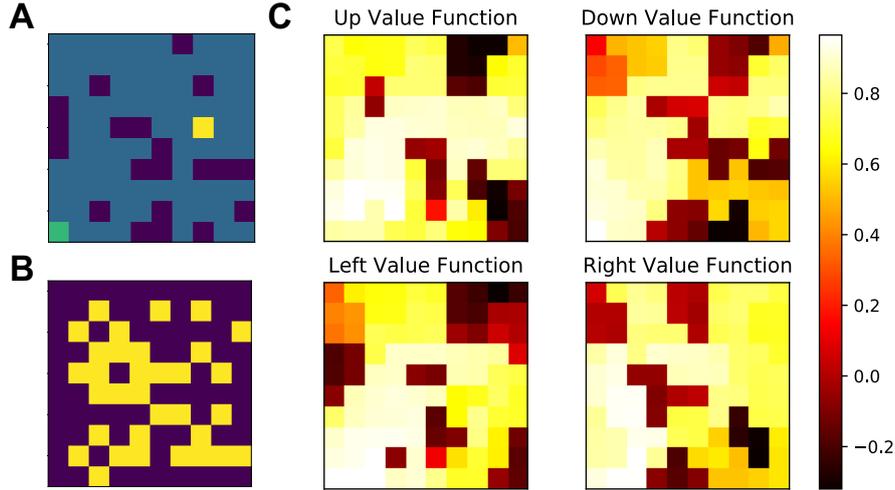


Figure 4.2: **A:** Randomly generated grid world, dark blue cells represent negative rewards (-1), the green cell represents the goal (+1) and the yellow cell represents the agents starting position. **B:** Image showing the locations encoded by the SOM component of CTDL (yellow cells) at the end of learning in A. **C:** CTDLs value function at the end of learning in A, the value is calculated as the weighted average of the predictions from the SOM and DNN. Each state has four possible values, corresponding to each of the four possible actions (up, down, left and right).

To improve our understanding of the mechanisms underlying CTDLs performance we isolated the contribution of the SOM to the calculation of the Q values from the replaying of the contents of the SOM to the DNN. Figure 4.3A shows the performance of CTDL both with and without replay. CTDLs performance was only marginally reduced by the removal of replay suggesting that the improvements over the DQN are due to the contribution of the SOM to the calculation of Q values. A key component of CTDL is the updating of the SOM using the TD error from the DNN. To investigate the importance of this interaction, we compared CTDL to a version of CTDL that did not update the SOM based on the TD error from the DNN. This was achieved by setting the learning rate of the SOM to 0 so that the weights β_u were not updated during learning. Figure 4.3B shows the results of this comparison. Removal of the interaction between the DNN and the SOM via the TD signal had a significant impact on the performance of CTDL, suggesting that it is a critical component of the model.

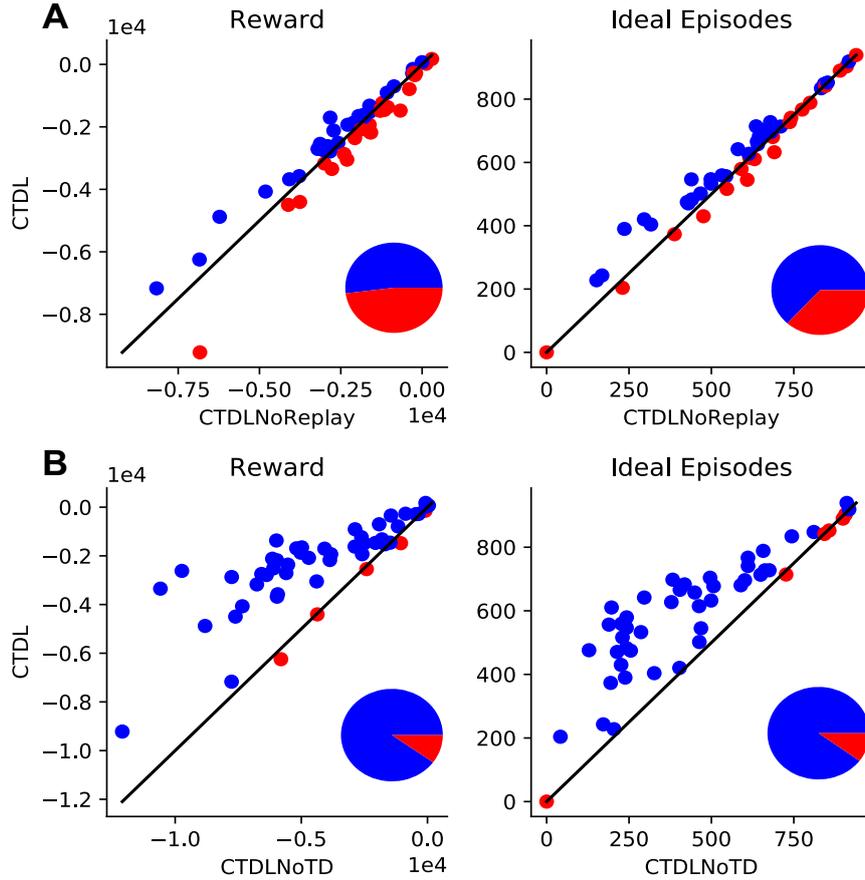


Figure 4.3: **A:** Scatter plots comparing the performance of CTDL and CTDL without replay on 50 different randomly generated grid worlds. Both CTDL and CTDL without replay were run 30 times on each maze. Blue points indicate grid worlds where CTDL out-performed CTDL without replay and red points indicate grid worlds where CTDL without replay out-performed CTDL. The pie chart to the lower right indicate the proportions of blue and red points. **B:** Scatter plots comparing the performance of CTDL and CTDL without TD learning in 50 different procedurally generated grid worlds. Both CTDL and CTDL without TD learning were run 30 times on each maze. Blue points indicate grid worlds where CTDL out-performed CTDL without TD learning and red points indicate grid worlds where CTDL without TD learning out-performed CTDL. The pie charts to the lower right indicate the proportions of blue and red points.

One interpretation of these results is that the SOM is able to store and use experiences that violate generalisations made by the DNN and that this confers a significant advantage during learning. To test this hypothesis we ran CTDL and DQN on three new mazes (Figure 4.4). The first maze had no negative rewards between the start and goal locations and the agent simply had to travel directly upwards. We predict that such a maze should favour the DQN because it can rely upon the generalisation that an increase in y corresponds to an increase in

expected return. The second and third mazes introduced negative rewards that violate this generalisation. For these mazes we predict that CTDL should perform better because it can store states that violate the generalisation in its SOM and when these states are re-visited CTDL can consult the Q values predicted by the SOM. Figure 4.4 shows the results of CTDL and DQN on these three mazes. To help visualise the locations encoded by the SOM we reduced the SOM size to 16 units. The results provide support for our predictions, with the DQN out-performing CTDL in the first maze but not in the second and third mazes. Interestingly, over the course of learning the locations encoded by the SOM appeared to reflect regions of the maze that correspond to violations in the ‘move upwards’ generalisation. We take these findings as evidence that the SOM is encoding states that violate generalisations made by the DNN and that this is responsible for CTDLs improved performance.

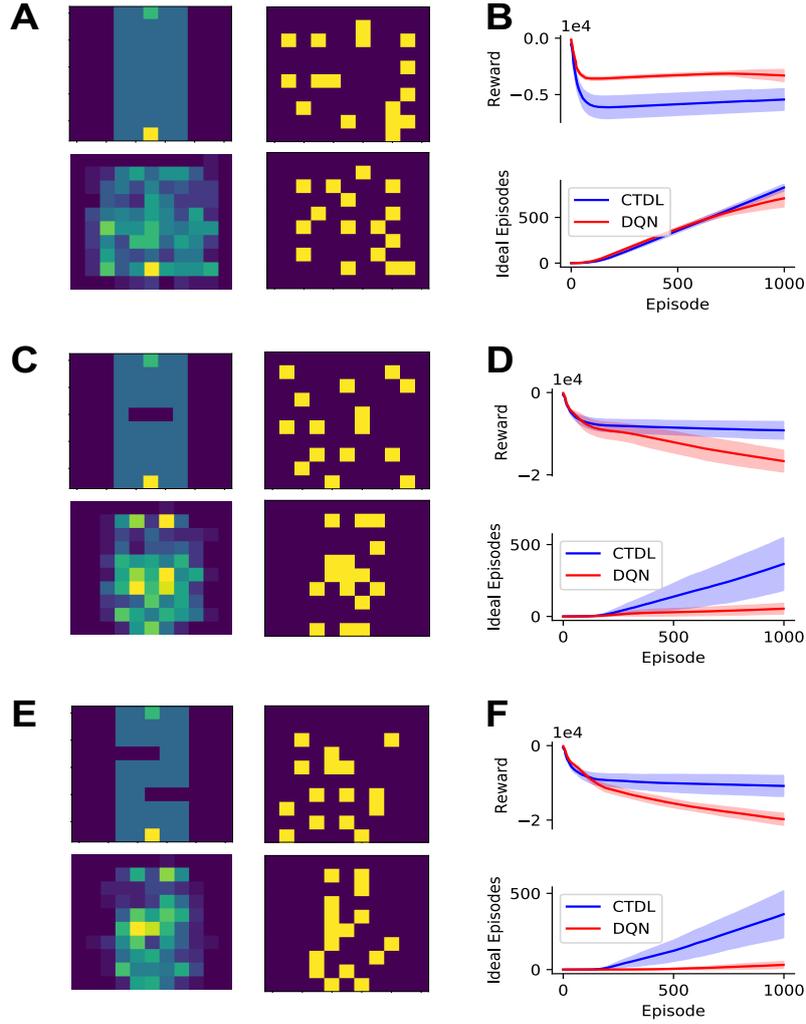


Figure 4.4: **A:** *Top-Left:* Grid world where the agent only has to travel upwards to reach the goal. Dark blue cells represent negative rewards (-1), the green cell represents the goal (+1) and the yellow cell represents the agents starting position. *Bottom-Left:* Locations encoded by the SOM component of CTDL at the end of learning, results are averaged over 30 runs. *Top-Right:* Locations encoded by the SOM component of CTDL at the start of learning for a single run. *Bottom-Right:* Locations encoded by the SOM component of CTDL at the end of learning for a single run. **B:** The performance of CTDL and DQN on the grid world from A in terms of cumulative reward and ‘ideal’ episodes. The solid line represents the mean and the shaded region represents the standard deviation. **C:** Same as A but an obstacle is introduced, in the form of negative rewards, that the agent must circumnavigate. **D:** The performance of CTDL and DQN on the grid world from C. **E:** Same as C but with two obstacles for the agent to circumnavigate. **F:** The performance of CTDL and DQN on the grid world from E.

If the SOM does encode states that violate generalisations made by the DNN, then this should translate to improved behavioral flexibility in the face of environmental changes. For example if an obstacle appears in one of the grid worlds then

this should lead to a large TD error and instruct the SOM to encode the position of the obstacle using its large learning rate. Subsequently, since the SOM keeps track of action values independently from the DNN, CTDL should be able to quickly adapt its behavior in order to avoid the obstacle. To investigate this hypothesis we ran CTDL and DQN on the grid world in Figure 4.4A immediately followed by the grid world in Figure 4.4C. Figure 4.5 shows the results of these simulations. As previously described, the DQN out-performed CTDL on the first grid world in terms of cumulative reward and the number of ideal episodes. Switching to the second grid world impacted the performance of both the DQN and CTDL. However, this impact was more pronounced for the DQN, with a larger decrease in cumulative reward and a plateauing of the number of ideal episodes. This suggests that CTDL is better equipped to handle changes in the environment. As before the locations encoded by the SOM appeared to reflect states immediately preceding the obstacle. This is consistent with the notion that the TD error from the DNN allows the SOM to identify regions that violate the generalisations made by the network and subsequently improve learning.

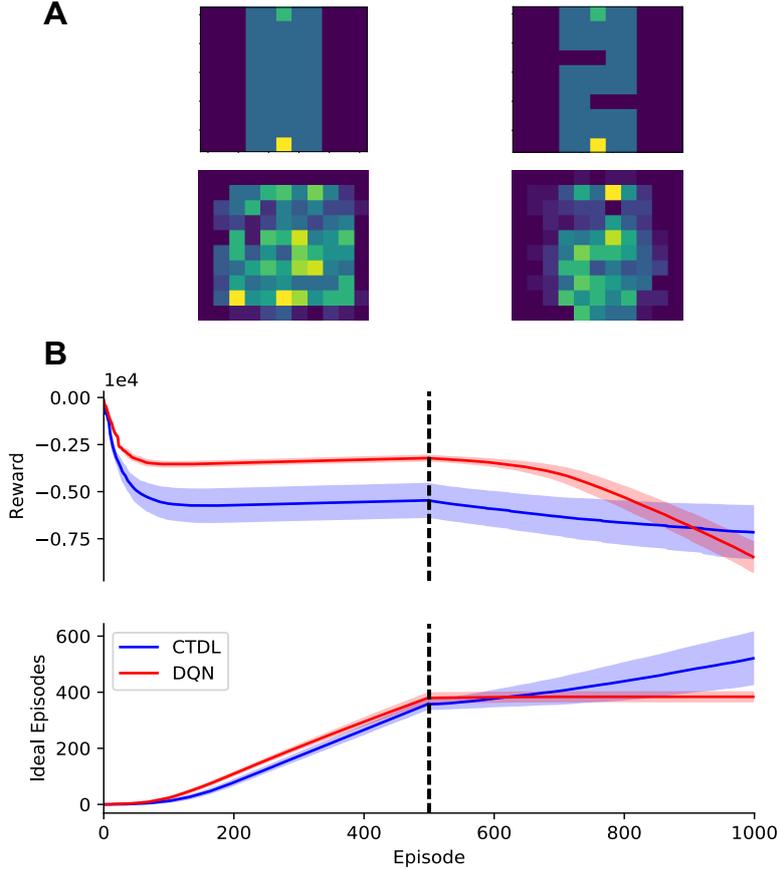


Figure 4.5: **A:** *Top-Left:* First grid world presented to the agent for 500 episodes. *Bottom-Left:* Locations encoded by the SOM component of CTDL at the end of learning in the first grid world, results are averaged over 30 runs. *Top-Right:* Second grid world presented to the agent for 500 episodes immediately after the first grid world. *Bottom-Right:* Locations encoded by the SOM component of CTDL at the end of learning in the second grid world, results are averaged over 30 runs. **B:** The performance of CTDL and DQN on the successive grid worlds from A. The solid line represents the mean and the shaded region represents the standard deviation. The dashed line indicates the change in grid worlds and the introduction of the obstacles.

One of the strengths of RL algorithms is that they can be applied to a wide array of tasks. If one can describe a task using a state space, an action space and a reward function then often it can be solved using RL techniques, especially if the states satisfy the Markov property. We therefore wanted to investigate whether the performance of CTDL was specific to grid worlds or whether it could be applied to other tasks. With this in mind, we chose to test CTDL on the Cart-Pole environment from OpenAI Gym because it is a common benchmark task in the RL literature and it involves a continuous state space, unlike the discrete state space of the grid world environments. The parameter values used for all Cart-Pole simulations were the

same as in the grid world simulations with two exceptions. Firstly, the number of time steps C between updates of the target network was changed to 500 in order to account for the shorter episodes experienced in the Cart-Pole task. Secondly, the size of the SOM was increased from 36 units to 225 units, which is still considerably smaller than the size of the replay buffer used by the DQN (100,000).

An important component of CTDL is the calculation of the euclidean distance between the current state s_t and the weights of each unit β_u . In the case of the Cart-Pole task this will cause the velocity values in the state representation to dominate the distance calculations because their values cover a much greater range. To account for this we maintain an online record of the largest and smallest values for each entry in the state representation and use these values to normalise each entry so that they lie in the range $[0, 1]$. This ensures that each entry in the state representation contributes equally to any euclidean distance calculations.

Figure 4.6 shows the results of both CTDL and DQN on the Cart-Pole task. While the DQN appeared to learn faster than CTDL, it did so with greater variance and the stability of the final solution was poor. In comparison, CTDL learnt gradually with less variance and there were no significant decreases in performance. These results demonstrate that CTDL can be applied to continuous state problems and is not restricted to discrete grid world problems. They also suggest that CTDL’s use of dual learning systems may confer a stability advantage that improves the robustness of learning.

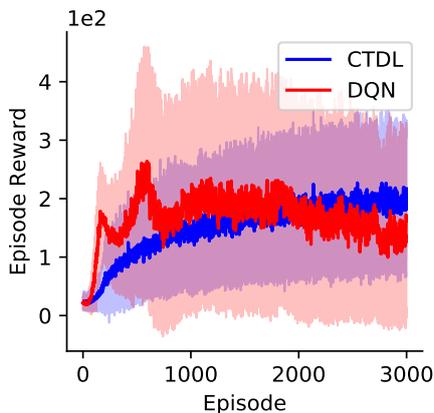


Figure 4.6: Episode reward achieved by CTDL and DQN on the Cart-Pole environment over the course of learning. Both CTDL and DQN were run 100 times on the Cart-Pole environment. The solid line represents the mean and the shaded region represents the standard deviation.

While the Cart-Pole environment uses a continuous state space it restricts the agent to a small discrete action space. We therefore explored whether CTDL could be applied to problems that require a continuous action space as well as state space. To this end we applied CTDL to the Continuous Mountain Car environment from OpenAI Gym. The DNN used in both DQN and CTDL has a single output unit for each action that outputs the Q value for that particular action. This is infeasible for continuous control problems and so a different underlying network architecture is required. A common approach to combining value estimates with continuous control problems is to use an actor-critic framework (Sutton and Barto, 2018). Under the actor-critic framework, the ‘critic’ is responsible for calculating value estimates and the ‘actor’ is responsible for choosing actions and updating the policy based on the values estimated by the critic. The benefit here is that the critic can calculate state values rather than action values and the actor can output a continuous distribution over possible actions.

In our simulations we represent both the actor and critic components as feed-forward neural networks and adopt an Advantage Actor-Critic (A2C) approach (Mnih et al., 2016) (see Section 2.3.2 for more details). The hyper-parameters for our A2C implementation can be seen in Table 4.4. Our implementation of A2C shares many common properties with DQN in that it relies upon the slow learning of distributed representations. We therefore hypothesised that the fast pattern-separated learning of CTDL should confer the same advantages to A2C as it did to DQN.

Table 4.4: A2C hyper-parameter values used for the continuous mountain car simulations.

Parameter	Value	Description
L_{critic}	3	Number of layers in critic network
U_{critic}	[128, 128, 1]	Number of units in critic network
α_{critic}	.0001	Critic learning rate for Adam
L_{actor}	3	Number of layers in actor network
U_{actor}	[128, 128, 2]	Number of units in actor network
α_{actor}	.00001	Actor learning rate for Adam

In order to augment A2C with the advantages of CTDL we used the same approach as before except the SOM recorded state value estimates rather than action value estimates. The state value estimates of the SOM were combined with the esti-

mates of the A2C ‘critic’ using the same weighted sum calculation and the TD error from the ‘critic’ was used to update the weights of the SOM. A2C is inherently an online algorithm and so weight and value updates were simply applied at each time step in an online fashion with no replay or target networks. We shall denote the CTDL version of A2C as CTDL_{A2C} and a full outline of the algorithm can be seen in Algorithm 5. The hyper-parameters used for CTDL_{A2C} are the same as those in Table 4.3.

Algorithm 5 - CTDL_{A2C} . Highlighted lines are unique to CTDL_{A2C} when compared to A2C

Initialize SOM weights β according to a standard normal distribution

Initialize SOM state-values $V^{SOM} = 0$

Initialize Critic V^{A2C} with random weights θ^V

Initialize Actor π with random weights θ^π

for $e = 1, E$ **do**

for $t = 1, T$ **do**

 Observe current state s_t and reward r_t

 Retrieve SOM unit u_t that is closest to s_t

$u_t = \arg \min_u \|\beta_u - s_t\|^2$

 Calculate weighting η based on distance

$\eta = \exp(-\|\beta_{u_t} - s_t\|^2 / \tau_\eta)$

 Calculate $V(s_t)$ as weighted average of SOM and Critic values

$V(s_t) = \eta V^{SOM}(u_t) + (1 - \eta) V^{A2C}(s_t; \theta^V)$

$$\text{set } y_t = \begin{cases} r_t & \text{if episode is over} \\ r_t + \gamma V(s_t) & \text{otherwise} \end{cases}$$

 Calculate the advantage/TD error $A(s_{t-1}, a_{t-1}) = y_t - V^{A2C}(s_{t-1}; \theta^V)$

 Update the Actor parameters θ^π

$\theta^\pi \leftarrow \theta^\pi + \alpha_{actor} A(s_{t-1}, a_{t-1}) \nabla_{\theta^\pi} \log(\pi(a_{t-1} | s_{t-1}; \theta^\pi))$

 Update the Critic parameters θ^V

$\theta^V \leftarrow \theta^V + \alpha_{critic} A(s_{t-1}, a_{t-1}) \nabla_{\theta^V} V^{A2C}(s_{t-1}; \theta^V)$

 Calculate δ based on the TD error produced by the Critic

$\delta = \exp(|A(s_{t-1}, a_{t-1})| / \tau_\delta) - 1$

 Calculate the neighbourhood function based on u_{t-1}

$T_{u_j, u_{t-1}} = \exp(-\|l_{u_j} - l_{u_{t-1}}\|^2 / 2(\sigma_c + (\delta * \sigma)))$

 Update the weights β of SOM

$\Delta \beta_{ji} = \alpha * \delta * T_{u_j, u_{t-1}}(s_{t-1, i} - \beta_{ji})$

 Update the state value $\Delta V^{SOM}(u_{t-1}) =$

$\rho * \eta_{t-1} * (y_t - V^{SOM}(u_{t-1}))$

 Sample action from Actor $a_t \sim \pi(a_t | s_t; \theta^\pi)$

 If the goal has been reached then **break** and end episode

end for

end for

Figure 4.7 shows the results of A2C and CTDL_{A2C} on the Continuous Mountain Car task. CTDL_{A2C} outperformed A2C on the Continuous Mountain Car task and also demonstrated much greater stability as training progressed. The high variability in reward obtained is due to the fact that if the agent does not find the target location quickly enough then it will learn to minimise negative rewards by staying still. This suggests that the fast learning of pattern-separated representations in the SOM component of CTDL_{A2C} may allow the agent to either explore more efficiently or better utilise information about states that are rarely visited. In general these results suggest that the dual learning systems of CTDL_{A2C} are advantageous for problems consisting of continuous state and action spaces.

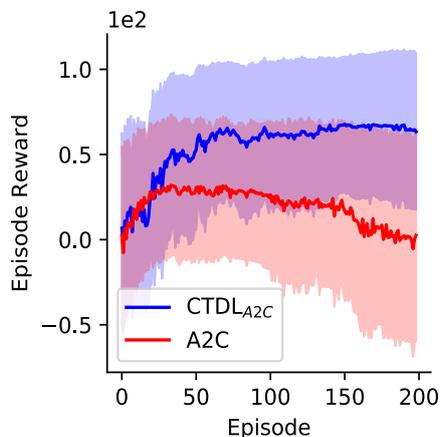


Figure 4.7: Episode reward achieved by CTDL_{A2C} and A2C on the Continuous Mountain Car environment over the course of learning. Both CTDL_{A2C} and A2C were run 50 times on the Continuous Mountain Car environment. The solid line represents the mean and the shaded region represents the standard deviation.

4.4 Neural Underpinnings

It is important to consider how our proposal maps onto real neural systems. As previously mentioned, the key components of CTDL are the independent contribution of a ‘hippocampal’ learning system to the evaluation of states and actions and the use of TD error to update representations in the ‘hippocampal’ learning system. With respect to the first of these components, it is well known that the striatum is a central location for updating and evaluating states and actions for decision making (Schultz et al., 1992; Houk et al., 1995; Schultz, 1998; Setlow et al., 2003; Roesch et al., 2009). Importantly, the striatum receives direct inputs from both cortical

areas and the hippocampus (Groenewegen et al., 1987; Thierry et al., 2000). It has also been proposed that pattern-separated hippocampal representations aid the reinforcement learning process (Duncan et al., 2018; Ballard et al., 2019). These findings lend support to the idea that a ‘hippocampal’ learning system may provide a value prediction that complements that of a ‘neocortical’ learning system. In addition, with converging cortical and hippocampal inputs, the striatum needs to be able to arbitrate between them in order to calculate a state or action value (Pennartz et al., 2011). The weighting process between the SOM and DNN components performed by CTDL may represent a simplified example of such an arbitration. Interestingly, CTDL predicts that the striatum should apply a greater weighting to hippocampal information when the current state closely matches one stored in episodic memory.

The second component of CTDL relies on the use of TD error to update both hippocampal value estimates and memory representations. A wealth of evidence currently suggests that the primary neural correlates of TD error are phasic dopamine neurons in the midbrain (Schultz et al., 1997; Schultz, 2016). One of the major projection sites of these neurons is the striatum and evidence of TD error has subsequently been found in the striatum (Doherty et al., 2003; McClure et al., 2003; Bray and Doherty, 2007). It therefore seems plausible that TD error can be used to update value estimates situated at hippocampal-striatal synapses that are independent of neocortical value estimates.

Nevertheless, the question remains whether TD error could modulate pattern-separated state representations in the hippocampus, as in CTDL. It has been widely reported that midbrain dopamine neurons project directly to the hippocampus and can influence synaptic plasticity in the hippocampus via Long-Term Potentiation (LTP) (Lisman and Grace, 2005; Lemon and Manahan-vaughan, 2006; Rosen et al., 2015). It is therefore believed that midbrain dopamine neurons can mediate the formation of episodic memories in order to guide memories towards experiences that are relevant for behavior (Shohamy and Adcock, 2010). This biasing of episodic memory towards reward-related experiences can take many forms. For example, reward cues appear to engage midbrain dopamine neurons, which then enhances episodic memory for those cues (Wittmann et al., 2005). Similarly, motivation to obtain future rewards also promotes the firing of midbrain dopamine neurons and

subsequent episodic memory of items, even in the absence of reward during learning (Adcock et al., 2006).

While these results lend support to the hypothesis that midbrain dopamine neurons can bias the formation of episodic memories in the hippocampus, they do not provide evidence that reward prediction errors, such as TD error, have an effect. Dopaminergic midbrain neurons are thought to encode many aspects of reward-related information such as reward outcome, expected reward, novelty and incentive salience (Shohamy and Adcock, 2010). Part of the ability of dopaminergic neurons to encode these different forms of information may lie in the differences between tonic and phasic dopamine responses. It is likely that many of the aforementioned effects on episodic memory are due to the tonic responses of dopamine neurons that encode reward-related information other than prediction errors (Shohamy and Adcock, 2010).

CTDL specifically predicts that reward prediction errors, as encoded by phasic midbrain dopamine neurons, should promote the formation of episodic memories in the hippocampus. Empirical support for such a prediction is beginning to emerge. In particular, a recent study by Rouhani et al. (2018) demonstrated that unsigned reward prediction errors enhance episodic memory for trial-unique images. When the reward outcome differed by a large amount from the participant’s subjective expected value of an image, the participant was better at recognising that image in a subsequent surprise recognition test. This effect was consistent even when controlling for reward outcome and subjective value estimates. The effect was also independent of sign (i.e. both large positive and negative reward prediction errors improved episodic memory for images that lead to the reward prediction error), which is consistent with CTDL as the algorithm uses the absolute value of the TD error to update the SOM. Interestingly, Rouhani et al. (2018) also found that when participants were presented with the same images again, they tended to choose the one that previously had the larger reward outcome. This suggests that they also encoded the rewards associated with the images. From the perspective of CTDL this could be seen as encoding the value of the episodic memory independently from the ‘neocortical’ learning system. Indeed, the result of the parameter sweep assigned a large value to the learning rate ($\rho = 0.9$, Table 4.3) for the Q values of the SOM,

perhaps reflecting a direct episodic encoding of the reward outcome rather than a running average.

Further evidence for the promotion of episodic memory via reward prediction errors comes from a study by Jang et al. (2018). In this study, participants had to decide whether to play or pass on a risky gamble. To make the decision participants were provided with information about the potential payout and an image from one of two categories, which they could use to incrementally learn the reward probability of that category. If the participants chose to play the risky gamble then the subsequent feedback was active, otherwise it was passive. Importantly, the authors showed that episodic memory for images was improved when reward prediction errors were large at the time of image presentation. This effect was only apparent for active as opposed to passive feedback, suggesting that it was dependent on decision-making. In addition, the effect was consistent regardless of whether the image recognition task was performed immediately after the decision-making task or 24 hours after. This suggests that the effect of reward prediction errors on episodic memory and subsequently decision-making are potentially fast acting and do not require consolidation mechanisms. Taken together these findings provide additional support for the modulation of episodic memory formation via reward prediction errors.

Reward prediction errors have also been proposed to have a role in the updating of long-term memories. The theory of memory reconsolidation posits that long-term memories which have been destabilised into a malleable form can be updated with new information to aid integration and avoid interference (Sara, 2000). This process is thought to be hippocampus-dependent (Debiec et al., 2002; Lee et al., 2004) and rely upon reward prediction errors from midbrain dopamine neurons to signal the need for integration of new information (Exton-mcguinness et al., 2015). Evidence for this comes from the fact that reconsolidation appears to decrease once a behavior becomes well-learnt, supposedly because reward prediction errors have decreased. This provides an example of how reward prediction errors, such as TD error, may be able to modify existing memories via the hippocampus; a process that is critical to CTDL.

While the aforementioned studies demonstrate that reward prediction errors, such as TD error, can modulate episodic memory, it is worth noting that these find-

ings are not unanimous. Most notably an fMRI study by Wimmer et al. (2014) found a negative correlation between striatal reward prediction errors and performance on an episodic memory task. The authors suggested that the formation of episodic memory interferes with classical reinforcement learning by disrupting reward prediction errors. Such a finding argues against the prediction of CTDL that TD error should promote the formation of episodic memories, however criticisms of the study have been highlighted. In particular, the trial-unique images were unrelated to the actual reward learning task and so the reward prediction errors may not have been elicited by the images themselves. In comparison, the studies by Rouhani et al. (2018) and Jang et al. (2018) both required participants to use the images to perform the decision-making task. In addition, it would have been interesting to see the results of the correlation analysis using absolute or unsigned reward prediction errors. Both CTDL and Rouhani et al. (2018) predict no linear relationship between signed reward prediction errors and episodic memory performance. Nevertheless, such confounding findings highlight the need for further empirical work to elucidate the role of reward prediction errors in the formation of episodic memories.

In reality, it is likely that a combination of reward-related information, encoded by midbrain dopaminergic neurons, is responsible for the dynamic and selective episodic memory present in humans. For example a study by Mason et al. (2017) explored the effect of different reward-related effects on episodic memory. In the study participants had to learn a collection of words in exchange for monetary rewards, thereby creating a motivated learning scenario. The authors found that reward outcome i.e. a combination of expected value and reward prediction error, was the best predictor of episodic memory. This demonstrates that reward prediction error on its own may not provide a holistic account of episodic memory formation. Taking this into account, we believe that CTDL provides a useful initial framework for exploring the effect of other reward-related signals on the formation of episodic memories and subsequently goal-directed behavior.

In addition to the algorithmic components, the behavior demonstrated by CTDL has several interesting parallels with biological findings. Firstly, the ‘hippocampal’ learning system (i.e. the SOM) of CTDL appears to encode violations of the generalisations made by the DNN. In the case of the grid worlds this corresponded

to regions close to obstacles. This finding has an interesting parallel with imaging work in rodents demonstrating that CA3 neurons appear to encode decision points in T-mazes that are different from the rodents current position (Johnson and Redish, 2007). Such decision points could be viewed as obstacles or important deviations from the animals general direction. Their encoding by the hippocampus is therefore consistent with being encoded by the SOM component of CTDL. In the future, application of CTDL to other reinforcement learning tasks may make testable predictions about the regions of the state space that should be encoded by the hippocampus. Another key behavior demonstrated by CTDL was increased flexibility when presented with a change in the environment i.e. the introduction of an obstacle. This was due to the ability of the SOM to quickly encode the states close to the obstacle. This suggests that the hippocampus may be important for adapting to changes in the environment and is consistent with recent studies that have implicated the hippocampus in reversal learning (Dong et al., 2013; Vila-Ballo et al., 2017).

Aside from ‘what’ should be encoded by the hippocampus, CTDL is also consistent with biological theories regarding the ‘duration’ of encoding. It has been proposed that memories stored in the hippocampus are consolidated to the neocortex over time via mechanisms such as replay (Olafsdottir et al., 2018). Interestingly, this process should naturally occur in CTDL; as the neural network improves its ability to evaluate the optimal value function the TD errors should reduce in magnitude and free up the SOM to represent other episodic memories. If part of the environment changes then a new episodic memory will form based on the TD error and it will remain in episodic memory until the ‘neocortical’ learning system has learnt to incorporate it. This process suggests that the transfer of information from the hippocampus to the neocortex is very much related to the ‘need’ for an episodic memory as encoded by TD errors.

4.5 Discussion

According to CLS theory, the brain relies on two main learning systems to achieve complex behavior; a ‘neocortical’ system that relies on the slow learning of dis-

tributed representations and a ‘hippocampal’ system that relies on the fast learning of pattern-separated representations. Both of these systems project to the striatum, which is believed to be a key structure in the evaluation of states and actions for RL (Schultz et al., 1992; Houk et al., 1995; Schultz, 1998; Setlow et al., 2003; Roesch et al., 2009). Current deep RL approaches have made great advances in modelling complex behavior, with DNNs sharing several similarities with a ‘neocortical’ learning system. However these approaches tend to suffer from poor data efficiency and general inflexibility (Lake et al., 2017). The purpose of the present study was to explore how a ‘neocortical’ and ‘hippocampal’ learning system could interact within an RL framework and whether CLS theory could alleviate some of the criticisms of deep RL.

Our novel approach, termed CTDL, used a DNN as a ‘neocortical’ learning system and a SOM as a ‘hippocampal’ learning system. Importantly the DNN used a small learning rate and distributed representations while the SOM used a larger learning rate and pattern-separated representations. Our approach is novel in that the SOM contributes to action value computation by storing action values independently from the DNN and uses the TD error produced by the DNN to update its state representations. More specifically, the TD error produced by the DNN is used to dynamically set the learning rate and standard deviation of the neighbourhood function of the SOM in an online manner. This allows the SOM to store memories of states that the DNN is poor at predicting the value of and use them for decision-making and learning. Importantly the size of the SOM is smaller than the state space encountered by the agent and so it requires less memory resources than the purely tabular case.

We compared the performance of CTDL to a standard DQN on a random set of 2D grid worlds. CTDL out-performed the DQN on the majority of grid worlds, suggesting that the inclusion of a ‘hippocampal’ learning system is beneficial and confirming the predictions of CLS theory. Removal of replay between the SOM and DNN appeared to have marginal impact upon the performance of CTDL suggesting that the SOMs contribution to the calculation of action values is the predominant benefit of CTDL. Future work should explore how information from the SOM may be replayed to the DNN in a more principled fashion (e.g. Mattar and Daw (2018))

instead of random sampling. We proposed that the SOM was able to contribute to the calculation of the action values in a targeted manner by using the TD error of the DNN to encode states that the DNN was poor at evaluating. We provided evidence of this by demonstrating that the removal of the TD signal between the DNN and SOM had a negative impact upon the performance of CTDL.

Our interpretation of these results is that, particularly early on in learning, the DNN is able to represent generalisations of the state space while the SOM is able to represent violations of these generalisations. In combination these two systems can then be used to formulate policies in both a general and specialised manner. We tested this hypothesis by presenting CTDL and DQN with a grid world consisting of a general rule and two other grid worlds consisting of violations of this rule. As our interpretation predicted, CTDL out-performed the DQN when violations of the general rule were present, presumably because the SOM was able to store states that were useful for circumnavigating these violations. This hypothesis was further supported by a simulation that ran both CTDL and DQN on sequential grid worlds. CTDL appeared to be better equipped to deal with the change in environment compared to the DQN. In addition, the SOM component of CTDL encoded states close to the change in the environment, providing further evidence of its ability to represent violations of predictions.

This ability of the SOM to encode violations of the generalisations made by the DNN has interesting parallels to imaging work in rodents demonstrating that CA3 neurons appear to encode decision points in T-mazes that are different from the rodents current position (Johnson and Redish, 2007). Such decision points can be viewed as obstacles or important deviations from the animals general direction and we therefore predict that they would be encoded by the SOM component of CTDL. In the future, application of CTDL to other reinforcement learning tasks may provide testable predictions about the regions of the state space that should be encoded by the hippocampus. In addition, the fact that the SOM encode states close to obstacles in order to account for changes in the environment suggests that the hippocampus may be important for adapting to changes in the environment and is consistent with recent studies that have implicated the hippocampus in reversal learning (Dong et al., 2013; Vila-Ballo et al., 2017).

To investigate the generality of CTDL we also applied it to the Cart-Pole and Continuous Mountain Car problems. The Cart-Pole problem is fundamentally different to the grid world problem because the state space observed by the agent is continuous. We found that in comparison to the DQN, the learning of CTDL was more gradual but also more robust. This is a surprising result given that the DQN has a perfect memory of the last 100,000 state transitions whereas CTDL has no such memory. Indeed, one would expect the SOM component of CTDL to have less of an effect in continuous state spaces because generalisation from function approximation becomes more important and the probability of re-visiting the same states decreases. That said, Blundell et al. (2016) demonstrated that even when the probability of re-visiting the same state is low, episodic information can still be useful for improving learning. Generalisation of episodic information in CTDL is likely controlled by the temperature parameter τ_η that scales the euclidean distance between the states and the weights of the SOM units.

The Continuous Mountain Car problem consists of both a continuous state and action space. In order to apply deep RL to the Continuous Mountain Car problem we used an A2C architecture, with two separate DNNs representing an actor and critic respectively. As with the original implementation of CTDL, we augmented A2C with a ‘hippocampal’ learning system in the form of a SOM and termed the resulting algorithm CTDL_{A2C} . CTDL_{A2C} both outperformed the standard A2C approach and demonstrated more robust learning with no substantial decreases in performance. A defining feature of the Continuous Mountain Car problem is that the agent will learn not to move unless it experiences the positive reward of the target location and then utilizes this information efficiently. It is possible that the addition of a learning system that quickly learns pattern-separated representations helps to alleviate this problem by storing rare and surprising events in memory and incorporating them into value estimates, rather than taking a purely statistical approach. More generally, these results demonstrate the applicability of CTDL to continuous control problems and further highlight the benefits of using TD error to inform the storage of episodic information.

The reduced benefit of CTDL on the Cart-Pole problem compared to the Grid World and Continuous Mountain Car problems may allude to interesting differences

in task requirements. In particular, both the Grid World and Continuous Mountain Car problems appear to rely on rare discrete events that are highly informative for learning a policy e.g. both tasks involve a goal location. In comparison, the Cart-Pole task relies on a range of rewarded events or states to inform the policy and so the utilization of episodic information may be less valuable. From a biological perspective, it is perhaps unsurprising that CTDL performs better on Grid World problems given that they represent spatial navigation tasks which are thought to heavily recruit the hippocampus in biological agents (Burgess et al., 2002). In comparison, the Cart-Pole problem can be seen as a feedback-based motor control task which involves learning systems such as the cerebellum in addition to any cortical-hippocampal contributions. CTDL may therefore represent a useful empirical tool for predicting the utilisation of hippocampal function in biological agents during RL tasks.

Future work will need to investigate whether the increased robustness and performance of CTDL in continuous state and action spaces is a general property that extends to more complex domains. In particular, it would be of interest to run CTDL on maze problems such as ViZDoom (Kempka et al., 2016), which are rich in visual information. Indeed, deep RL approaches using convolutional neural networks are at the forefront of RL research and these could be easily incorporated into the CTDL approach. In the case of ViZDoom, each state is represented by a high-dimensional image and so the generalisation capabilities of a DNN are crucial. From a biological perspective, it is worth noting that the hippocampus operates on cortical inputs that provide latent representations for episodic memory. This is captured in ‘model-free episodic control’, which relies on an embedding function to construct the state representation for episodic memory (Blundell et al., 2016; Pritzel et al., 2017). An embedding function therefore represents a biologically plausible method of scaling CTDL up to complex visual problems such as VizDoom. The embedding function could be pre-trained in an unsupervised manner or sampled from the DNN component of CTDL. We leave this interesting avenue of research to future work.

In addition to relatively low complexity, one consistent feature of the tasks presented in the present study was a low degree of stochasticity. As with discrete state spaces, low stochasticity means that events re-occur with high probability and the

episodic component of CTDL can exploit this. It is likely that in more stochastic environments the benefits of CTDL will be reduced as the DNN is required to generalise over several outcomes. It is therefore an open question how well CTDL will perform on tasks that have a high degree of stochasticity, which are also supposedly harder for biological agents.

One interesting element of CTDL that was not explored in the present study was the temporal evolution of pattern-separated representations in the SOM. Logically as the DNN improves its ability to evaluate the optimal value function its TD errors should reduce in magnitude and therefore free up the SOM to represent other episodic memories. If part of the environment changes then a new episodic memory will form based on the new TD error and it will remain in episodic memory until the ‘neocortical’ learning system has learnt to incorporate it. CTDL therefore suggests that the transfer of information from the hippocampus to neocortex is based to the ‘need’ for an episodic memory as encoded by TD errors. This can be viewed as a form of ‘consolidation’ whereby memories stored in the hippocampus are consolidated to the neocortex over time (Olafsdottir et al., 2018).

4.6 Conclusions

In this chapter we have proposed a novel algorithm for combining a neocortical and hippocampal learning system within a Reinforcement Learning (RL) framework. We show that this algorithm, termed Complementary Temporal Difference Learning (CTDL), demonstrates substantial benefits over just using a neocortical learning system to evaluate states and actions. We therefore believe CTDL represents a promising avenue for achieving complex, human-like behavior and exploring efficient RL within the brain. Importantly, CTDL incorporates the three main pathways highlighted by Complementary Learning Systems (CLS) theory (Figure 3.1). Firstly, both the ‘neocortical’ and ‘hippocampal’ system of CTDL contribute in parallel to the calculation of action values for decision-making. This reflects pathways 1 and 2 in Figure 3.1, where both systems are thought to project to the striatum in the brain. In the case of CTDL, the arbitration between these two systems is dependent on the memory content of the ‘hippocampal’ system. Secondly, CTDL updates

the contents of its ‘hippocampal’ system using the TD error from the ‘neocortical’ system. This allows the ‘hippocampal’ system to target regions of the state space that the ‘neocortical’ system is poor at evaluating or that violates generalisations made by the ‘neocortical’ system. Communication also flows back in the other direction in the form of replay from the ‘hippocampal’ system to the ‘neocortical’ system. These interactions reflect pathway 3 in Figure 3.1, which represents the connections between the neocortex and the hippocampus in the brain.

These key properties of CTDL represent directions for future research both computationally and empirically. From a computational perspective, it will be interesting to explore how embedding functions can be used to reflect the fact that the hippocampus receives latent representations from cortical areas as input. This may be a key component for scaling up CTDL to complex problems with high dimensional state spaces. With respect to future empirical work, CTDL can be used to make predictions about which tasks should utilize the hippocampus and which regions of the state space should be encoded by it. CTDL also predicts that TD errors should promote the formation of episodic memories in the hippocampus and so we highlight this as a key area for further investigation and clarification.

As a concluding remark, it is worth noting that CTDL only represents one small step in the direction of understanding efficient RL in the human brain. Chapter 3 highlighted many potential avenues for explaining the efficiency of human RL and it is likely that a combination of these approaches is required to fully understand it. For example, CTDL uses a SOM to represent a hippocampal learning system due to its fast updating of values and pattern-separated representations. However, we have already seen in Chapter 3 that the hippocampus may offer many other beneficial computational properties such as predictive representations (Dayan, 1993; Momennejad et al., 2017; Stachenfeld et al., 2017), recurrent similarity computation (Kumaran and McClelland, 2012) and cognitive maps (Whittington et al., 2019). We believe that these other properties are also key to efficient RL and future work should investigate whether they can be incorporated into the CTDL framework for further improvements in efficiency. We therefore emphasise that the presence of a ‘hippocampal’ learning system is likely to have additional benefits above and beyond those demonstrated by CTDL. Nevertheless, we believe that the work in this chapter

has demonstrated that a neocortical and hippocampal system working in parallel and communicating via Temporal Difference error are important for achieving efficient RL.

In the next chapter we propose that the standard view of CLS theory should be extended in order to understand efficient RL in the brain. More specifically, we suggest that a distinction should be made between the Pre-Frontal Cortex (PFC) and sensory cortices due to their fundamentally different roles in cognition. This highlights additional pathways that are important for supporting efficient RL and improves the analogy between the brain and Deep RL. As in Chapter 3 we shall review recent advancements in computational modelling based on the pathways involved in this extended CLS framework.

Chapter 5

Extending CLS Theory to Include Pre-Frontal Cortex

Overview

The purpose of this chapter is to review past literature and argue that Complementary Learning Systems (CLS) theory should be extended to reflect the differences between Pre-Frontal Cortex (PFC) and sensory cortices. This extension allows us to address several cognitive phenomenon that are associated with the PFC and that are important for efficient Reinforcement Learning (RL), such as meta-learning and selective attention. Our proposed extension highlights three more pathways that are important for efficient RL: connections between (1) the PFC and the striatum, (2) the PFC and the hippocampus, and (3) the PFC and sensory cortices. As before, we will consider each pathway individually and review recent advancements in Deep RL and computational modelling (Sections 5.2 - 5.4). Our hope is that this chapter will help to further elucidate the mechanisms underlying efficient RL in the brain and also highlight where the analogy between Deep RL and the brain could be further improved.

5.1 Pre-Frontal Cortex as an Additional Learning System

Complementary Learning Systems (CLS) theory emphasises the role of two learning systems in the brain; the neocortex and the hippocampus. Both of these learning systems project to the striatum (Pennartz et al., 2011), which is thought to play a key role in Reinforcement Learning (RL) by evaluating states and/or actions (Schultz, 1998; Houk et al., 1995; Joel et al., 2002; Maia, 2009; Setlow et al., 2003). In Chapter 4, we demonstrated the benefits of utilising both systems to evaluate states/actions and that the two systems could communicate via Temporal Difference (TD) errors. However, this still represents a major simplification of the different learning systems in the brain. Most notably, the distinction between different neocortical areas has largely been ignored by CLS theory. In particular, the distinction between sensory cortices and the pre-frontal cortex (PFC) has been well characterised empirically and yet the two structures are aggregated under CLS theory. The PFC has long been implicated in executive function and is therefore likely to be an important component of learning and transfer. From an RL perspective, empirical evidence linking the PFC with reward-based learning is gradually emerging. For example, the PFC has been shown to encode the expected value of states, objects and actions (Rushworth and Behrens, 2008; Seo and Lee, 2008; Padoa-Schioppa and Assad, 2006) as well as a history of previous actions and rewards (Seo et al., 2012; Barraclough et al., 2004); all hallmarks of an RL algorithm.

The purpose of this chapter is to argue that a dissociation between the PFC and sensory cortices is important for understanding efficient RL and that CLS theory should be extended to recognise this. Figure 5.1 provides a graphical depiction of this extension to CLS theory. The striatum serves as the primary locus for evaluating states and/or actions via Reinforcement Learning (RL) while the other three learning systems provide the representations for evaluation. In total there are 6 pathways, with Pathways 1, 2 and 3 already discussed in Chapter 3. Pathways 1, 2 and 4 provide direct connections between each of the learning systems and the striatum while Pathways 3, 5 and 6 provide connections between each of the different learning systems. We believe that this network of interactions is required for efficient RL and

that empirical examples of rapid learning and transfer rely on different combinations of these learning systems. As with Chapter 3, the remainder of this chapter will review advances in Deep RL and computational modelling based on whether they relate to Pathway 4, 5 or 6. We hope that this will serve as evidence for the need to treat the PFC as its own distinct learning system with specific computational properties.

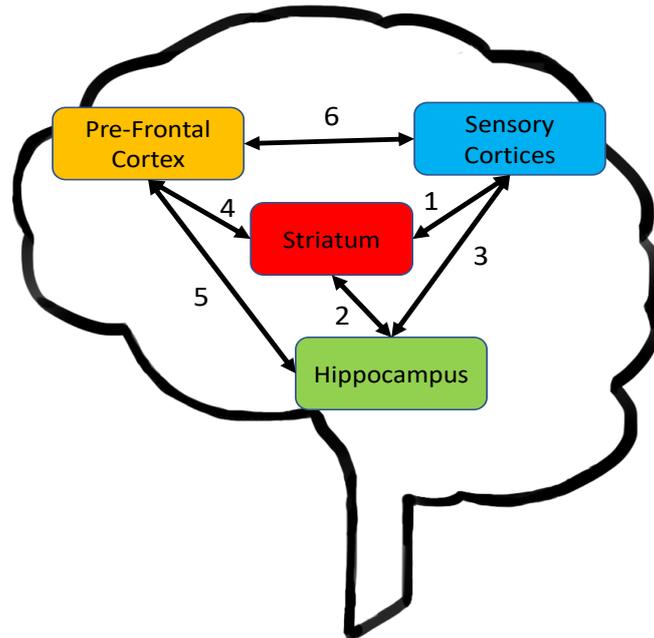


Figure 5.1: *Diagram extending our framework to include the distinction between Pre-Frontal Cortex (PFC) and Sensory Cortices. This introduces three additional pathways; (4) projections from the PFC to the striatum, (5) projections between the PFC and the hippocampus and (6) projections between the PFC and sensory cortices.*

5.2 4. Connections Between the Pre-Frontal Cortex and Striatum

A wealth of evidence has implicated the Pre-Frontal Cortex (PFC) in reward-based learning. In particular several signals associated with Dopamine (DA)-driven Reinforcement Learning (RL) have been found in the PFC, such as state, object and action values (Rushworth and Behrens, 2008; Seo and Lee, 2008; Padoa-Schioppa and Assad, 2006) and a history of actions and rewards (Seo et al., 2012; Barraclough et al., 2004). These findings suggest that the PFC may implement an RL algorithm

in collaboration with midbrain dopamine neurons and the striatum (Pathway 4 in Figure 5.1). A classical interpretation of these results is that reward-prediction errors encoded by DA are responsible for model-free RL whereas activity in the PFC is responsible for model-based RL (Daw et al., 2005). Model-based RL is important for transfer because the ability to use a world model in new situations can lead to inferences with little or no experience in the new situation. It has been suggested that the PFC is capable of model-based RL because it is able to utilise representations of task structure to produce behaviour akin to planning. However recent findings have questioned this dissociation between model-free and model-based systems because DA reward-prediction errors appear to be influenced by task structure, suggesting that they reflect model-based value estimates. This therefore raises the question: what RL algorithm could the PFC be implementing and is it responsible for the model-based value estimates seen in DA reward-prediction errors? An answer to this question could help to imbue Deep RL agents with the ability to perform model-based RL, which could greatly improve their efficiency and ability to perform rapid learning and transfer.

5.2.1 Meta-Reinforcement Learning

Recent work by Wang et al. (2016, 2018) has begun to answer this question using an approach known as Meta-Reinforcement Learning (Meta-RL). The basic premise behind Meta-RL is that DA and the PFC represent two different RL algorithms, with one implemented in the activation dynamics of the PFC and the other implemented in the altering of synaptic weights in the PFC. The RL procedure implemented in the PFC’s activation dynamics is shaped by the altering of the synaptic weights in the PFC by DA reward-prediction errors. The result of this interaction is that learning from DA reward-prediction errors helps to improve the learning present in the activation dynamics. Such an interaction is often called ‘learning to learn’ as the learning of one system helps to improve the learning of another system. The reason this is interesting from a transfer perspective is that the DA-driven algorithm can operate over a distribution of tasks and tune the activation-driven algorithm to exploit similarities between tasks thereby transferring knowledge about general task structure. In other words, an agent is better at obtaining reward in a novel

environment because it has an RL procedure that is biased towards solving similar, related tasks efficiently.

In order to implement Meta-RL, Wang et al. (2016, 2018) used a Long Short-Term Memory (LSTM) network (Figure 5.2), which is a form of recurrent neural network with gating mechanisms for inputs, outputs and internal state. In brief, the activation dynamics of the LSTM network implements an RL algorithm that can solve a single task, while the modification of connection weights improves the ability of this RL algorithm to solve other related tasks. To train the network, a task was randomly sampled from a distribution of related tasks and weight updates were performed after a short amount of experience. A new task was then sampled and the process was repeated. As a result, the LSTM was able to extract similarities between tasks using the weight updates and then solve a novel task using only its memory and recurrent dynamics. The authors found that the network’s activation dynamics produced estimates consistent with model-based RL in a variety of simulations. They therefore concluded that the RL algorithm implemented by the network’s activation dynamics can be qualitatively different to the one used for updating the weights, which was purely model-free.

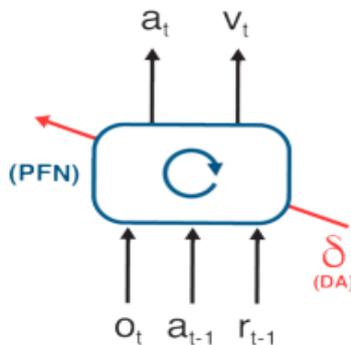


Figure 5.2: *Meta-RL agent architecture used by Wang et al. (2016, 2018). The network was a simple LSTM with the current observation o_t , the previous action a_{t-1} and the previous reward r_{t-1} as inputs. The outputs of the LSTM were the chosen action a_t and the predicted state value v_t . The weights of the LSTM were updated according to a model-free RL algorithm known as Advantage Actor Critic. Figure adapted from Wang et al. (2018).*

Of most importance to this thesis is the efficiency of Meta-RL on novel tasks. Wang et al. (2016) applied their Meta-RL algorithm to a virtual replication of a study by Harlow (1949) (Figure 5.3). In the original Harlow (1949) study, primates

were shown two objects with a food item underneath one of the objects. Periodically these objects would be replaced by two new objects and a food reward would again be placed under one of them. Upon a change in objects, the optimal strategy is to use the first trial to work out which object is rewarded and then to select that object repeatedly on subsequent trials. This is often referred to as ‘one-shot’ learning because only one piece of feedback is needed to infer the optimal strategy. It is also a form of meta-learning because it involves learning that the task structure consists of one object always being rewarded and another not being rewarded.

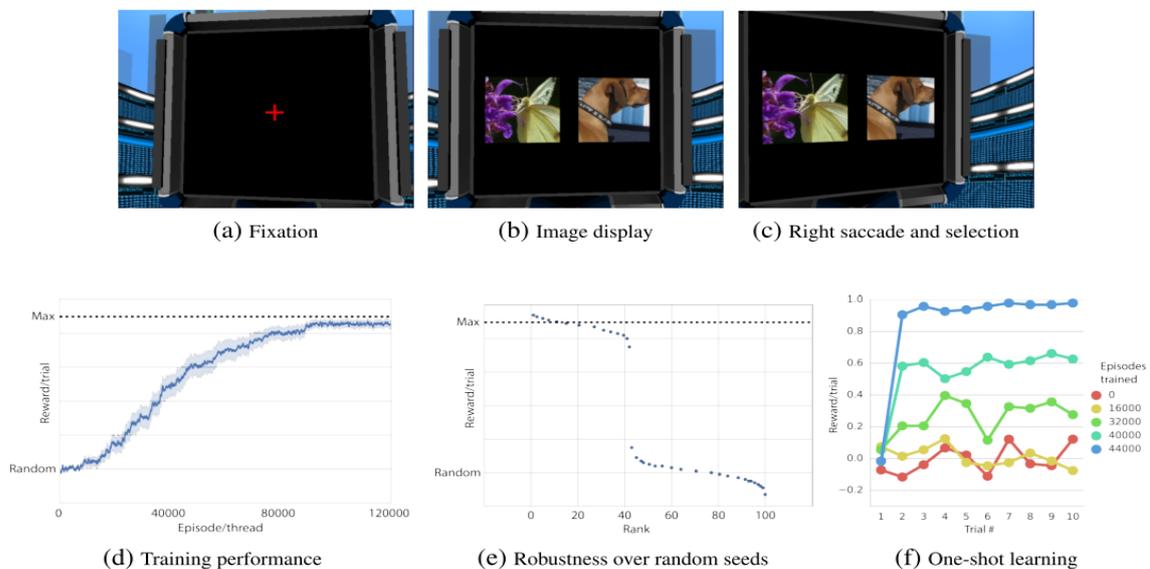


Figure 5.3: *The virtual Harlow task used to explore the ability of Meta-Reinforcement Learning (RL) to perform transfer. (A-C) The agent was shown a screen in a 3D environment with two images on it. The agent had to select the rewarded image on each trial. The location of the images was randomised on each trial. The agent made a decision by performing a saccade to the chosen image. (D) Average performance during training of the top 40 random seeds. (E) Performance of each random seed on episode 100,000. (F) Probability of selecting the image associated with reward over the course of learning. Figure adapted from Wang et al. (2016).*

In the virtual replication by Wang et al. (2016), the agent was provided with the pixel values of a screen that displayed a 3D representation of a world containing another screen which had two images on it. These two images represented the objects in Harlow’s original task and the agent had to perform a saccade to one of the images in order to choose it and receive any associated reward. The saccade itself caused the pixel values of the 3D world to change as if the agent was looking around the 3D world. As with Harlow’s original study, after training on one set

of images the agent was tested on another set of images. The Meta-RL algorithm proposed by Wang et al. (2016) successfully discovered a ‘one-shot’ strategy, thereby minimising the reward lost in the face of a new pair of images. Importantly, this demonstrates that the Meta-RL algorithm can learn the structure of a set of related tasks and use it to select the most efficient strategy for a new task.

The ability of Wang et al. (2016)’s Meta-RL approach to learn the structure of the Harlow experiment and infer an optimal policy from a single trial is a promising demonstration of efficient RL. This being said, one potential problem of Meta-RL is the lack of a mechanism for selecting appropriate past experiences to learn from. Meta-RL needs to match the current task with a related set of previous tasks, so that only useful prior knowledge is transferred. Unfortunately the work by Wang et al. (2016) circumvents this problem by only using related tasks for learning and testing. This means that the agent is only ever exposed to a single distribution of tasks and so it can use all the knowledge that it gains for transfer and no selection is required. If we compare this to the challenge faced by a human in a novel environment, we immediately see that one of the greatest challenges faced by the learner is the appropriate selection of past experiences for inference. Ultimately, the application of transferable knowledge is secondary to the problem of selecting which experiences to use for transfer.

In the closely related analogy literature, the problem of selecting appropriate experiences for transfer has also represented a significant challenge. Much of the work on analogy has involved explicit analogies, whereby a human participant is presented with two domains that are to be compared. In contrast, an implicit analogy involves just a single domain and it is up to the participant to retrieve from memory a suitable base domain for comparison. In transfer terms, the participant has to search past experiences in order to find the best match and analogical solution for the current problem. Some of the first work on analogy by Gick and Holyoak (1980) highlighted the difficulty of retrieving useful experiences for solving analogies. In this work they found that exposure to useful experiences only provided a small improvement in analogical reasoning and that a much larger improvement occurred when human participants were explicitly told that a past experience was useful for the task at hand. Subsequent work has demonstrated that the problem of retrieving

useful analogs appears to be harder for human participants than mapping or using the analogs once they have been retrieved (Gentner et al., 1993). The exact factors that affect the retrieval of past experiences for transfer or analogical reasoning are yet to be elucidated. In terms of domain retrieval at least, there appears to be a trade off between the surface similarity of objects and the underlying abstract relationships that determines whether a domain is retrieved for mapping and transfer (Wharton et al., 1994).

5.3 5. Connections Between the Pre-Frontal Cortex and Hippocampus

We saw in Chapters 3 and 4 how a hippocampal learning system is a central component of efficient Reinforcement Learning (RL). These chapters highlighted how the fundamental properties of the hippocampus and its interactions with the neo-cortex can promote rapid learning and transfer. However, in this chapter we argue that the PFC should be considered as its own learning system that is separate from sensory cortices. With this in mind, in this section we consider how specific interactions between the PFC and the hippocampus could improve the efficiency of RL by promoting rapid learning and transfer (Pathway 5 in Figure 5.1). Central to these interactions are two main ideas; (1) the PFC may support the retrieval of episodic memories for transfer based on the current task and (2) the PFC may support the formation of conceptual representations in the hippocampus by providing attentional signals. Both of these processes are fundamental to making accurate inferences in novel situations and help to demonstrate the importance of extending CLS theory to include the distinction between the PFC and sensory cortices.

5.3.1 Memory Recall

We saw in the previous section that one of the issues with Meta-RL was the lack of a mechanism for selecting related memories for learning and transfer. Similarly, in the field of analogical reasoning, the quality of an analogy will be highly dependent on the source domain that is chosen for comparison with the target domain, regardless of how effective the comparison or alignment process is. Both Meta-RL and analog-

ical reasoning therefore highlight the fact that context-dependent episodic memory retrieval is crucial for inferences in novel situations.

Within the field of Deep RL, several approaches have been suggested that address context-dependent retrieval of information from memory. For example, one line of research focuses on approaches that allow neural networks to learn to read from, and write to, an external memory store (Graves et al., 2014; Jaeger, 2016; Graves et al., 2016). Similarly, the models proposed by Blundell et al. (2016) and Pritzel et al. (2017) (see Section 3.3.1) learn latent representations that can be used to query episodic information based on similarity in representational space. These approaches all demonstrate how context-dependent memory retrieval can improve the efficiency of RL. Importantly, the extent of this improvement is likely to be dependent on the nature of the representations used for retrieving memories. For example, if the representations used for memory look-up represent relational information instead of perceptual similarities then the degree of transfer may improve because the comparison process will exploit the underlying relational structure (Gentner, 1983).

From the perspective of the brain, it has been suggested that interactions between the PFC and hippocampus (Pathway 5 in Figure 5.1) may be responsible for the retrieval of episodic memories based on the current context (Place et al., 2016; Eichenbaum, 2017; Dobbins et al., 2002). One metaphor that has been used to describe the interaction between the hippocampus and the PFC is that of railroads (Miller and Cohen, 2001). More specifically, the hippocampus has been described as laying down train tracks while the PFC switches between these tracks. In other words the hippocampus is responsible for forming memories while the PFC switches between them based on the current context. Empirical evidence for this analogy comes from studies that investigate the effect of PFC damage. Such damage is routinely associated with an inability in rodents, primates and humans to switch between tasks (Dias et al., 1996; Birrell and Brown, 2000; Rich and Shapiro, 2007; Ragozzino et al., 2003). One study by Navawongse and Eichenbaum (2013) investigated the effect of inactivating the PFC on representations in the hippocampus of rats. Rats were trained on two different spatial contexts with the same two objects present in each context but with the object-reward associations reversed. Navawongse and Eichenbaum (2013) found that certain neurons in the hippocampus

preferentially responded to a specific object in a specific context. However, upon PFC inactivation, these neurons lost this specificity and either became inactive or fired indiscriminately. This suggests that the neurons in the hippocampus lost their context-specific tuning and that the PFC is indeed required for filtering episodic memories based on the current context.

If interactions between the PFC and the hippocampus are indeed responsible for context-dependent retrieval of episodic memories, then this suggests that Deep RL models could benefit from utilising the computational properties of the PFC to model episodic memory retrieval. In the previous section we saw how the high recurrency of the PFC can be used for meta-learning and transfer in Deep RL models. A recent study by Ritter et al. (2018) has extended this idea to demonstrate how it can also be used to produce context-dependent episodic memory retrieval. The model, referred to as Episodic Metal-RL (EMRL), uses the same LSTM architecture as Meta-RL but also includes an episodic memory. The episodic memory corresponds to a table, where each entry is a pairing between the hidden state of the LSTM and the perceptual input that caused it. At the end of each trial EMRL takes a copy of the hidden state of the LSTM and the current perceptual context and appends them to the table. Then on each time-step EMRL finds the closest matching perceptual context in memory by finding the one that minimises the cosine distance between the current context. EMRL then re-instates the associated hidden state of the LSTM using a learnt gating function. This mechanism allows EMRL to recognize previously encountered experiences and retrieve previously learnt solutions. Interestingly, this form of re-instatement appears to parallel the finding that episodic memory retrieval in humans re-instates activity patterns that were present during the encoding of the original memory (Cohen and O'Reilly, 1996; Xiao et al., 2017; Hoskin et al., 2019).

Ritter et al. (2018) tested EMRL on bandit tasks and found that EMRL was able to exhibit both model-based and episodic learning as well as improved performance compared to standard Meta-RL. EMRL therefore demonstrates the utility of allowing a PFC-like system access to episodic memory for context-dependent episodic memory retrieval. However, it is worth noting that the perceptual contexts used by Ritter et al. (2018) were represented as hand-coded binary vectors. As a result EMRL is effectively given perfect information about whether two perceptual con-

texts match or not. As a result this does not solve the problem of learning how to match the current problem with a previously encountered problem.

5.3.2 Concept Formation

The previous section highlighted how interactions between the PFC and hippocampus may support the retrieval of episodic memories based on the current task. In addition to this phenomenon, it has been suggested that interactions between these two systems may support the formation of concepts. The ability to form concepts is critical to the brain's ability to make inferences in novel situations. By their very nature concepts extract commonalities across a set of experiences and allow for the categorisation of unseen exemplars. This categorisation subsequently has a large impact on how the brain interprets a new situation and ultimately selects actions. For example, even being able to categorise someone you have never seen before as a human being can have a large impact upon how you interact with them for the first time. A range of mechanisms have been proposed for the learning of concepts including the memorization of individual exemplars, rule abstraction and hypothesis testing, and slow perceptual learning (Ashby and Maddox, 2005; Mareschal et al., 2010). Importantly, recent work has begun to highlight the importance of interactions between the Pre-Frontal Cortex (PFC) and the hippocampus for learning concepts (Preston and Eichenbaum, 2013; Schlichting and Preston, 2016), which corresponds to Pathway 5 in Figure 5.1. In particular, it has been suggested that the PFC may guide attention during the learning of concepts towards relevant features that constitute a specific concept.

Many of the seminal models on category learning rely on selectively attending to stimulus dimensions that are relevant for a particular category (Kruschke, 1992; Love et al., 2004). This includes Supervised and Unsupervised STratified Adaptive Incremental Network (SUSTAIN) (Figure 5.4), which represents attention as a receptive field over a stimulus dimension. A category in SUSTAIN corresponds to a cluster with its own set of receptive fields. All the receptive fields for a given stimulus dimension have the same tuning but each cluster centers their receptive fields over different locations. The activation of a cluster is based on how closely a given stimulus matches the receptive fields of the cluster. When a new stimu-

lus is presented the clusters compete with each other and the one closest to the stimulus in representational space wins. The winning cluster can then predict an unknown stimulus dimension such as the category membership of the stimulus. If the prediction is incorrect then a new cluster is recruited, which allows SUSTAIN to dynamically adapt its complexity to the problem at hand. From an RL perspective, by clustering stimuli and using category membership as a state representation, the learning problem can be simplified and generalization improved (Niv, 2019).

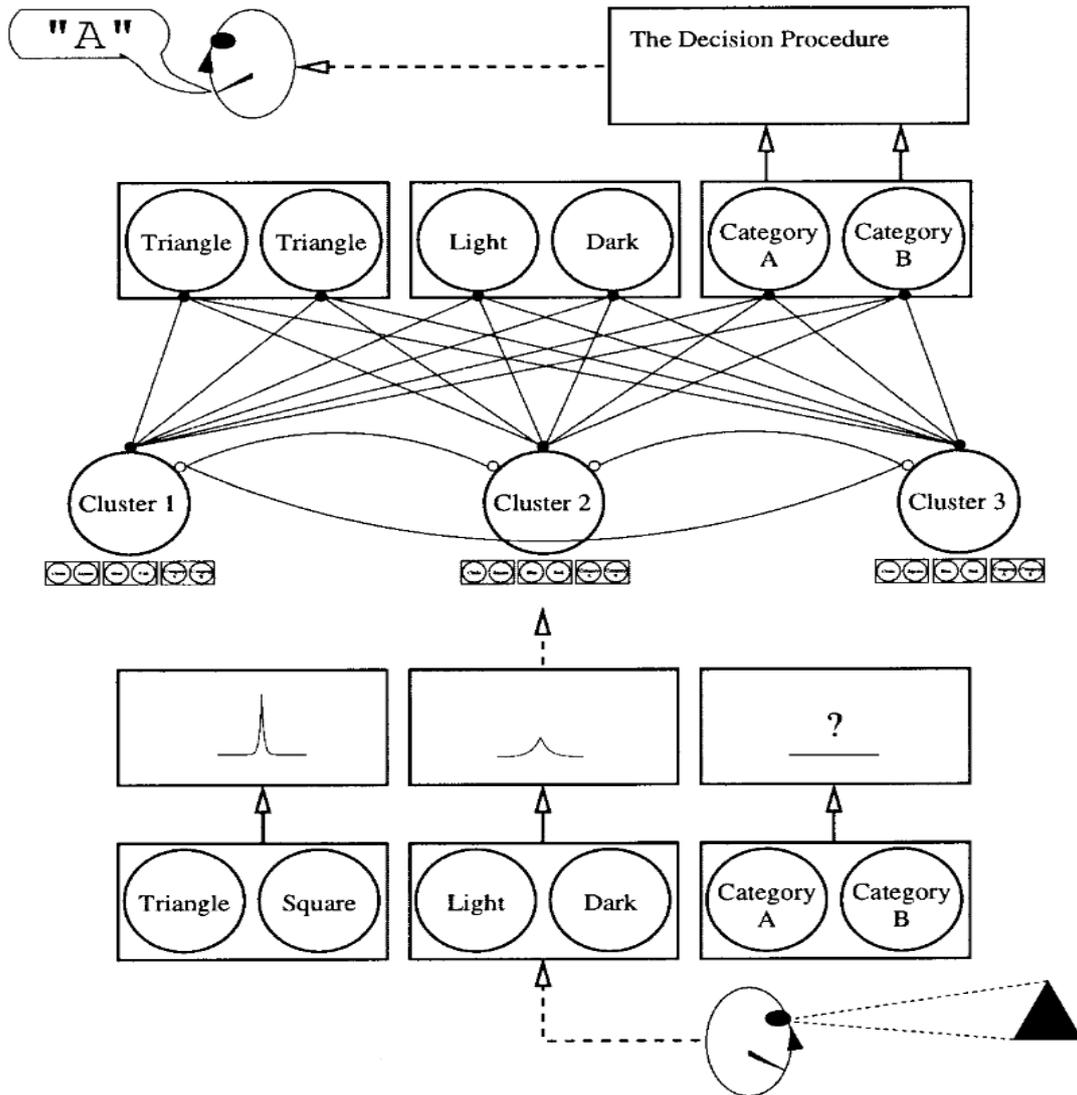


Figure 5.4: *Depiction of Supervised and Unsupervised STratified Adaptive Incremental Network (SUSTAIN). A stimulus is encoded along distinct dimensions with different possible values. Each dimension has a different receptive field tuning. Different clusters position their receptive fields in different locations for each stimulus dimension based on the category that they represent. The clusters compete with each other and the winning cluster is the one who's receptive fields are closest to the stimulus in representational space. The winning cluster is then used to infer the queried stimulus dimension, which in this case is category membership. Figure adapted from Love et al. (2004)*

Interestingly, increasing evidence is suggesting that interactions between the PFC and the hippocampus (Pathway 5 in Figure 5.1) may support the mechanism outlined by SUSTAIN. For example, a study by Mack et al. (2016) used functional Magnetic Resonance Imaging (fMRI) to explore how activity in the hippocampus and PFC related to Supervised and Unsupervised STratified Adaptive Incremental

Network (SUSTAIN) (Love et al., 2004). Participants were exposed to pictures of insects as stimuli and were asked to categorise them in two separate categorisation tasks; e.g. whether they liked warm or cold climates and whether they were found in the eastern or western hemisphere. The first categorisation task was based on a single feature while the second task relied upon an exclusive disjunction rule involving two features. Importantly, this experimental design allowed the stimuli to be kept constant but the relevant features and conceptual similarity to be changed over the course of the experiment. The experiment therefore promotes the need to quickly update ones conceptual representations based on the task at hand.

The performance of each participant was used to fit SUSTAIN at an individual level. This fitting process allowed Mack et al. (2016) to make predictions about how a participant represented each of the stimuli in vector space and how similar they perceived different stimuli to be. Using representational similarity analysis (Kriegeskorte et al., 2008), Mack et al. (2016) found that the predictions made by SUSTAIN best matched representations in the hippocampus. This suggests that the hippocampus plays a role in the construction of concepts and that its representations are updated as concepts change based on task demands. In addition to this correspondence between the predictions of SUSTAIN and hippocampal representations, Mack et al. (2016) also found that, particularly during early learning, the hippocampus demonstrated a strong functional coupling with the PFC. This suggests that interactions with the PFC are also important for the hippocampus to be able to update its conceptual representations.

From a mechanistic view, this dependence on the PFC for conceptual representations in the hippocampus may be due to the PFC's role in selective attention (Miller and Cohen, 2001). In particular SUSTAIN relies on a mechanism of selecting features that are predictive of a specific category across sequences of examples and it is possible that the PFC is responsible for this attention mechanism. This hypothesis has been backed by recent work by Mack et al. (2020), which used the same stimuli as the previous study (Mack et al., 2016) but included a third categorisation task where all three features are needed to categorise an insect. This resulted in three different categorisation tasks of increasing complexity. Mack et al. (2020) hypothesised that if the PFC is responsible for attending to the relevant feature dimensions

then its activation should be inversely related to the amount of compression needed. For example, when only one feature needs to be attended to then two features can be compressed whereas if all three features need to be attended to then none of them can be compressed. Mack et al. (2020) used fMRI to measure the activity in the PFC and measured the degree of compression in this activity using principle component analysis. Interestingly, the authors found that, after learning, the degree of compression in the activity of the PFC was inversely related to the number of features that needed to be attended to. This suggests that the PFC does indeed play a central role in attending to relevant features based on the current goal and that this ability supports the formation of abstract concepts.

5.4 6. Connections Between Pre-Frontal Cortex and Sensory Cortex

In the previous section we saw how interactions between the PFC and hippocampus could support the retrieval of context-appropriate episodic memories and the formation of concepts. One of the key roles of the PFC in concept formation was the ability to attend to features that were diagnostic of a particular concept. This notion of the PFC being responsible for attention is widespread and is not limited to interactions with the hippocampus (Desimone and Duncan, 1995; Miller and Cohen, 2001). In particular, the notion of ‘top-down’ or ‘selective attention’ is often used to refer to the ability of the PFC to filter incoming information from sensory cortices based on the the current goal (Desimone and Duncan, 1995) (Pathway 6 in Figure 5.1). The primary benefit of this top-down attention is that it simplifies the RL problem by identifying task-relevant features. From a transfer perspective, top-down attention identifies features from previous experience that are useful for the current task. This allows for both rapid learning and quick adaptation to changes in task because new features do not have to be learnt for each task. Different tasks can simply be associated with different patterns of attention. In terms of Deep RL, the DNNs used in Deep RL can be tentatively compared to sensory cortices. This suggests that the addition of a learning system like the PFC that performs selective attention on the features of DNNs could greatly improve the efficiency of Deep RL

algorithms.

5.4.1 Selective Attention

Animals and humans constantly face highly complex environments and problems that involve many dimensions. This greatly increases the complexity of the learning problem and many RL algorithms either struggle to solve such problems or are extremely slow to solve them in comparison to biological agents. One potential explanation for this is that biological agents rely on ‘top-down’ or ‘selective’ attention. Selective attention involves focusing on or prioritising dimensions/features that are useful for the task at hand and is thought to originate in the Pre-Frontal Cortex (PFC) (Desimone and Duncan, 1995). Selective attention greatly simplifies the problem faced by the brains RL machinery because it reduces the dimensionality of the environmental state so that the remaining dimensions are task relevant. At its core, transfer involves selecting information from past experiences to guide decisions in novel situations. Transfer can therefore be viewed as a problem of selective attention; knowledge from past experiences is selected from based on its utility for the current task.

Selective attention effects both choice and learning in RL. With regards to choice, instead of weighting all dimensions equally, selective attention can differentially weight certain dimensions when performing value computation. For example, experiments investigating RL over multidimensional states have demonstrated that people are able to alter their state representations based on the task at hand. People appear to be able to adopt an object-based representation (each combination of features is evaluated) or a feature-based representation (each feature is evaluated and then combined) based on which one is more predictive of reward (Farashahi et al., 2017).

As with disentangled representations (see Section 3.2.3), the effect of selective attention on choice can be reduced to a problem of representation; i.e. learning a useful state representation for the RL machinery to act upon. A natural question that arises is how does selective attention differ from standard learning procedures? For example, instead of selective attention one could just learn a new representation that ignores irrelevant dimensions. Selective attention differs from standard learn-

ing because it *acts on existing representations to reduce the dimensionality of the problem* and as a result the learning of new representations is not required. This improves the flexibility of learning because existing representations are left unperturbed for future tasks. If the current task changes then selective attention need only shift, while the representations it acts upon remain the same. Importantly, this means that different sets of attentional weights can be used to switch between different tasks. This allows the PFC to utilise different sub-networks to quickly refactor existing representations for different task demands. In comparison, the learning of new perceptual representations is thought to occur in early sensory cortices and does not change (or changes very slowly) in response to task demands (Schyns et al., 1998), allowing them to retain a degree of generality.

In addition to choice, selective attention also effects learning in RL. Selective attention can bias learning towards certain dimensions by updating their values more given a certain reward prediction error. This leads to much faster and efficient learning as only the dimensions that matter are updated and the influence of other dimensions is negated. Interestingly, this effect on learning creates a bi-directional relationship between selective attention and learning. On the one hand selective attention biases learning from reward prediction errors and on the other learning is required to learn which dimensions to attend to. How the brain learns which dimensions to attend to is still an open question. Two main theories have been proposed; 1) the brain learns to attend to features that are most predictive of reward (Mackintosh, 1975), 2) the brain learns to attend to features that it is most uncertain about i.e. features that generate a lot of reward prediction errors (Pearce and Hall, 1980). Several studies have suggested that both of these proposals may be true although it is under contention whether both processes occur independently (LePelley and McLaren, 2004; Pearce and Mackintosh, 2010) or in combination (Esber and Haselgrove, 2011; Nasser et al., 2017). Equally, some studies argue that the balance between the two processes may be different between choice and learning. In particular, Dayan et al. (2000) has suggested that during choice it makes sense to attend to the features most predictive of reward whereas during learning it makes sense to attend to the features with the most uncertainty.

What empirical evidence is there for a relationship between selective attention

and RL in the brain? A study by Leong et al. (2017) has demonstrated the effects of selective attention on RL in humans using a combination of computational modelling and functional magnetic resonance imaging (fMRI). In the study participants were given a simple RL task where they had to select one of three stimuli. Each stimuli was made up of three dimensions (a face, landmark and tool) and for each block of questions (25 questions), only one of the dimensions (e.g. tool) was relevant for predicting reward based on a target feature (e.g. wrench). The target feature was associated with a high probability of reward ($p=0.75$) while all other features were associated with a low probability of reward ($p=0.25$). Participants had to use trial and error to identify the target feature and obtain a high rate of reward.

While participants performed this task, Leong et al. (2017) conducted eye-tracking and fMRI. Eye-tracking was used to quantify selective attention based on the proportion of time spent looking at each dimension. Similarly, fMRI was used to quantify selective attention based on the amount of face-, tool- and landmark-specific neural activity. The choices made by each participant, along with their measures of selective attention were used to fit a variety of RL models to see which one best described the experimental data. After fitting each model, feature values were summed to produce a stimulus value, which was then input into a softmax function to calculate choice probabilities. The feature values were learnt using the trial-by-trial reward prediction error. The differences between models came from the influence of selective attention on choice and learning. In total there were four different RL models; Uniform Attention (UA) where all stimulus dimensions were equally weighted at choice and learning, Attention at Choice (AC) where dimension weights were based on estimates of attention when calculating stimulus value, Attention at Learning (AL) where dimension weights were based on estimates of attention when updating feature values, and Attention at Choice and Learning (ACL) where dimension weights were based on estimates of attention for both calculating stimulus value and updating feature values (Figure 5.5).

choice $V \begin{pmatrix} \text{Person} \\ \text{Spoon} \\ \text{City} \end{pmatrix} = \Phi_F V \begin{pmatrix} \text{Person} \end{pmatrix} + \Phi_L V \begin{pmatrix} \text{City} \end{pmatrix} + \Phi_T V \begin{pmatrix} \text{Spoon} \end{pmatrix}$

prediction error $\delta = \underset{\text{point}}{\overset{\text{YOU WIN}}{1}} - V \begin{pmatrix} \text{Person} \\ \text{Spoon} \\ \text{City} \end{pmatrix}$

learning $v_{\text{new}} \begin{pmatrix} \text{Person} \end{pmatrix} \leftarrow v_{\text{old}} \begin{pmatrix} \text{Person} \end{pmatrix} + \eta \cdot \delta \cdot \Phi_F$

Figure 5.5: *Effect of attention during choice and learning for the Reinforcement Learning (RL) models used by Leong et al. (2017). At choice, the value of a stimulus was calculated as the sum of the features values, which were weighted by their corresponding attention weights. During learning, the prediction error was weighted by another attention weight. Depending on the model, the attention weights were based on estimates from empirical data or set to 1/3. Figure adapted from Leong et al. (2017).*

Using a cross-validation procedure and metrics that penalised model complexity (e.g. Bayesian Information Criterion), Leong et al. (2017) found that the ACL model fit the behavioural data the best. This suggests that selective attention does indeed modify both choice and learning during human RL. To further test this hypothesis Leong et al. (2017) also investigated whether the effect of selective attention at choice and learning could be seen in the neural data as well as the behavioural data. To achieve this, Leong et al. (2017) entered the trial-by-trial value estimates of the chosen stimuli for each of the four RL models into a generalized linear model. This allowed the authors to look for brain regions that correlated with the value estimates while controlling for the value estimates of the other models. Leong et al. (2017) found that activity in the ventro-medial PFC, which has consistently been associated with expected value signals, was only significantly correlated with the value estimates from the ACL model. The same analysis was repeated for the reward prediction error estimates of the four models. In this case, only the reward prediction error estimates of the ACL model correlated with activity in the striatum, which is commonly associated with reward prediction errors. Taken together, these

results provide neural evidence that selective attention modifies both choice and learning during RL.

As previously mentioned, it has been proposed that the relationship between selective attention and learning in RL is bi-directional; selective attention biases RL value updates while learning guides selective attention towards important features. To explore the effect of learning on selective attention Leong et al. (2017) looked at the trajectory of selective attention over the course of learning. More specifically, Leong et al. (2017) calculated the standard deviation of the three attentional weights from the ACL model on each trial. A high standard deviation corresponds to highly selective attention whereas a low standard deviation corresponds to uniform attention. The authors found that attention became more selective and consistent over the course of learning as participants learnt to focus on the diagnostic dimension. In addition, attention was directed towards the dimension with the highest feature value, particularly when attention was more selective and when the difference in values was larger. Similarly, Leong et al. (2017) found that attention was more likely to switch from one dimension to another when the difference between expected values was lower. Finally the authors conducted a model-based analysis and found that the attention measures were best explained by a model that used feature values as opposed to choice or reward history for calculating attention. These results suggest that values learnt by participants modify the dimension attended to, the strength of attention and when a switch in attention occurs.

While the work of Leong et al. (2017) provides both behavioural and neural evidence for the dynamic relationship between attention and RL, the biological and computational mechanisms underlying the relationship are still to be elucidated. In a step towards a biological explanation Radulescu et al. (2019) proposed a neural model that attempts to explain how RL and selective attention may interact in the brain (Figure 5.6). The model proposes that cortical pools found in the PFC may be responsible for selective attention by representing hypotheses about the current task structure. These cortical pools use lateral inhibition to compete with each other and the inhibition is governed by the strength of connections between each pool and the striatum. The stronger a pool's connectivity with the striatum the stronger the striatal response and the stronger the thalamic feedback onto the cortical pool.

The result of this is that unexpected rewards cause a positive reward prediction error that preferentially strengthens synapses from the most active cortical pool. This eventually leads to one pool inhibiting the other pools and a ‘winner takes all’ scenario. The winning cortical pool performs selective attention by potentiating the responses of a subset of features in sensory cortices so that the corresponding striatal synapses are preferentially altered by reward prediction errors. The result of this is that RL is biased towards a subset of features that are relevant for decision-making. Features that are present in the environment but that aren’t part of the winning hypothesis will still be reinforced to some extent and this will increase the likelihood that associated rules will be chosen in the future. This neurobiological explanation for selective attention demonstrates how reinforcement learning can both affect, and be affected by, selective attention.

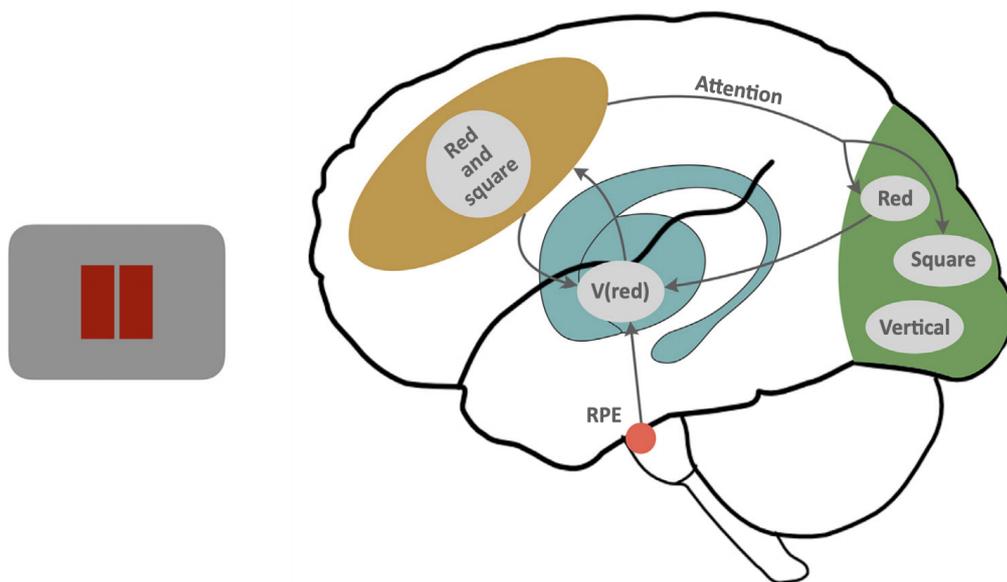


Figure 5.6: *Neurobiological model of selective attention by Radulescu et al. (2019). The hypothesis that ‘red and square’ is the correct categorisation rule is represented in the Pre-Frontal Cortex (PFC) (yellow). This provides top-down signals to sensory cortices (green) and biases attention towards features corresponding to ‘red’ and ‘square’. This influences the calculation of values in the basal ganglia (blue) and the updating of values using Reward Prediction Errors (RPEs). Subsequently, this completes the cycle by influencing the selection of hypotheses in the PFC. Figure adapted from Radulescu et al. (2019).*

While Radulescu et al. (2019) propose a neurobiological mechanism for selective attention within an RL framework, a computational implementation is still to be elucidated. Outside of an RL setting recent work has started to explore how selective

attention from the PFC to visual areas could be implemented. Of note, Luo et al. (2020) combined ‘top-down’ attentional weights with a Deep Convolutional Neural Network (DCNN) to model how selective attention can bias already existing representations towards a particular task. Attentional weights ($w \in [0, \infty]$) were used as a filter for the feature maps of a DCNN so that only information from a subset of the features maps was passed to deeper layers of the network (Figure 5.7). The attentional weights were trained using a modified cost function, which increased the cost of wrongly classifying examples that were pertinent to the task at hand. The task consisted of correctly predicting a target ImageNet class; e.g. correctly classifying images of cats. The trade-off between correctly classifying the target class as opposed to the other classes was governed by a parameter (α) that represented the attention intensity.

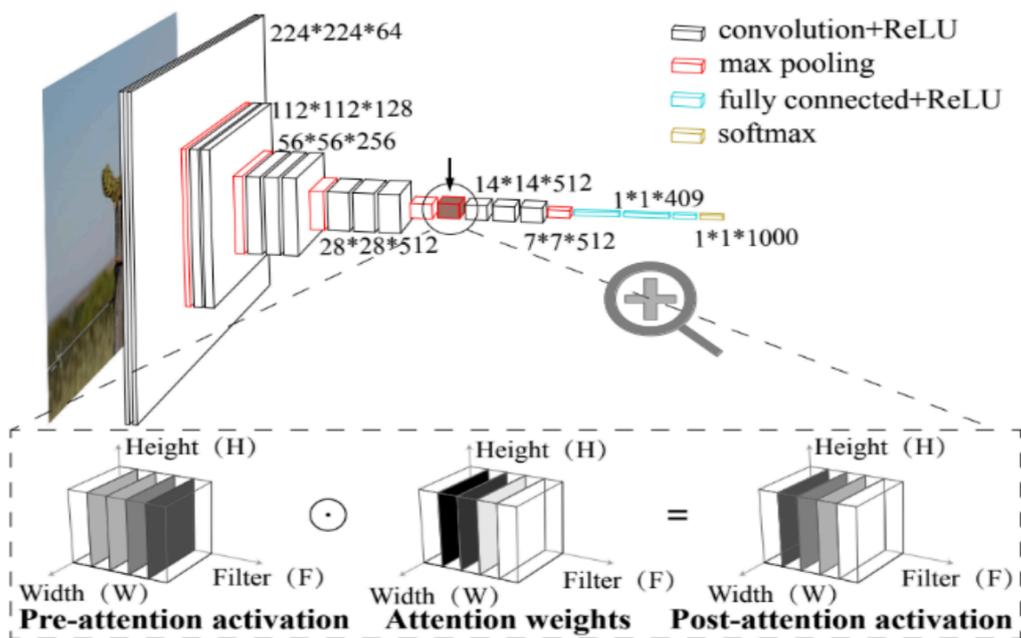


Figure 5.7: Architecture of the approach proposed by Luo et al. (2020). An attention layer (dashed box) is inserted into the architecture of VGG-16. Attention weights are applied to the feature maps produced by VGG-16 using element-wise multiplication. This serves to re-weight the activation values before they are passed on to the rest of VGG-16. Figure adapted from Luo et al. (2020).

Luo et al. (2020) found that there was a balance between the benefits and costs of selective attention. As attention intensity increased, there were increases in both true positives and false positives with respect to the target class. Correspondingly,

the sensitivity index (d') was highest for moderate levels of attention. These results demonstrate how a pre-trained network can be re-purposed for a particular task using a simple attentional mechanism that is driven by the current goal. The authors propose that this approach serves as a model of top-down attention, with the attentional weights modelling the output of the PFC and the DCNN modelling the ventral visual stream.

While the work of Luo et al. (2020) has taken a promising first step in the direction of modelling selective attention between the PFC and sensory cortices, it remains an open question how this would work in an RL setting. The approach suggested by Luo et al. (2020) involved a weighted cost function where the target class was pre-specified. Unfortunately, in an RL setting it is unclear which predictions are most salient and how the prediction error (whether it be value or policy based) relates to selective attention. In an ideal RL scenario, an agent would attend to dimensions of the state space that are most predictive of future reward, rather than dimensions most diagnostic of a category.

In the field of artificial intelligence, one recent study by Mott et al. (2019) has explored an approach that utilises top-down attention in an RL setting (Figure 5.8). The approach consisted of a collection of neural networks, whereby a top-level LSTM actively queried a DCNN for visual information that it deemed relevant for predicting rewards. This process relied on the generation of attentional maps that were used to weight information provided by the DCNN. These attentional maps were calculated by taking the inner product between a ‘query’ vector and a ‘keys’ tensor. The ‘keys’ tensor was generated by the DCNN and the ‘query’ vector was generated using information from the LSTM. This allowed the top-level LSTM to attend to both content (the ‘what’) and spatial location (the ‘where’). Importantly, the whole agent architecture was fully differentiable and therefore amenable to gradient-based optimisation.

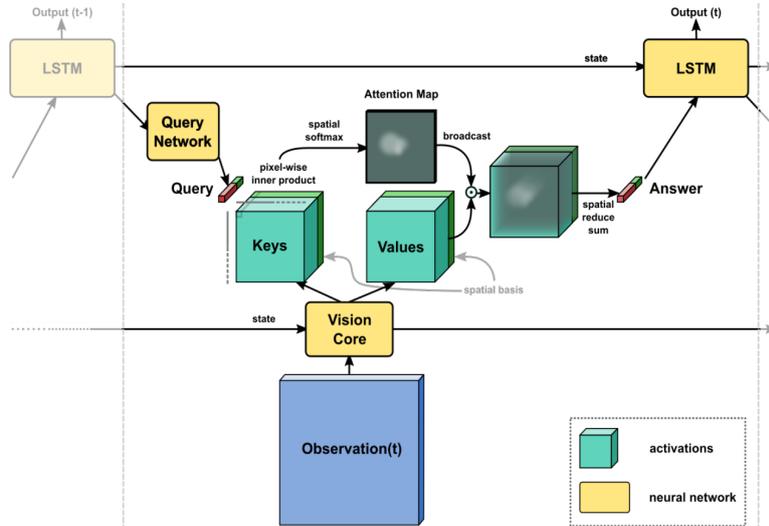


Figure 5.8: Architecture of the approach proposed by Mott et al. (2019). Observations are converted into key and value tensors using a recurrent Convolutional Neural Network (CNN). A Long Short-Term Memory (LSTM) network takes the result of the previous time step and provides an input to a third network, which is responsible for generating a query tensor. The inner product between the query and key tensors is passed through a soft-max function to produce an attention map. This attention map is multiplied element-wise with the values tensor to produce an answer tensor, which is passed to the LSTM for action selection. Figure adapted from Mott et al. (2019).

Mott et al. (2019) tested the agent architecture on the ATARI learning environment, which consists of a variety of computer games consisting of rich visual input. The authors found that their approach was competitive with state-of-the-art baselines without attention. Interestingly, the attention based architecture was not significantly better for every game and average performance across all games was only marginally better. When investigating the attention maps learnt by the agent, the authors found that the agent did indeed learn to attend to task-relevant features such as the player object and potential hazards, even when they were introduced unexpectedly. As a further analysis, Mott et al. (2019) compared their model to other approaches that utilise bottom-up attention (i.e. the vision network has control over the attention process) instead of top-down attention. Mott et al. (2019) found that top-down attention performed significantly better than bottom-up attention for all

but one game known as Seaquest. Interestingly, the authors note that Seaquest consists of a very stereotyped pattern of enemies and so they hypothesise that the requirement for top-down decision making is low.

The work of Mott et al. (2019) demonstrates the utility of a highly recurrent top-down module, that can selectively attend to different elements of visual perception, for solving the RL problem. With respect to the brain, it seems likely that the PFC may fulfill the role of such a module. Indeed, the degree of recurrency within the PFC is high (Mante et al., 2013; Wang et al., 2018) and the vision network proposed by Mott et al. (2019) could be viewed as modelling early visual sensory cortices that the PFC queries against and selectively attends to. However, several discrepancies exist between the agent architecture proposed by Mott et al. (2019) and selective attention mechanisms in the brain. In particular, the agent architecture proposed by Mott et al. (2019) is fully differentiable, and so all of the learnable parameters are updated via back-propagation. This means that all of the networks learn slowly and are dependent on each other. Most importantly the representations learnt by the vision network are dependent on the current task and the learning occurring in the top-down attention network. This is the opposite of the work by Luo et al. (2020) who used a top-down attention mechanism to re-configure a pre-trained vision network that does not change based on the task at hand. Typically, representations that are supervised and depend on the current task are less flexible to changes in the task and are slow to adapt to changes in the environment or goal. The architecture of Luo et al. (2020) only needs to learn a new set of attention filters for a new task, which allows it to quickly re-purpose the representations learnt by the vision network and can create sub-networks for different tasks. In comparison, the architecture of Mott et al. (2019) will alter the parameters across all of its networks in response to a new task. This is likely to greatly reduce the flexibility of Mott et al. (2019)’s approach and misses out on one of the greatest benefits of selective attention; fast and dynamic changes in state representation based on the current goal. In both Luo et al. (2020) and Mott et al. (2019) the ability of each of the agent architectures to deal with task changes is not explored qualitatively or quantitatively.

5.5 Conclusions

In this chapter we have seen that the Pre-Frontal Cortex (PFC) is able to support efficient Reinforcement Learning (RL) by interacting with the striatum, hippocampus and sensory cortices. These interactions highlight the need for Complementary Learning Systems (CLS) theory to account for the distinction between the PFC and other cortical regions. The PFC has long been associated with executive control and this is apparent in each of the pathways that we have described here. In particular, we have highlighted the ability of the PFC to perform meta-learning, contextual memory retrieval and selective attention as important for performing rapid learning and transfer in RL. Imbuing Deep RL systems with these abilities is likely to dramatically improve their efficiency.

Wang et al. (2016, 2018) have proposed that the PFC can act directly with the striatum to support efficient RL through meta-learning or ‘learning to learn’. This proposal suggests that the high-degree of recurrency within the PFC allows it to implement an RL algorithm within its activity dynamics, while the updating of synapses are used to improve this RL algorithm. In this way the PFC is able to use dopamine reward-prediction errors to learn how to improve its learning on a series of related tasks. The only conditions required for such an ability is the presence of a recurrent network with memory and access to reward prediction errors, which are both satisfied by the PFC-striatal pathway. No task specific engineering is required and so this form of meta-learning can be considered a domain general mechanism. Due to these simple requirements Wang et al. (2016) have demonstrated that this form of meta-learning can occur naturally in Deep RL algorithms and that it can perform interesting forms of rapid learning and transfer, such as solving the Harlow (1949) task via ‘one-shot’ learning.

Aside for direct interactions with the striatum, the PFC can also interact with the hippocampus to support efficient RL. Often mechanisms of transfer and analogy do not consider the problem of selecting memories to transfer knowledge from and instead focus on the transfer or alignment process itself. A wealth of empirical evidence has started to suggest that the selection of episodic memories based on the current context might be due to interactions between the PFC and the hippocampus (Place et al., 2016; Eichenbaum, 2017; Dobbins et al., 2002; Navawongse and Eichen-

baum, 2013). In addition to contextual memory retrieval, interactions between the PFC and hippocampus have also been suggested to give rise to conceptual representations in the hippocampus (Preston and Eichenbaum, 2013; Schlichting and Preston, 2016; Mack et al., 2016, 2020). Concepts are likely to be a critical component of efficient RL as they allow for reasoning about exemplars that have never been seen before. It appears that the PFC provides an attentional signal to the hippocampus in order to highlight the features that are diagnostic of a particular concept. Evidence for this comes from the fact that a computational model of concept learning called SUSTAIN, which utilises attentional signals, is able to predict the neuronal responses of the hippocampus and PFC (Mack et al., 2016).

The role of the PFC in selective attention is not only limited to interactions with the hippocampus. The PFC is also thought to exert selective attention on sensory cortices to guide perception towards features of the environment that are important for the task at hand (Radulescu et al., 2019). From a transfer perspective, selective attention on sensory cortices can be seen as selecting features from past experience that are useful for the current goal. In this way, selective attention can influence RL during both choice and learning. At choice, selective attention can bias state and/or action evaluations by increasing the influence of certain features, while during learning it can bias synaptic updates towards certain features (Leong et al., 2017). This can greatly improve the efficiency of RL by reducing the dimensionality of the state representation and streamlining learning towards the most relevant features. Current models of selective attention in Deep RL have demonstrated these benefits (Mott et al., 2019) but have missed out on another major benefit; increased flexibility. One of the key properties of selective attention is that it can be changed quickly in the face of changing task demands thereby greatly increasing the flexibility of RL. Deep RL models typically rely on backpropagation to learn attentional weights and are therefore slow to adapt to changes in the task. This suggests that these approaches are missing key computational properties of selective attention and the interactions between the PFC and sensory cortices.

With this in mind, in the next chapter we propose a novel algorithm for imbuing a Deep RL agent with the ability to perform selective visual attention. The goal of this algorithm is to provide a computational account of how the brain is able to

flexibly learn which features to attend to based on the task at hand. Rather than using slow incremental learning, such as backpropagation, our approach relies on Bayesian approximation in the form of a particle filter. The particle filter represents different hypotheses about which latent visual features are useful for the current task, which is thought to be the responsibility of the PFC (Miller and Cohen, 2001; Radulescu et al., 2019). The particle filter is updated based on bottom-up and top-down influences, which are both important components of human attention (Desimone and Duncan, 1995; Treue, 2003). Critically, we demonstrate that our approach improves the efficiency of RL both in terms of rapid learning and transfer. This further serves to highlight the importance of extending CLS theory to include the distinction between the PFC and sensory cortices in order to understand efficient RL.

Chapter 6

Selective Particle Attention

Overview

In this chapter we present a novel algorithm called Selective Particle Attention (SPA) that uses visual feature-based attention to improve the efficiency of Reinforcement Learning (RL). SPA mimics interactions between the Pre-Frontal Cortex (PFC) and sensory cortices by identifying learnt features of the visual scene that are important for the current task, regardless of their spatial location. Such a mechanism has been proposed to improve the efficiency of Reinforcement Learning (RL) in the brain by reducing the dimensionality of state representations and guiding learning towards relevant features. From a mechanistic point of view, SPA uses a particle filter to model how the PFC represents multiple hypotheses about which features to attend to based on the current task (Section 6.2). SPA filters these hypotheses based on how accurately each one predicts future reward and the bottom-up saliency of individual features. We evaluate SPA on a multiple choice task and a 2D video game that both involve raw pixel input and dynamic changes to the task structure (Section 6.3). We show various benefits of SPA over approaches that naively attend to either all or random subsets of features. Our results demonstrate (1) how visual feature-based attention in Deep RL models can improve their learning efficiency and ability to deal with sudden changes in task structure and (2) that particle filters may represent a viable computational account of how visual feature-based attention occurs in the brain. ¹

¹The work in this chapter is under review and a pre-print can be found at: Blakeman, S., & Mareschal, D. (2020). Selective Particle Attention: Visual Feature-Based Attention in Deep

6.1 Introduction

In the previous chapter we made a case for extending Complementary Learning Systems (CLS) theory to include the differentiation between sensory cortices and the PFC. From making this distinction we saw that the interaction between the PFC and sensory cortices (Pathway 6 in Figure 5.1) is responsible for selective attention (Desimone and Duncan, 1995; Miller and Cohen, 2001), which refers to the identification of features of the environment that are relevant for the task at hand. Selective attention can occur across several modalities including vision and audition. In vision, selective attention is often broadly split into two types; spatial and feature-based attention (Lindsay, 2020). Spatial attention refers to the selective processing of specific areas of the visual field. In comparison, feature-based attention is used to selectively process specific features of the visual input regardless of their location in the visual field and is the focus of this chapter. Feature-based attention is typically studied by priming individuals to attend to a certain feature and then measuring their ability to detect the primed feature. For example, participants who are primed to attend to a specific orientation of visual grating are subsequently better at detecting that grating (Rossi and Paradiso, 1995). From a neural point of view, the firing of neurons that encode the attended feature appear to increase, while the firing rates of neurons encoding the non-attended feature appear to decrease (Treue and Trujillo, 1999; Saenz et al., 2002). This modulation of firing rates in the visual stream is thought to originate from the PFC (Bichot et al., 2015; Paneri and Gregoriou, 2017) and be most effective in higher order visual areas (Lindsay and Miller, 2018; Baluch and Itti, 2011). Importantly, these top-down goal-driven influences from the PFC are thought to work in combination with bottom-up influences (Desimone and Duncan, 1995), which are driven by intrinsic properties of the stimuli that are task-agnostic (Wolfe and Horowitz, 2004). While these priming studies demonstrate the effect of selective attention they do not tackle the problem of how to learn which features to attend to.

The work in this chapter focuses on a computational account of how visual feature-based attention can interface with Reinforcement Learning (RL) to help us learn which features to attend to in a task with a specified reward structure. Selective

Reinforcement Learning. *arXiv preprint arXiv:2008.11491.*

feature-based attention can greatly reduce the complexity of the RL problem by reducing the dimensionality of the state representation to only the features that are important for the current task (Jones and Canas, 2010; Niv et al., 2015; Wilson and Niv, 2012). It is typically thought to be much faster acting than the incremental learning of new representations as it allows for the quick and flexible re-purposing of existing representations. Selective attention in RL can affect both learning and choice. During learning, selective attention can modify the magnitude of weight updates for each feature and during choice it can alter the magnitude of each features contribution to a decision (Leong et al., 2017). There is therefore a reciprocal relationship between attention and learning in RL; attention biases learning but learning guides which features are attended to. It remains an open question how we learn which features to attend to based on the reward structure of the current task. One proposal is that we learn to attend to the features that are most predictive of reward (Mackintosh, 1975). For example, people are able to switch between an object-based or a feature-based state representation based on which one is the best predictor of reward (Farashahi et al., 2017).

Advances in Deep RL have provided the opportunity to begin examining how the brain may go from naturalistic high-dimensional input to action based on reward signals. Deep RL has been particularly powerful in the visual domain where Deep RL agents have learnt to perform complex tasks from raw pixel inputs, such as playing video games at human level performance (Mnih et al., 2015, 2016). These models typically rely on the use of Deep Convolutional Neural Networks (DCNNs) to approximate a value function and/or a policy. This is of interest to cognitive scientists and neuroscientists because a growing body of evidence is suggesting that the hierarchical representations learnt by DCNNs are similar to those found in the ventral stream of the human brain (Güçlü and van Gerven, 2015; Yamins and DiCarlo, 2016; Schrimpf et al., 2018). Despite these successes, one consistent criticism of Deep RL models is that they lack the efficiency and flexibility demonstrated by human learning. Selective attention represents one potential mechanism for helping to address these issues. We saw in Chapter 5 that initial attempts to imbue agents with selective attention have involved a mechanism known as self-attention, which maps query and key-pair vectors to an output vector (Mott et al., 2019; Manchin

et al., 2019; Zambaldi et al., 2018; Bramlage and Cortese, 2020). This computation is fully differentiable and so can be trained end-to-end using backpropagation.

These forms of selective attention in Deep RL, and indeed other forms (Sorokin et al., 2015), have several striking differences to feature-based attention in the human visual cortex. Firstly, when presented with a novel situation humans typically select from pre-existing features and use a function of these features to guide decision-making, rather than learning a completely new set of features. This is important because it eludes to one of the most powerful properties of selective feature-based attention; it can quickly and dynamically re-purpose representations without overwriting them. In comparison, learning new features in the visual stream is thought to occur slowly over many experiences and is often referred to as perceptual learning (Schyns et al., 1998). Being fully differentiable, the aforementioned approaches rely on a selective attention mechanism that ultimately learns the underlying representations that are to be attended to, which is both slow and inflexible. A second noticeable difference between current visual selective attention mechanisms in Deep RL and those in the brain is the lack of bottom-up influences. Bottom-up influences are a critical component of human visual attention and help to guide us to salient pieces of information in the environment (Treue, 2003). This suggests that their inclusion in Deep RL models may help to improve their performance and bring them closer to models of human visual selective attention.

With these criticisms in mind, we propose a novel algorithm that we term *Selective Particle Attention (SPA)*, which implements visual feature-based attention in a Deep RL agent (Figure 6.1A) (Blakeman and Mareschal, 2020b). *SPA* consists of three main steps; feature extraction, feature selection and value computation. At its core *SPA* uses Bayesian principles to quickly and flexibly re-purposes features that are learnt slowly over many examples. This speeds up learning by reducing the dimensionality of the problem and reduces interference by preserving the underlying features. Each step of *SPA* resembles the problem faced by the human brain in several important ways. Firstly, a pre-trained DCNN is used to extract features of the visual input. We use VGG-16 due to its relatively low computational costs and good correspondence with representations found in the visual stream (Schrimpf et al., 2018). Importantly, no further training of VGG-16 is performed in order to

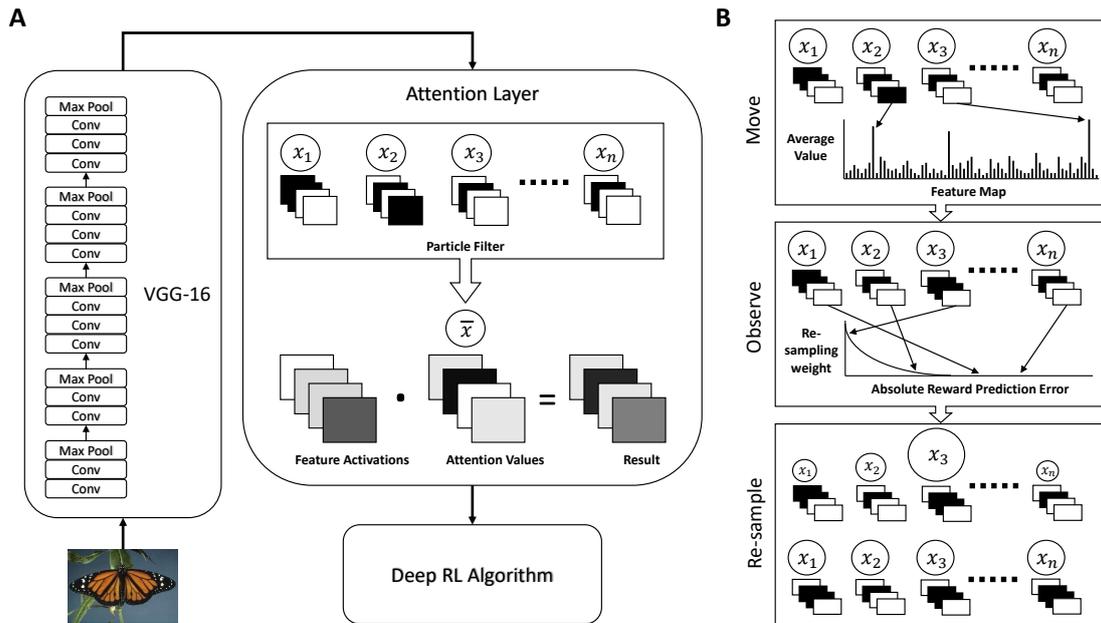


Figure 6.1: *Depiction of Selective Particle Attention (SPA). (A) Overall architecture of SPA. A pre-trained deep convolutional neural network (VGG-16) is used to extract 2D feature maps from an image. The feature maps are then fed to an attention layer, which uses a particle filter to generate attention values. Each particle (x_i) in the particle filter uses a binary vector to represent a hypothesis about which feature maps are useful for the current task. The attention values are calculated as the mean over all particle states, normalised to sum to one (\bar{x}). Each of the feature maps is multiplied by its corresponding attention value in an element-wise fashion to produce the output of the attention layer. The result is passed to a Deep Reinforcement Learning (RL) algorithm for action selection. (B) Updating of the particle filter. In the first (movement) step, a small proportion of particle states (x_i) are updated to reflect the most active features given the current input. In the second (observe) step, particles are assigned re-sampling weights based on how accurately they predict reward. In the final (re-sample) step, the re-sampling weights are normalised to produce a probability distribution and the particles are re-sampled with replacement.*

mimic how the human brain has to pick from existing representations that change very slowly in sensory cortices.

Once the features have been extracted from VGG-16, we implement selective feature-based attention by filtering the features using a set of attention values generated by a particle filter (Figure 6.1B). Particle filters have been proposed to represent a computationally plausible model of selective attention in the brain (Radulescu et al., 2019) and have been shown to better model shifts in attention than gradual error-driven learning (Radulescu et al.). In our approach, each particle represents a hypothesis about which features are important for the current task. This is analo-

gous to the PFC, which represents competing hypotheses about which features of the visual input to attend to given the current goal (Miller and Cohen, 2001; Radulescu et al., 2019). Each particle is updated based on how accurately it predicts future reward and the bottom-up salience of individual features. *SPA* therefore combines both bottom-up and top-down attention to help guide feature selection. The attention values are calculated by normalising the mean particle state to sum to one. Interestingly, computational models of visual attention highlight normalisation as a key step for capturing competition between features in the visual cortex (Reynolds and Heeger, 2009).

After the particle filter has been used to selectively attend to specific visual features, we apply a Deep RL algorithm to approximate the value function and/or policy. This can be tentatively compared to the role of the striatum, which is thought to evaluate states and/or actions based on inputs from cortical regions (Schultz, 1998; Houk et al., 1995; Joel et al., 2002; Maia, 2009; Setlow et al., 2003) and temporal difference errors (Schultz et al., 1997). Our approach is compatible with a wide range of Deep RL algorithms, the only requirement is that the Deep RL algorithm utilises a value function. This is so that the particle filter can use the value predictions to assess how accurately each particle predicts future reward.

We assess our approach, on two key tasks; a multiple choice task and a 2D video game based on collecting objects. Both tasks involve processing observations from raw pixel input and dealing with unannounced changes in task structure. In both cases the selective attention mechanism of *SPA* leads to improved performance in terms of the onset, speed, robustness and flexibility of learning compared to naive approaches that either attended to all or a random subset of features. We also show that these findings occur independently of the RL algorithm used, making it applicable to a variety of problems. Overall our results demonstrate that *SPA* is a viable method for performing visual feature-based attention in a Deep RL agent and that it may capture some of the key computational properties of selective attention in the brain. In particular, *SPA* provides a mechanistic explanation of how bottom-up and top-down attention may interact in the brain in order to guide feature selection based on the task at hand.

6.2 Methods

6.2.1 Tasks

We used two different tasks to assess our approach: a multiple choice task and a 2D video game called the object collection game. Both tasks require the agent to select actions based on raw pixel input and to switch attention in response to changing task demands. The multiple choice task involves making a single decision on each trial, which is equivalent to a single step Markov Decision Process (MDP). In comparison, the object collection game involves sequential decisions and therefore corresponds to a multi-step MDP. The subsequent sections will discuss each task in more detail.

6.2.1.1 Multiple Choice Task

For the multiple choice task we used the Caltech 101 data set (Fei-Fei et al., 2006), which consists of 101 object categories with approximately 40 to 800 images per category. Three categories from the Caltech 101 data set were chosen at random and the images from those categories were used for the multiple choice task (Figure 6.2). The task consisted of 200 blocks of 50 trials. A single trial consisted of presenting the agent with 3 different natural images, one from each of the chosen categories. The images were presented separately, one after the other. For each block one of the categories was chosen at random and associated with a positive reward of +1 while the others were associated with no reward. The agent therefore has to work out which image is associated with a reward on each trial based on features of a specific category. Every time a new block starts the agent also has to adapt to the change in reward structure using only reward feedback.

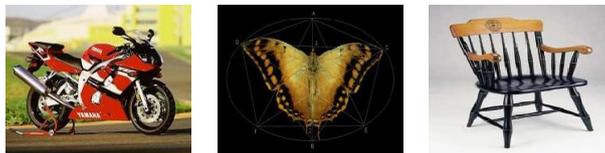


Figure 6.2: *Example images from the Caltech 101 data set. Left image is from the ‘motorbikes’ category, the middle image is from the ‘butterfly’ category and the right image is from the ‘chair’ category. On each trial the agent was presented with three separate images each selected randomly from three different categories. For a given block, only one image category was associated with a reward of +1 and the rest were associated with a reward of 0. The rewarded category was chosen at random for each block of trials.*

To ensure that the agent did not learn to remember specific exemplars in the data set we had training and test phases. During the training phase the agent was able to update the parameters of the Deep RL network in response to reward feedback, however during the test phase it was not. With respect to attention, the agent was allowed to update its attention values during test but the network weights were kept fixed. The test phase used images from the chosen object categories that were not presented during training and consisted of 10 blocks of 50 trials.

6.2.1.2 Object Collection Game

The 2D video game, which we refer to as the object collection game, was made using Pygame (www.pygame.org). In the game, the agent controlled a grey block at the bottom of the screen and could move it either left or right at each time step. Every second an object that could vary in shape and colour was generated at the top of the screen and moved downwards to the bottom of the screen. The goal of the agent was to collect objects by colliding with them as they reached the bottom of the screen. Agents were trained for 2000 episodes, with each episode lasting 60 seconds.

We explored two different variations of this basic game configuration. In the first variation, the agent was given a reward of +1 for collecting objects of a certain shape and a reward of 0 for all other shapes (Figure 6.3A). This tested the agent’s ability to focus on particular features of the environment and ignore others; the agent had to attend to the target shape while ignoring colour and all other shapes. As we saw in the multiple choice task, one benefit of an effective selective attention mechanism is that attention can be altered in response to changes in the reward structure. To test whether this was also the case in the object collection game, we randomly selected a new shape to be rewarded after 1000 episodes. This is akin to changing the reward function of the environment and requires the agent to re-evaluate its policy. We refer to this variation as the reward revaluation task.

In addition to the reward function, other aspects of the environment can change such as the set of perceptual observations experienced by the agent. In RL this is commonly referred to as a change in state space. The second variation of the object collection game therefore explored the ability of *SPA* to deal with changes in

state space (Figure 6.3B). For the first 1000 episodes, all the objects were the same shape and colour and were associated with a reward of +1. After 1000 episodes the shape and colour of the objects were changed to a different shape and colour. This corresponds to a change in the perceptual input of the agent because the objects present in the first half of training are different to the ones present in the second half of training. We refer to this variation as the state revaluation task.

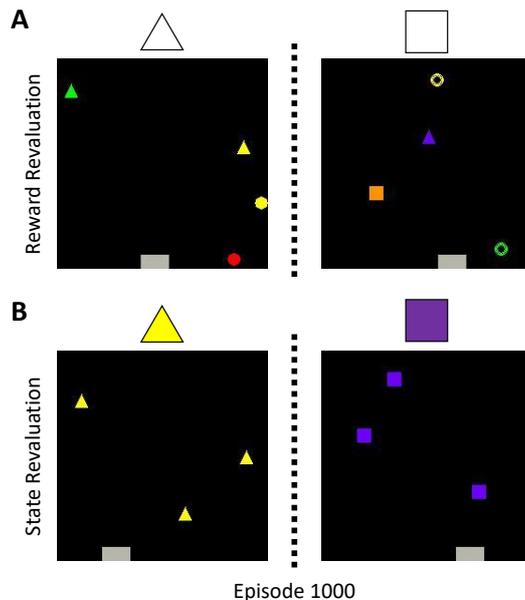


Figure 6.3: *Example screenshots from the two variations of the object collection game. The agent is in control of the grey rectangle and can move it either left or right to catch objects that move from the top of the screen to the bottom. The rewarded object is depicted above each screenshot. (A) The reward revaluation task. A specific shape is associated with a reward of +1 regardless of its colour. All other shapes are associated with a reward of 0. The rewarded shape is then randomly changed after 1000 episodes, which corresponds to a change in the reward function of the environment. (B) The state revaluation task. All objects are the same shape and colour, and are associated with a reward of +1. The shape and colour of these objects is then randomly changed after 1000 episodes, which corresponds to a change in the state space of the environment.*

6.2.2 Selective Particle Attention

Our approach, termed Selective Particle Attention (SPA), consists of three main components; a pre-trained Deep Convolutional Neural Network (DCNN) for feature extraction, a particle filter for selective attention and a Deep RL algorithm for action selection. The subsequent sections break-down each component and their underlying mechanisms in more detail.

6.2.2.1 VGG-16

To perform feature extraction on raw pixel values we use a pre-trained Deep Convolutional Neural Network (DCNN). We chose VGG-16 (Simonyan and Zisserman, 2014) as the DCNN for all simulations because of its relative simplicity and good correspondence with representations found in the visual stream (Schrimpf et al., 2018). The architecture of VGG-16 consists of 5 blocks of convolutional layers each with a max pooling layer at the end. VGG-16 was trained on a classification task using the Imagenet dataset (Deng et al., 2009), which consists of naturalistic images. For our simulations we performed no further training of VGG-16 and removed the fully connected layers after the final convolutional block. This left us with 512 feature maps as output, which served as the basis for selective attention. For all simulations the input images were resized to be 224 X 224 pixels and converted to RGB format in order to match the required inputs of VGG-16. Pre-processing was performed in the standard manner for VGG-16 (Simonyan and Zisserman, 2014).

6.2.2.2 Attention Layer

The attention layer is applied to the final feature maps provided by VGG-16 using a process similar to the one described by Luo et al. (2020). Attention is represented as a K dimensional vector that can take on any real valued number between 0 and 1 inclusive:

$$\mathbf{A} \in [0, 1]^K \tag{6.1}$$

Where K is the number of feature maps, which in our case was 512. This attention vector is applied to the final feature maps of VGG-16 using the hadamard product between \mathbf{A} and the values of each feature map. This requires that each entry in \mathbf{A} is replicated to match the dimensions of a single feature map, with the same attentional value applied across all spatial locations. This process re-weights the feature map activations, amplifying feature maps with a large attentional weight. The output of the attention layer is then reshaped into a single vector and passed on to a Deep Reinforcement Learning (RL) algorithm for action selection. Our approach is independent of the Deep RL algorithm used, so long as it involves the

approximation of a state value function (see below).

6.2.2.3 Particle Filter

The particle filter is responsible for learning the values of the attention vector given the current task. Particle filters are typically used to estimate the value of a latent variable (X) given noisy samples of an observed variable (O) when the number of potential values is large. The overall goal of a particle filter is to use a set of ‘particles’ to represent a posterior distribution over the latent variable. Each particle represents a belief or hypothesis about the value of the latent variable and the density of the particles can be used to approximate the posterior distribution.

In our case the latent variable X represents the configuration of features that are useful for the current task. We use a value of 1 to denote a feature as useful and a value of 0 to denote a feature as not useful. This means that x can be any binary vector of size K :

$$x \in \{0, 1\}^K \tag{6.2}$$

Where K is the number of features, which in our case was 512 corresponding to the number of feature maps of VGG-16. As the number of possible values that X can take is 2^K , we use N particles to approximate the posterior distribution over X where $N \ll 2^K$. The state x^i of the i^{th} particle therefore corresponds to a binary vector of length K and provides a hypothesis about which features it deems relevant for the current task.

A particle filter consists of two main steps; a movement step and an observation step. In the movement step, the particles are updated based on some known transition probability for the latent variable:

$$x' \sim P(X'|x) \tag{6.3}$$

This process is often used to represent the passing of time. In our approach we introduce the notion of bottom-up attention during the movement step. Let \bar{f}_t^k denote the average value over all the units in feature map k from VGG-16 at time t . At each time-step t a particle is updated as follows with some probability ϕ :

$$v_t^k = \frac{\bar{f}_t^k}{\sum_{j=1}^K \bar{f}_t^j} \quad (6.4)$$

$$p^k = \frac{\exp(v_t^k * \tau_{\text{BU}})}{\max_j \exp(v_t^j * \tau_{\text{BU}})} \quad (6.5)$$

$$P(x'_k = n) = \begin{cases} p^k & \text{for } n = 1 \\ 1 - p^k & \text{for } n = 0 \end{cases} \quad (6.6)$$

First the mean activation values for each feature map of VGG-16 are normalised to sum to 1. This normalisation accounts for differences in overall activation values between time steps and preserves relative differences between activation values. The activation values are then exponentiated and normalised by the maximum value across all feature maps. This ensures that the most active feature will receive a value of 1. Finally this is used as the probability that the k^{th} entry of the particle state will be equal to one, as described by a Bernoulli distribution. In this way, a proportion of the particles are updated to represent the most active features given the current input. This is akin to bottom-up attention, whereby highly salient perceptual features capture ones attention in an involuntary manner. In our agent this serves to introduce a prior to attend to the highly active features of the current task. The free parameter τ_{BU} controls the strength of the bottom up attention, a higher value of τ_{BU} leads to a higher probability of the most active features being attended to and the least active features not being attended to.

In the observation step of a particle filter, particles are weighted based on the likelihood of the observed variable O given the value of the latent variable X represented by a particle. These weights are then used to re-sample the particles and update the posterior distribution ready for the next time step. In our approach, we introduce top-down attention during this step. We take our observed variable O to be the return from a given state R_t . We calculate the likelihood of this return by using the particle state as the attention vector and calculating the error between the predicted state value and the return. Let x^i denote the state of the i^{th} particle, the likelihood for x^i is calculated as follows:

$$A_k^i = \frac{x_k^i}{\sum_{j=1}^K x_j^i} \quad (6.7)$$

$$\delta^i = (R_t - V(s_t; \mathbf{A}^i))^2 \quad (6.8)$$

$$P(R_t|x^i) \propto \exp(-(\delta^i - \min_j \delta^j) * \tau_{TD}) \quad (6.9)$$

Where R_t is the return from state s_t and $V(s_t; \mathbf{A}^i)$ is the predicted state value calculated by using the normalised particle state x^i as the attention vector. The normalisation step in Equation 6.7 accounts for the different numbers of features that are attended to by different particles. Equation 6.8 calculates the squared error between the return and predicted state value. The likelihood of the return R_t is then proportional to this error value. This process can be seen as evaluating the accuracy of a particle’s hypothesis over which features are relevant for the given task. If the particle’s hypothesis is good then it will more accurately predict the target return and so will produce a larger likelihood. In this way we capture the effect of top-down attention, whereby a set of hypotheses are evaluated and the most accurate ones are considered for the next time step. τ_{TD} controls the strength of this top down attention; a larger value will more strongly penalise hypotheses that are inaccurate.

Once the likelihoods have been calculated they are normalised to form a probability distribution and the particles are re-sampled with replacement:

$$P(x') = \frac{\sum_{i=1}^N P(R_t|x^i)\mathbb{I}(x^i = x')}{\sum_{i=1}^N P(R_t|x^i)} \quad (6.10)$$

$$x' \sim P(x') \quad (6.11)$$

Once the re-sampling has been performed, the final step is to reset the value of the attention vector. This is done by setting the attention vector to be the mean of the particle states and then normalising the vector to sum to one:

$$\bar{x}_k = \frac{1}{N} \sum_{n=1}^N x_k^n \quad (6.12)$$

$$A_k = \frac{\bar{x}_k}{\sum_{j=1}^K \bar{x}_j} \quad (6.13)$$

Where N is the number of particles and K is the number of features. The full algorithm used to update the attention vector can be seen in Algorithm 6.

Algorithm 6 UpdateAttention($\mathbf{v}, \mathbf{s}, \mathbf{R}, \phi$)

Receive vector of normalised mean feature map values $\mathbf{v} = \{v_t, \dots, v_{t+Z}\}$

Receive vector of states $\mathbf{s} = \{s_t, \dots, s_{t+Z}\}$

Receive vector of returns $\mathbf{R} = \{R_t, \dots, R_{t+Z}\}$

Receive movement probability ϕ

1. Perform Movement Step

Calculate feature probabilities

$$p^k = \frac{\frac{1}{Z} \sum_{z=t}^{t+Z} \exp(v_z^k * \tau_{BU})}{\max_j \frac{1}{Z} \sum_{z=t}^{t+Z} \exp(v_z^j * \tau_{BU})}$$

Update each particle state x^i with probability ϕ

$$P(x_k^i = n) = \begin{cases} p^k & \text{for } n = 1 \\ 1 - p^k & \text{for } n = 0 \end{cases}$$

2. Perform Observe Step

Calculate the likelihoods of each particle x^i

$$A_k^i = \frac{x_k^i}{\sum_{j=1}^K x_j^i}$$

$$\delta^i = \frac{1}{Z} \sum_{z=t}^{t+Z} (R_z - V(s_z; \mathbf{A}^i))^2$$

$$P(\mathbf{R}|x^i) \propto \exp(-(\delta^i - \min_j \delta^j) * \tau_{TD})$$

Re-sample particles based on calculated likelihoods

$$P(x') = \frac{\sum_{i=1}^N P(\mathbf{R}|x^i) \mathbb{I}(x^i = x')}{\sum_{i=1}^N P(\mathbf{R}|x^i)}$$

$$x^i \sim P(x')$$

3. Update Attention

Calculate mean particle state and normalise

$$\bar{x}_k = \frac{1}{N} \sum_{n=1}^N x_k^n$$

$$A_k = \frac{\bar{x}_k}{\sum_{j=1}^K \bar{x}_j}$$

6.2.2.4 Deep Reinforcement Learning Algorithm

The final component of our approach is a standard Deep Reinforcement Learning (RL) algorithm for selecting actions based on the results of the selective attention mechanism. Our approach can be used with a variety of Deep RL algorithms as long as they involve a value function. The use of a value function is critical because the particle filter uses the value predictions of each of its particles to calculate likelihood values for re-sampling (Equation 6.9). A value function therefore allows the particle filter to assess different hypotheses based on how predictive they are of reward. The specifics of each of the Deep RL algorithms used are covered in the following sections based on the task being considered.

Multiple Choice Task

For the multiple choice task we passed the results of the attention layer to a three layer feed forward Deep Neural Network (DNN) (Figure 6.4). The DNN was used to approximate the value of a given state, which in this case was the value of a given image. To select an action the value of each image was calculated and one of the images was chosen in an ϵ -greedy manner, with $\epsilon = .2$. As each trial was based on a single choice, the problem is equivalent to a single step Markov Decision Process (MDP). We therefore trained the DNN after each trial to minimise the squared error between the reward experienced on a trial (r_t) and the predicted value of the chosen image (s_t):

$$J(\theta) = \frac{1}{2}(R_t - V(s_t; \theta, \mathbf{A}))^2 \quad (6.14)$$

$$= \frac{1}{2}(r_t - V(s_t; \theta, \mathbf{A}))^2 \quad (6.15)$$

$$\nabla_{\theta} J(\theta) = -(r_t - V(s_t; \theta, \mathbf{A})) \nabla_{\theta} V(s_t; \theta, \mathbf{A}) \quad (6.16)$$

Where θ are the parameters of the DNN, \mathbf{A} is the attention vector, s_t is the image chosen at time t and r_t is the reward associated with the image chosen. We used RMSProp as an optimizer and the hyper-parameter values can be seen in Table 6.1. As mentioned in Section 6.2 the weights of VGG-16 were kept constant and a particle filter was used to dynamically update the features that are being attended

to. For the particular filter the return (R_t) in Equation 6.8 was simply the reward experienced at the end of each trial (r_t). The full algorithm for the multiple choice task is shown in Algorithm 7.

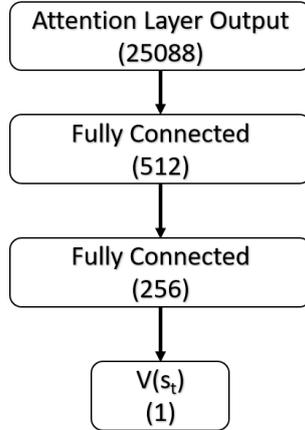


Figure 6.4: Architecture used in the multiple choice task for the deep reinforcement learning component of Selective Particle Attention (SPA). Numbers in brackets represent the number of units and each layer was fully connected. $V(s_t)$ corresponds to the value of a particular image.

Table 6.1: Hyper-parameter values used for the multiple choice task.

Parameter	Value	Description
N	250	Number of particles
τ_{BU}	10	Strength of bottom-up attention
τ_{TD}	10	Strength of top-down attention
ϵ	.2	Probability of selecting a random action
λ	.00025	Learning rate for RMSProp
κ	.95	Momentum for RMSProp
ι	.01	Constant for denominator in RMSProp

Algorithm 7 Full algorithm for the multiple choice task.

Initialize value function parameters with random weights θ
Initialize attention vector $\mathbf{A} \leftarrow \mathbf{1}^{\mathbf{K}} \cdot \frac{1}{K}$
Initialize state of each particle $x^i \leftarrow \mathbf{0}^K$
Initialise movement probability $\phi \leftarrow 1$
for $t = 1, T$ **do**
 Observe the three images $\{s_t^1, s_t^2, s_t^3\}$
 With probability ϵ select a random action a_t
 Otherwise $a_t \leftarrow \arg \max_a (V(s_t^a); \theta, \mathbf{A})$
 Receive reward r_t
 Get mean feature maps \bar{f}_t from random image $\sim U\{s_t^1, s_t^2, s_t^3\}$
 Normalise feature maps $v_t^k \leftarrow \bar{f}_t^k / \sum_{j=1}^K \bar{f}_t^j$
 $UpdateAttention(\{v_t\}, \{s_t^{a_t}\}, \{r_t\}, \phi)$
 $\theta \leftarrow \theta + \alpha(r_t - V(s_t^{a_t}; \theta, \mathbf{A})) \nabla_{\theta} V(s_t^{a_t}; \theta, \mathbf{A})$
 $\phi \leftarrow .01$
end for

Object Collection Game

For the object collection game we passed the results of the attention layer to a fully connected layer followed by a Long Short-Term Memory (LSTM) network, which output the state value and a softmax distribution over actions (Figure 6.5). The network therefore resembled an advantage actor critic (A2C) architecture (Mnih et al., 2016), whereby the predicted state value from the critic was used to calculate the advantage function. We chose this setup to explore whether *SPA* could work when the network also had to compute a policy. The agent received every 8th frame as input. The cost function of the critic was the mean squared error between the return and the predicted value:

$$J(\theta) = \frac{1}{2}(R_t - V(s_t; \theta, \mathbf{A}))^2 \quad (6.17)$$

$$\nabla_{\theta} J(\theta) = -(R_t - V(s_t; \theta, \mathbf{A})) \nabla_{\theta} V(s_t; \theta, \mathbf{A}) \quad (6.18)$$

Where θ are the parameters of the DNN, \mathbf{A} is the attention vector, s_t is the frame of the game at time t and R_t is the return from state s_t . The actor was updated using the advantage calculated from the critic:

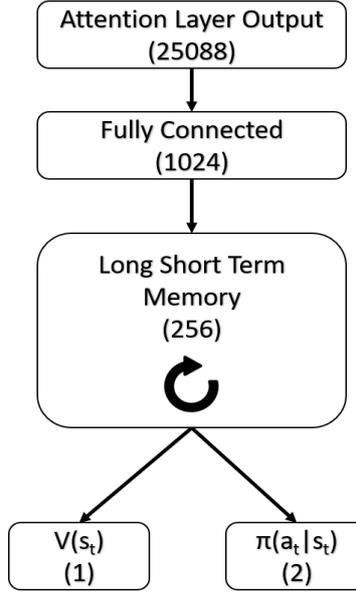


Figure 6.5: Architecture used in the object collection game for the deep reinforcement learning component of Selective Particle Attention (SPA). Numbers in brackets represent the number of units and each layer was fully connected. An actor-critic architecture was used so that the network output both a state value and a probabilistic policy in the form of a soft-max distribution.

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t) \quad (6.19)$$

$$= R_t - V(s_t; \theta, \mathbf{A}) \quad (6.20)$$

$$\nabla_{\theta} L(\theta) = A(s_t, a_t) \nabla_{\theta} \ln \pi(a_t | s_t; \theta, \mathbf{A}) + \beta \nabla_{\theta} H(\pi(a_t | s_t; \theta, \mathbf{A})) \quad (6.21)$$

Where H represents the entropy of the softmax distribution over actions and is included to encourage exploration. β is a free parameter that controls the strength of this entropy term and was set to 0.01 for all simulations. In all cases, the return R_t from state s_t was estimated using n-step Temporal Difference (TD) learning:

$$R_t^n = r_t + \gamma r_{t+1} + \dots \gamma^{n-1} r_{t+n-1} + \gamma^n V(s_{t+n}; \theta, \mathbf{A}) \quad (6.22)$$

$$= \sum_{i=0}^{n-1} \gamma^i r_{t+i} + \gamma^n V(s_{t+n}; \theta, \mathbf{A}) \quad (6.23)$$

This was used for calculating the target for the critic via Equation 6.17 and for calculating the advantage in Equation 6.20. The training procedure was the same

as in Mnih et al. (2016) with t_{max} set to 10. More specifically, the agent selects actions up until t_{max} or the trial ends. At this point the the gradients are calculated for each of the n -step TD-learning updates using the longest possible n -step return. For example, the update for the last state will be a one-step update whereas the update for the first state will be a t_{max} -step update. These gradients are all applied in a single update.

The returns calculated using the aforementioned procedure were also used in Equation 6.8 for calculating the observation step of the particle filter. However, the movement and update steps of the particle filter were not performed on every training step (when t_{max} or the trial ends). Instead they were performed every c_{max} training steps, we found that this improved both stability and reduced training time. The full algorithm is shown in Algorithm 8 and Table 6.2 shows the hyper-parameter values used.

Algorithm 8 Full algorithm for the object collection game.

Initialize A2C network parameters with random weights θ
Initialize time step and attention step counters $t \leftarrow 0, c \leftarrow 0$
Initialize attention vector $\mathbf{A} \leftarrow \mathbf{1}^K \cdot \frac{1}{K}$
Initialize state of each particle $x^i \leftarrow \mathbf{0}^K$
Initialise movement probability $\phi \leftarrow 1$
for $e = 1, E$ **do**
 $d\theta_\pi, d\theta_v \leftarrow 0$
 $t_{start} \leftarrow t$
 Get state s_t and mean feature maps \bar{f}_t
 repeat
 Normalise mean feature maps $v_t^k \leftarrow \bar{f}_t^k / \sum_{j=1}^K \bar{f}_t^j$
 Perform action a_t using policy $\pi(a_t | s_t; \theta_\pi, \mathbf{A})$
 Receive reward r_t , state s_{t+1} and mean feature maps \bar{f}_{t+1}
 $t \leftarrow t + 1$
 until terminal s_t **or** $t - t_{start} == t_{max}$
 $R_t = \begin{cases} 0 & \text{if terminal } s_t \\ V(s_t; \theta_v, \mathbf{A}) & \text{otherwise} \end{cases}$
 for $i \in \{t - 1, \dots, t_{start}\}$ **do**
 $R_i \leftarrow r_i + \gamma R_{i+1}$
 end for
 $c \leftarrow c + 1$
 if $c == c_{max}$ **then**
 $\mathbf{v} \leftarrow \{v_{t_{start}}, \dots, v_{t-1}\}$
 $\mathbf{s} \leftarrow \{s_{t_{start}}, \dots, s_{t-1}\}$
 $\mathbf{R} \leftarrow \{R_{t_{start}}, \dots, R_{t-1}\}$
 UpdateAttention($\mathbf{v}, \mathbf{s}, \mathbf{R}, \phi$)
 $\phi \leftarrow .01$
 $c \leftarrow 0$
 end if
 for $R \in \{R_{t_{start}}, \dots, R_{t-1}\}$ **do**
 $d\theta_\pi \leftarrow d\theta_\pi + \nabla_{\theta_\pi} \log \pi(a_t | s_t; \theta_\pi, \mathbf{A})(R - V(s_t; \theta_v, \mathbf{A}))$
 $d\theta_v \leftarrow d\theta_v + (R - V(s_t; \theta_v, \mathbf{A})) \nabla_{\theta_v} V(s_t; \theta_v, \mathbf{A})$
 end for
 $\theta_\pi \leftarrow \theta_\pi + \alpha d\theta_\pi$
 $\theta_v \leftarrow \theta_v + \alpha d\theta_v$
end for

Table 6.2: Hyper-parameter values used for the object collection game.

Parameter	Value	Description
N	250	Number of particles
τ_{BU}	1	Strength of bottom-up attention
τ_{TD}	10	Strength of top-down attention
c_{max}	1000	Frequency of attention updates
t_{max}	10	Maximum length of return
β	0.01	Exploration strength
γ	.99	Discount factor for future rewards
m	8	Number of frames skipped
α	0.0001	Learning rate for Adam optimizer

6.3 Results

6.3.1 Multiple Choice Task

As a baseline to measure the effectiveness of our proposed attention mechanism we compared the performance of *SPA* to a version of *SPA* where each entry of the attention vector was set to a fixed value of $1/K$. This corresponds to attending to all features of VGG-16 equally and we refer to this approach as *SPA_{ALL}*. We also ran an ideal observer model on the multiple choice task to provide a measure of ceiling performance. The ideal observer model selects the image corresponding to the last rewarded object category. All models select a random action with probability ϵ in order to encourage exploration.

Figure 6.6A shows the performance of *SPA*, *SPA_{ALL}* and the ideal observer during training for one random combination of object categories over 5 random seeds. The rewarded image category was changed every 50 trials. In this example *SPA* performs close to optimal as it shows a similar learning trajectory to the ideal observer. In comparison, *SPA_{ALL}* performs poorly and is substantially worse than both *SPA* and the ideal observer. To test the robustness of these findings we ran all three approaches over 5 random seeds on 20 different combinations of object categories. Figure 6.6C shows the results of these simulations. *SPA* out-performed *SPA_{ALL}* during training for every combination of categories that we tested.

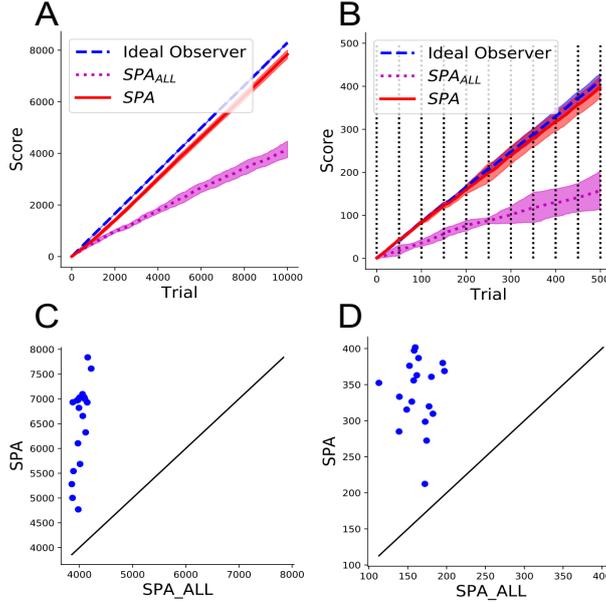


Figure 6.6: Performance of Selective Particle Attention (*SPA*) on the multiple choice task. **(A-B)** Training **(A)** and test **(B)** scores over 5 random seeds for the leopards, faces and soccer balls image categories. The rewarded image category was changed randomly every 50 trials and is represented by a vertical black line in **B**. Error bars represent one standard deviation. *SPA* used selective attention to attend to features that it deemed useful for the current task. *SPA_{ALL}* did not use selective attention and instead attended to every feature of VGG-16. The Ideal observer model represents ceiling performance. **(C-D)** Comparison of *SPA* and *SPA_{ALL}* in terms of total training **(C)** and test **(D)** scores for 20 different image category combinations. Each point represents the mean performance over 5 random seeds for an image category combination. Solid line represents equal performance between *SPA* and *SPA_{ALL}*

While the dynamic attention mechanism of *SPA* appeared to provide a substantial benefit during training, we also wanted to test whether this benefit generalized to unseen images. Figure 6.6B shows the results of the three approaches on the test phase after the training seen in Figure 6.6A. Again the rewarded image category was changed every 50 trials. Importantly the test blocks used images that were not used during training and all the weights of the Deep RL algorithm were frozen so that only the attention vector could change in the case of *SPA*. Again *SPA* exhibited performance similar to that of the ideal observer, while *SPA_{ALL}* showed significantly worse performance. Figure 6.6D shows the test results over 5 random seeds for all 20 of the different category combinations. As with training, *SPA* outperformed *SPA_{ALL}* for all of the category combinations. These results suggests that the benefit of the attention mechanism of *SPA* generalizes well to unseen images.

Both the training and test results suggest that *SPA* is able to cope with changes in the reward function by dynamically re-configuring existing representations for the purpose of state evaluation. Figure 6.7 shows an example of the attention vector during the test phase. The attention vector reliably changes when the target image category changes. This confirms that the attention mechanism of *SPA* is able to use changes in the reward function to re-evaluate the features that need to be attended to. Interestingly, the attention vector is not the same every time a given image category is made the target. This is likely due to the fact that *SPA* will be biased towards attending to features that are present in the first few images, which will be different for each block. In addition, there is likely to be a contextual effect of the rewarded image category in the previous block. For example, if in the previous block the category ‘soccer ball’ was rewarded and in the current block ‘faces’ are rewarded, then this might bias the selection of features that correspond to ‘round’ in the current block.

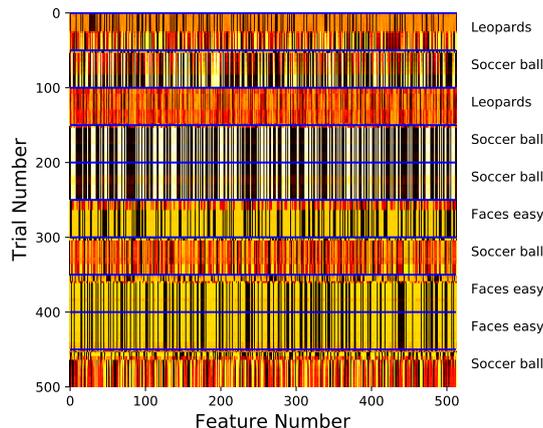


Figure 6.7: *Example of the attention vector values for the leopards, faces and soccer balls image categories. Black represents a value of 0 and white represents a value of 1. The solid blue horizontal line represents a random change in the rewarded image category. The rewarded image category for a given block is presented on the right.*

The results shown in Figures 6.6B and 6.6D indicate that the performance of *SPA* can vary depending on the combination of image categories that are chosen. This suggests that *SPA* finds it easier to discriminate between certain image categories compared to others based on their features. Figures 6.8A and 6.8B show the mean feature map values of VGG-16 for the image categories that *SPA* performed best (leopards, faces and soccer balls) and worst (cups, chandeliers and cellphones)

on. In the best case scenario, the feature maps contain several features that are substantially more active than others. This likely provides a good substrate for bottom-up attention because there are a handful of features that are reliably more active than the others, which corresponds to a strong prior over hypotheses. In comparison, in the worse case scenario, the features take on a more uniform distribution of activation values and so the prior over hypotheses is weaker.

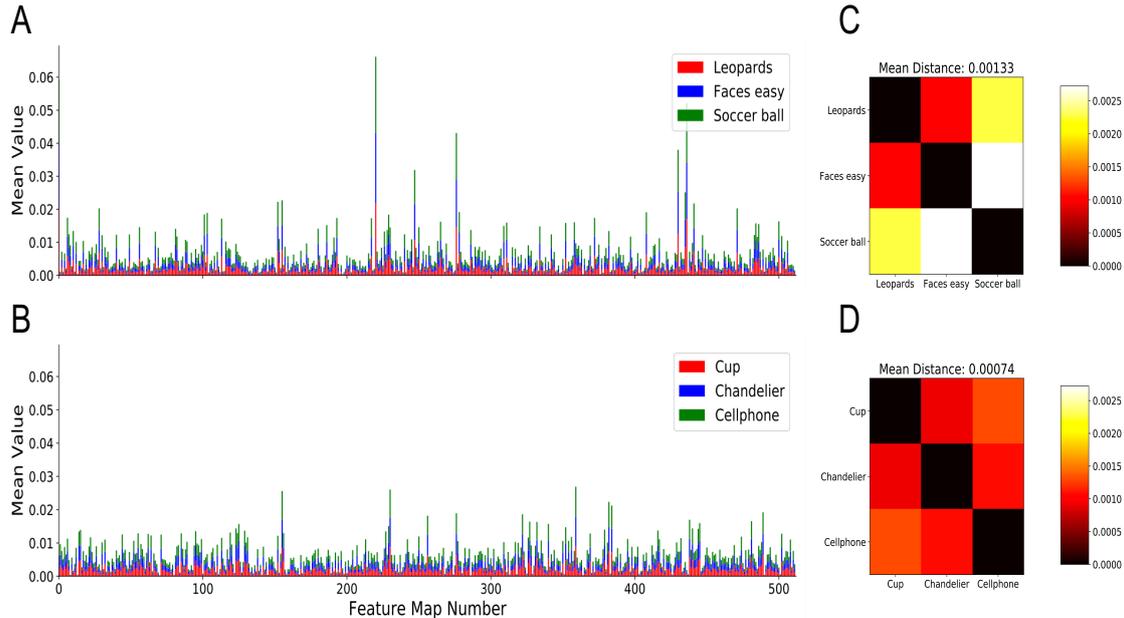


Figure 6.8: *Analysis of the feature map values produced by VGG-16 during the multiple choice task. (A-B) Average feature map values of VGG-16 for the combination of image categories that lead to the best (A) and worst (B) training performance of SPA. (C-D) Euclidean distances between the average feature map values of VGG-16 for the combination of image categories that lead to the best (C) and worst (D) training performance of SPA.*

Bottom-up attention aside, having a few highly active features is only useful for the multiple choice task if they help to discriminate between the different image categories. As seen in Figures 6.8A and 6.8B, each image category produces 512 average feature values, which can then be expressed as a vector in Euclidean space. Figures 6.8C and 6.8D show the Euclidean distance between these vectors for the best and worst case scenarios respectively. In the best case scenario, the euclidean distance between the majority of the image categories is larger than in the worst case scenario. In addition the mean euclidean distance over all pairwise comparisons in the best case scenario is nearly double that of the worst case scenario. This suggests that the categories in the best case scenario are easier to discriminate

between because they are further apart in euclidean space. This is likely to help the top-down attention of SPA because each particle will produce very different value estimates depending on the image category being attended to.

6.3.2 Object Collection Game

As with the multiple choice task, we compared SPA to a baseline approach that attended to all features of VGG-16, which we refer to as SPA_{ALL} . In addition to SPA_{ALL} , we also included a condition that set the attention vector to a random binary vector, which was then normalised to sum to 1. This condition was included to account for the fact that random feature reduction may lead to improved performance and we refer to it as SPA_{RANDOM} .

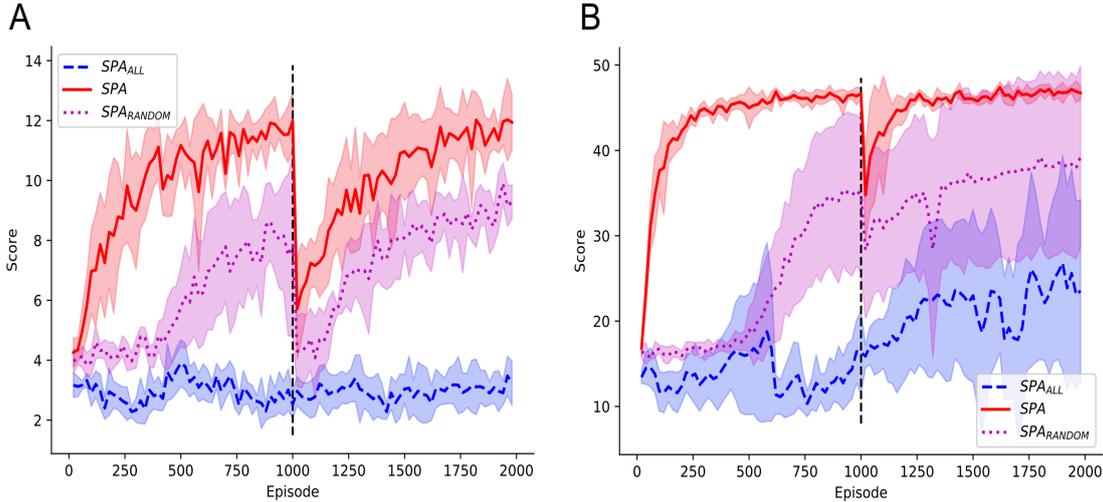


Figure 6.9: *Performance of SPA , SPA_{ALL} and SPA_{RANDOM} on the object collection game. Each point is the average score over the last 20 episodes. Error bars represent one standard deviation. Each approach was run over 5 different random seeds and the error bars represent one standard deviation. SPA used selective attention to attend to features that it deemed useful for the current task. SPA_{ALL} did not use selective attention and instead attended to every feature of VGG-16. SPA_{RANDOM} attended to a random subset of features, which changed with each random seed. (A) The vertical dashed line indicates where the rewarded object was changed without warning. (B) All objects were the same colour and shape. The vertical dashed line indicates where the shape and colour of these objects was changed without warning.*

Figure 6.9A shows the results of SPA , SPA_{ALL} and SPA_{RANDOM} on the reward revaluation task. Focusing on the first 1000 episodes before the change in reward function, SPA significantly out-performed SPA_{ALL} and SPA_{RANDOM} over

the course of learning. SPA_{ALL} saw the worst performance showing no evidence of learning over the first 1000 episodes. This suggests that naively learning over all features is a highly ineffective strategy given a limited amount of experience. In contrast, SPA_{RANDOM} showed evidence of learning after around 500 episodes. This learning was highly variable as would be expected given that the features were randomly selected for on each random seed. Nevertheless this demonstrates that simply reducing the number of features at random is sufficient to provide a learning benefit. Out of all of the approaches, SPA performed the best on the first 1000 episodes for several reasons. Firstly, SPA began improving its score almost immediately after a handful of episodes. This onset of learning is noticeably earlier than the other approaches. Not only did learning occur earlier but it was also much faster, as indicated by the sharper increase in score compared to the other approaches. As a result, by episode 1000 SPA was able to achieve a higher score than the other approaches. Importantly, SPA was also more robust than SPA_{RANDOM} as indicated by the smaller standard deviation in Figure 6.9A. Overall, these results suggest that the particle filter mechanism of SPA is able to identify features useful for value computation and that this translates into a substantial learning benefit given the current task.

Upon the change in reward function after 1000 episodes, SPA_{ALL} continued to demonstrate no evidence of learning. In comparison, SPA_{RANDOM} showed a marked decrease in performance as soon as the reward function changed and was slow to recover performance. SPA also showed a marked decrease in performance immediately after the change in reward function. This is to be expected as the agent had no prior warning that the reward function was about to change. However, the performance of SPA dropped to a level that was still above SPA_{RANDOM} and remained significantly higher than SPA_{RANDOM} throughout the recovery in performance. This resulted in SPA achieving a higher score than all other approaches on the final episode. These results suggest that the particle filter mechanism of SPA is better equipped to handle changes in the environment by quickly re-evaluating which features are important for the current task.

Figure 6.9B shows the results of the three approaches on the change in state space task. As before, for the first 1000 episodes SPA demonstrated evidence of

learning that occurred earlier, faster and more robustly than the other approaches. Upon the change in state space after 1000 episodes, *SPA* showed a slight decrease in performance but was quickly able to recover from it. In comparison, *SPA_{RANDOM}* also saw a drop in performance but the rate of recovery was slower, while *SPA_{ALL}* continued its slow gradual learning. For both *SPA_{RANDOM}* and *SPA_{ALL}*, performance for the second 1000 episodes was highly variable and significantly lower than *SPA*. These results further support the idea that the selective attention mechanism of *SPA* is able to quickly adapt to changes in the current task by re-orientating its attention towards a different set of features. In addition, the consistently high performance of *SPA* and its fast rate of recovery during the change in state space indicates that much of the knowledge learnt by the Deep RL algorithm in the first 1000 episodes was still of use in the second 100 episodes. This suggests that the selective attention mechanism of *SPA* is also able to promote the transfer of knowledge between tasks.

6.4 Discussion

When presented with a visual scene we need to be able to quickly identify the relevant features based on our current goal. For example, if we are in a forest and are looking to start a fire then we need to attend to features indicative of dry wood. Conversely, if we are thirsty then we must attend to features that are indicative of a water source. This goal driven modulation of perceptual features is often referred to as selective attention and is thought to be the responsibility of the Pre-Frontal Cortex (PFC) (Desimone and Duncan, 1995; Miller and Cohen, 2001). However, it remains an open question how we are able to learn which features of the environment to attend to given the current goal. Deep Reinforcement Learning (RL) represents a promising avenue for modelling how humans learn to map raw perceptual input to goal-driven behaviour. Deep RL systems typically rely on the incremental learning of representations via backpropagation in Deep Neural Networks (DNNs). As a result they lack the ability to quickly adjust their representations given a sudden change in task.

To address these issues and imbue Deep RL agents with the ability to perform

selective attention we propose a novel algorithm called *Selective Particle Attention (SPA)*. *SPA* uses a pre-trained deep convolutional neural network VGG-16, to extract features that are similar to those found in the visual stream (Schrimpf et al., 2018). Rather than using gradual error-driven learning, *SPA* uses a particle filter to learn attention values that filter these features based on the current goal (Radulescu et al., 2019; Radulescu et al.). Each particle in the particle filter represents a hypothesis about which set of features are important for the task at hand, which has been proposed to be the role of the PFC (Miller and Cohen, 2001; Radulescu et al., 2019). Crucially, the particle filter combines bottom-up and top-down attention to perform selective attention (Desimone and Duncan, 1995; Treue and Trujillo, 1999). Bottom-up attention occurs during the movement step of the particle filter; particles are biased to attend to features that are highly active. Top-down attention occurs during the observation step; particles that are better at predicting reward are more likely to be re-sampled. In this way hypotheses are biased towards features that are highly active and are constantly evaluated based on their ability to predict reward (Mackintosh, 1975). The particles are used to generate attention values by calculating the mean particle state and normalising it to sum to one. This is consistent with other computational models of visual attention that highlight normalisation as a key computation for capturing competition between features (Reynolds and Heeger, 2009). The attention values are used to filter the features produced by VGG-16, which are then passed on to a Deep RL algorithm to evaluate the current state of the environment. This evaluation process is thought to be the responsibility of the striatum through the use of cortical inputs (Schultz, 1998; Houk et al., 1995; Joel et al., 2002; Maia, 2009; Setlow et al., 2003) and temporal difference errors (Schultz et al., 1997).

We evaluated *SPA* on two different tasks. The first task was a multiple choice task involving naturalistic images. Three object categories were chosen at random and on each trial the agent was presented with an image from each of the categories. Trials were organised into blocks and for any given block only one of the object categories would result in a reward. *SPA* was able to achieve close to ceiling performance on this task for several examples and dramatically out-performed a naive version that attended to all features. Inspection of *SPA*'s attention vector

showed that it was able to quickly change its configuration in response to changes in the reward function. This highlights how selective attention can be used to quickly respond to unannounced changes in the environment and improve the efficiency of learning.

The power of RL is its ability to deal with temporal dependencies and to make actions that lead to reward in the future. The aforementioned multiple choice task can be viewed as a single step Markov Decision Process (MDP). We therefore wanted to test if *SPA* could be applied to more complex domains and multi-step MDPs. With this in mind we chose the second task to be a simple 2D video game. The agent was in control of a rectangular block that could be moved left and right. The goal of the agent was to collect objects that moved from the top of the screen to the bottom based on their shape. As with the multiple choice task, *SPA* was able to perform significantly better than a naive approach that attended to all features. It also performed better than an approach that selected a random subset of features to attend to. This provides evidence that *SPA* can improve the efficiency of learning on complex RL problems that require temporal dependencies. In addition, it demonstrates that *SPA* is not dependent on a specific Deep RL algorithm as long as the algorithm uses a form of value approximation.

Interestingly, in order to successfully apply *SPA* to the 2D object collection game the attention vector was only updated every 1000 time-steps rather than on every time step. This was necessary because it allowed the Deep RL algorithm time to respond to the change in attention and attempt to learn a useful function of the currently attended features. This is akin to a person learning based on a single hypothesis for a fixed amount of time before deciding whether to change to another hypothesis or not. This may explain why attentional inertia is prevalent in children and adults (Anderson et al., 1987; Burns and Anderson, 1993; Richards and Anderson, 2004; Longman et al., 2014), as the brain requires time to evaluate a given hypothesis before deciding whether to switch attention. Future work should systematically explore this apparent trade-off between the potential benefits of switching to a new hypothesis and the time needed to sufficiently evaluate a hypothesis.

Part of the reason for the design of the object collection game was that it allowed for the rewarded shape to be easily changed during learning. This change can

be manipulated to correspond to a change in the reward function or state space. We found that *SPA* was significantly better equipped to deal with such changes compared to other naive approaches that either attended to all or a random subset of features. In particular, *SPA* showed an extremely fast recovery in response to a change in state space. This is often considered an extremely hard problem as deep neural networks typically fail catastrophically when the input distribution is changed (Lake et al., 2017). Nevertheless, the small drop in performance and rapid recovery to asymptotic levels suggests that *SPA* was able to transfer much of the knowledge that it had acquired before the change in state space. These results further support the findings of the multiple choice task, which demonstrated *SPA*'s improved ability to deal with changes in the environment.

We propose that the ability of *SPA* to focus on a subset of features based on the current task is beneficial for several reasons. Firstly, it helps to reduce the dimensionality of the problem, which reduces the impact of noisy features and the complexity of the function that needs to be learnt. This improves generalization because the learned function does not fit spurious features and therefore ignores any changes to them. This benefit of dimensionality reduction can even be seen in the case of *SPA_{RANDOM}*, which out-performed *SPA_{ALL}* on the object collection game. *SPA* takes this one step further however, by providing targeted dimensionality reduction rather than selecting a random subset of features. Furthermore, the selective attention of *SPA* helps to reduce interference by guiding learning onto different sets of features based on the task at hand. This results in the learning of different sub-networks that can be quickly identified based on the current task. This greatly improves the capacity of the Deep RL algorithm to represent different functions and to switch between them in the face of changes to the environment.

Selective attention in *SPA* is implemented using a particle filter that captures the influences of both bottom-up and top-down attention. Previous work has already shown that particle filters may represent a viable computational account of selective attention during learning based on their ability to fit eye-tracking data (Radulescu et al.). Here we extend this work to show how they can be interfaced with Deep RL principles in order to learn which features to attend to just from raw pixel values and external reward signals. One of the primary benefits of using a particle

filter to implement selective attention is that it relies on sampling to produce an approximate value of a hidden variable. This is important because the potential number of feature combinations that need to be evaluated for a given task can be extremely large. In our case there were 2^{512} potential feature combinations and so it would be computationally infeasible to evaluate all of them. However, by using only 250 particles we were still able to converge to a satisfactory solution thanks to the iterative re-sampling procedure of the particle filter. Future work should explore how the number of particles in *SPA* affects its ability to find the best combination of features and therefore its asymptotic performance on tasks.

The approximate inference of the particle filter in *SPA* is guided by bottom-up attention, which introduces a bias towards the most active features. This introduces a prior over the hypothesis space, favouring hypotheses with relatively few features. This is consistent with findings that people tend to make decisions based on individual features before reasoning about objects that involve more complex combinations of features (Farashahi et al., 2017; Choung et al., 2017). Similarly, people find it harder to perform classification tasks as the number of relevant dimensions increases (Shepard et al., 1961). The bottom-up attention captured in the movement step of the particle filter therefore seems to introduce a biologically plausible bias over the hypothesis space.

Top-down attention is captured during the observation step of the particle filter. This process involves evaluating different hypotheses based on their ability to predict reward (Mackintosh, 1975). Interestingly it has been proposed that such a mechanism may occur in corticostriatal circuitry (Radulescu et al., 2019). Different pools of neurons in the PFC may represent different hypotheses about the structure of the current RL task. These pools then compete via mutual lateral inhibition and this competition is biased via connections to the striatum that favour pools which lead to reward. This process parallels the observation step of *SPA*, whereby hypotheses that are more predictive of reward are more likely to be re-sampled and out-compete other hypotheses. The specific mechanism aside, the phenomenon of representing and testing multiple hypotheses during RL appears to be prevalent in human populations (Wilson and Niv, 2012).

The effectiveness of *SPA* will naturally depend on the nature of the features or

representations that it is attending too. In the human brain it has been proposed that the usefulness of selective visual feature attention decreases as you move back through the visual stream (Lindsay and Miller, 2018; Baluch and Itti, 2011). This is because features present later in the visual stream consist of higher-order representations that are increasingly abstract. For example one of the object categories in the multiple choice task was faces, which are known to be represented later on in the visual stream of the brain (Grill-Spector et al., 2018). Having such a representation makes the multiple choice task easy for the brain because it only has to attend to one feature rather than a collection of low level features. This also reduces the need to consider lots of complicated hypotheses. Future work should test whether *SPA* displays similar behaviour by testing whether its performance decreases as attention is applied to earlier convolutional layers.

For both the multiple choice task and the object collection game, the weights of VGG-16 were frozen so that no further learning occurred in the network. This decision was made to reflect the fact that learning in the visual stream is slow and so the brain needs to pick from pre-existing representations. If the brain had to rely on learning new sensory representations for every task then learning would be extremely slow and the learning of one task would over-write the learning of a previous task. By using attentional mechanisms on pre-existing representations, the brain can greatly increase the speed of learning and learn different sub-networks depending on the task at hand, which reduces interference between tasks. However, gradual learning does still occur in the visual stream and is thought to underlie the gradual change from novice to expert on perceptual categorization tasks (Schyns et al., 1998). From the perspective of *SPA*, allowing gradual learning in VGG-16 would lead to the emergence of new features for the particle filter to attend to and for the Deep RL algorithm to utilise. As the learning would be slow and occur over several tasks, these new features would generalize over the tasks being repeatedly performed by the agent. Future work should therefore explore whether allowing slow incremental learning in VGG-16 would improve the performance of *SPA* over a continuous sequence of different tasks.

While no learning occurred in VGG-16 during our simulations, we did use network weights that were the result of extensive pre-training. Importantly, the data

that was used to pre-train VGG-16 involved images that were not used for the multiple choice task and the object collection game. In theory, using the original Imagenet dataset that it was trained on may have lead to better results in the multiple choice task. This is because VGG-16 will have already been trained to produce feature values that distinguish between the object categories in Imagenet, making it easier for *SPA* to attend to discriminating sets of features. This hypothesis is supported by our analysis of the features produced by VGG-16 during the multiple choice task. For the combination of image categories that *SPA* performed best on, the vectors of mean feature values were further apart in Euclidean space compared to the combination of image categories that *SPA* performed worst on. This suggests that the more dissimilar the feature values are between the different image categories, the easier it is for *SPA* to discriminate between them and quickly change its focus of attention.

This dependence of *SPA* on the properties of the underlying representations that it attends to opens up several interesting avenues of future research. In particular, it would be interesting to explore whether the use of disentangled representations (Higgins et al., 2016, 2018) could further improve performance. Independent factors of variation may be easier to attend to because the informative features are separate from each other and so only simple hypotheses are required to solve the task at hand rather than complex combinations of features. Another major benefit of disentangled representations would be that the resulting attention vector would be more interpretable as each attended feature has a natural interpretation; e.g., colour or shape.

6.5 Concluding Remarks

In this chapter we have presented a novel method for performing selective visual feature attention in a Deep-RL agent. Our approach, termed Selective Particle Attention (*SPA*), uses a particle filter to identify useful pre-existing features based on the task at hand. These features are then passed on to a Deep-RL algorithm to perform action selection. The particle filter incorporates bottom-up and top-down influences into the selective attention process via the movement and observation

steps respectively. The movement step serves to introduce a prior over the features that are being considered so that attention is biased towards the most active features given the current input. In comparison, the movement step biases attention towards combinations of features that are most predictive of reward. Crucially these two interacting processes help to reduce the dimensionality of the learning problem in a targeted manner. This not only speeds up the efficiency of learning but also improves the agent's ability to deal with changes in the environment. Future work should explore how the nature of pre-existing representations affects the attention mechanism of *SPA*. The depth of the representation, the data used to produce them and their degree of disentanglement are all variables that may impact upon the performance of *SPA*.

The work in this chapter again highlights the importance of considering the brain as a multitude of different learning systems all interacting to support efficient RL. In particular, it supports the proposal to divide the neocortex into the PFC and sensory cortices so that phenomena such as selective attention can be captured. Importantly, *SPA* only describes one benefit of this division and it is likely that many other benefits also exist. For example we saw in Chapter 5 that attentional signals from the PFC to the hippocampus may be responsible for concept formation and that the PFC may implement a meta-learning algorithm with the striatum. It is therefore likely that all of these process work together to help support efficient RL. This is a similar conclusion to Chapter 3, which explored how the division between the neocortex and the hippocampus is important for efficient RL.

In the next chapter we will move away from using computational approaches to investigate the underlying computations that support efficient RL in the brain. Instead, we shall turn to an empirical investigation of how external factors in the environment can affect the efficiency of RL. In particular, we shall focus on how the degree of perceptual similarity between consecutive experiences can influence the transfer of information between related experiences. To achieve this we get human participants to play 2D video games that can vary in terms of perceptual surface features but that all share the same underlying rules. This allows us to vary the perceptual similarity between consecutive games while ensuring that they are related. We discuss the results of this work in light of the analogy between the brain

and Deep RL that we have built thus far. Our hope is that this empirical work will help to constrain computational accounts of transfer in the brain and therefore deepen our understanding of efficient RL.

Chapter 7

The Effect of Perceptual Similarity on Transfer in Humans

Overview

In this chapter we present novel empirical work exploring how the degree of perceptual similarity between consecutive experiences affects transfer. We therefore switch from focusing on the internal computations that support efficient Reinforcement Learning (RL) to the external factors that influence it. We conduct three experiments that test participants' ability to transfer relational rules between problems. Importantly, for each experiment the perceptual features between consecutive problems are varied to different extents but the relational rules are preserved to allow for transfer. In Experiment 1 (Section 7.2) we investigate the effect of perceptual similarity on transfer in a simple match-to-sample task. In Experiment 2 (Section 7.3) we use a similar experimental design but apply it to a more naturalistic setting; a 2D video game. Finally, in Experiment 3 (Section 7.4) we repeat Experiment 2 but tell the participants the rules of the games before-hand. Our results demonstrate that (1) participants can perform 'zero-shot' learning by transferring the relational rules, (2) participants can also perform 'one-shot' learning by transferring task structure, and (3) participants are better at using explicit knowledge for transfer when the degree of perceptual similarity between consecutive experiences is high. We hope that these results can be used to help constrain theories of transfer in the brain and improve the ability of Deep RL approaches to model transfer in the brain.

7.1 Introduction

Previous chapters have used Complementary Learning Systems (CLS) theory to explore how different learning systems in the brain and their interactions contribute to efficient Reinforcement Learning (RL). These internal computations represent fundamental mechanisms for rapid learning and transfer in the human brain. However, they do not operate in isolation and are naturally affected by external factors present in the environment. For example, the ability to transfer knowledge between experiences or tasks is likely to interact with variables such as the number of related previous experiences and the time between these experiences. This chapter focuses on one specific variable; the perceptual similarity between consecutive experiences. In contrast to previous chapters, this investigation will use human adults as subjects and rely on the methods of experimental psychology.

Consecutive experiences can have varying degrees of perceptual similarity and different bodies of literature (discussed in detail below) produce conflicting predictions about whether high or low perceptual similarity should benefit transfer. An answer to this question may help us to further understand the internal computations underlying transfer by providing empirical constraints on potential theories and computational models. The rest of this section will outline the conflicting predictions made by different bodies of literature about how the degree of perceptual similarity between experiences should affect transfer.

7.1.1 Analogy: The Relational Shift and Progressive Alignment

The problem of transfer in Reinforcement Learning (RL) can be framed as an analogy problem, whereby one needs to match the current environment to a previous environment(s) and use the inferred similarities between them to guide decisions. With this in mind, the literature on analogical reasoning makes a prediction about how perceptual similarity should interact with the ability to transfer. The theory of ‘progressive alignment’, proposed by Gentner and Markman (1994), suggests that solving analogical problems with a high degree of perceptual similarity should facilitate the solving of harder analogical problems with a low degree of

perceptual similarity. This occurs because problems with a high degree of perceptual similarity allow for easier more concrete abstractions, which then increase the probability of being able to form harder abstractions between problems with low perceptual similarity. These abstractions between experiences are critical because they allow people to generalise to new experiences and become less dependent on perceptual features.

A wealth of empirical evidence has been put forth to support progressive alignment. One such study by Kotovsky and Gentner (1996) presented arrays of shapes such as three circles or three triangles, to 4-year-old children. These arrays could vary along different dimensions, for example each shape could increase in size or saturation within an array. This meant that arrays could either match according to low order, perceptual commonalities e.g. circles increasing in size and triangles increasing in size, or they could match according to higher order, abstract similarities across dimensions e.g. circles increasing in size and circles increasing in saturation. For the task, the children were shown one array and then asked to choose a subsequent matching array without any feedback from their previous choice. Kotovsky and Gentner (1996) found that children could solve the low order matches when the arrays were presented randomly but not the high order matches. However, when they were presented in order, starting with low order matches and moving to high order matches, the children had improved performance on the high order matches. These findings demonstrate how gradually dissimilar past experiences can help to iteratively improve a child's ability to learn abstractions for transfer.

Another example of progressive alignment and how perceptual similarity affects transfer comes from a study by Gentner et al. (2007). In this study children were shown a picture of an alien and were told that the alien had a body part with a novel name e.g. 'a dibble'. The children were then shown other aliens and were asked which one also had that body part. Two versions of this task existed, one where the options were perceptually similar to the exemplar and one where they were perceptually dissimilar. Interestingly, children performed better on the perceptually dissimilar version when they were first presented the perceptual similar version even though feedback was not given. This is a direct example of how comparing high similarity experiences can improve a child's ability to transfer knowledge to low

similarity experiences.

These studies show that consecutive experiences with high perceptual similarity improve children’s ability to form abstractions and perform transfer. However the degree of perceptual similarity between experiences lies on a continuum rather than at the two extremes. It is therefore likely that for a given individual there is an ideal degree of perceptual similarity for transfer, whereby the differences between experiences allow for a high degree of abstraction but are not so different that no abstraction can be formed. If the perceptual similarity is too high then the abstraction may be weak and lead to an over-reliance on misleading perceptual commonalities. Equally if the perceptual similarity is too low then the abstraction may be too difficult to acquire. Where this optimum degree of perceptual similarity lies will likely depend on a range of factors including the age of the person and the nature of the abstraction itself.

It is worth noting that progressive alignment not only appears to operate over short time-scales (e.g. during a psychology experiment) but also operates over long time-scales. Gentner and Hoyos (2017) have proposed that progressive alignment also underlies development as people gradually rely less and less on perceptual similarities and instead focus on abstract relational information to guide their behaviour. This phenomena has been termed the ‘relational shift’ and has been well characterised during child development (Gentner, 1988; Richland et al., 2006). Importantly this shift from perceptual reasoning to abstract relational reasoning appears to be domain specific (Gentner and Rattermann, 1991), suggesting that it is indeed past experiences in a particular domain that drives these changes rather than some general processing change.

7.1.2 Concept Learning and Category Structure

While the analogy literature predicts that past experiences with high perceptual similarity should bootstrap the ability to transfer to experiences with low perceptual similarity, other bodies of literature disagree with this prediction. In particular, the concept learning literature provides conflicting predictions about how the degree of perceptual similarity between experiences should affect transfer. Several studies have highlighted how randomly presenting exemplars from different conceptual categories

can improve the transfer of knowledge, as measured by classification performance on novel exemplars (Kornell and Bjork, 2008; Rohrer et al., 2014). These findings suggest that minimising the perceptual similarities between exemplars should facilitate transfer with respect to the task of classification. Interestingly these findings are not unanimous; several other studies on concept learning have found evidence for improved acquisition of conceptual knowledge when exemplars have been presented in a blocked manner; i.e. when exemplars are blocked in a way that maximises perceptual similarity (Kurtz and Hovland, 1956; Whitman and Garner, 1963; Goldstone, 1996).

Work by Carvalho and Goldstone (2014b) has tried to reconcile these opposing results by investigating how perceptual similarity within and between categories affects whether random or blocked category learning is beneficial for concept learning. Carvalho and Goldstone (2014b) presented participants with both random and blocked learning protocols but one group was given categories with *high* within- and between-category similarity while another group was given categories with *low* within- and between-category similarity. The results of the study showed that randomly organised exemplars lead to improved generalisation in the high similarity group whereas blocked exemplars lead to improved generalisation in the low similarity group. Carvalho and Goldstone (2014b) hypothesised that these results were due to the fact that random exemplars help participants to attend to the differences between categories because different categories are presented in quick succession. This is particularly important in the high similarity case because the differences are hard to detect and are more informative than the similarities. In comparison, blocked exemplars help participants to attend to the similarities within a category because exemplars from the same category are presented in quick succession. This is useful in the low similarity case because the similarities are hard to detect and are more informative than the differences. It therefore appears that perceptual similarity is a key factor that affects whether people attend to similarities or differences between experiences and this interacts with whether similarities or differences are useful for the task at hand.

7.1.3 Insights From Deep Reinforcement Learning Approaches

Throughout this thesis we have used an analogy between the brain and Deep Reinforcement Learning (RL) to understand how humans perform efficient RL through rapid learning and transfer. With this in mind, what does this analogy predict about the impact of perceptual similarity on transfer? The answer to this question lies in the way that Deep RL models are trained.

Deep RL methods rely on the use of Deep Neural Networks (DNNs) to approximate a value function and/or a policy. These DNNs are sensitive to the order in which the input data is presented because they are typically trained using stochastic gradient descent. If inputs are spuriously correlated then the parameters of the network will be updated to reflect these correlations, which ultimately leads to local over-fitting. To combat this problem DNNs are trained in an interleaved fashion with the input data being presented in a random order. Indeed, classic Deep RL approaches, such as the Deep Q-Network (DQN) (Mnih et al., 2015), train a DNN by randomly sampling from a memory buffer of past state transitions. This helps to address the fact that the input to these methods is a constant stream of temporally correlated perceptual observations. Interestingly, this has been tentatively compared to biological ‘replay’ whereby the hippocampus randomly reinstates activity patterns from past experiences (see Section 3.4.1).

This demonstrates that classic Deep RL methods rely on training mechanisms that reduce the degree of perceptual similarity between training examples so that the network generalises well. These training examples could represent state transitions within a single task (e.g. a level of an Atari video game) or state transitions from different tasks (e.g. different levels of an Atari game). Either way, the DNN will attempt to fit the underlying similarities between the training examples and so the fewer spurious perceptual similarities the better the transfer.

More recent work in Deep RL, which attempts to address rapid learning and transfer, also predicts that reducing the perceptual similarity between experiences will benefit transfer. For example, one recent and encouraging approach to modelling transfer is that of meta-learning (see Section 5.2.1) (Wang et al., 2016). A key component of meta-learning is the use of an outer-RL algorithm to improve the generalisation of an inner-RL algorithm. Importantly this process requires that tasks

are sampled from a pool of tasks in a random manner. As with DQN, this random sampling reduces spurious correlations between tasks and ultimately prevents overfitting, only this time it explicitly focuses on the correlations between tasks.

If Deep RL approaches appear to favour low perceptual similarity between experiences then what does this mean for our predictions about the brain? If we assume, as has been the case throughout this thesis, that DNNs are a useful model of the neocortex and semantic memory, then cortical learning should also benefit from consecutive experiences with low degrees of perceptual similarity. However, as we have highlighted through the computational work in this thesis, the neocortical learning system does not work alone and relies on the complementary properties of other learning systems such as the hippocampus. Indeed, just as DQN relies on randomly sampling from a memory buffer, the neocortex may rely on replay from the hippocampus. If this is the case, then Deep RL may instead predict that the degree of perceptual similarity between experiences should make little difference to transfer ability because the brain has compensatory mechanisms to account for spurious correlations in the perceptual input. That said, empirical research into biological replay has suggested that it occurs offline, such as during periods of rest or sleep. As a result, replay is likely unable to address the issue of highly correlated perceptual input during tasks that do not afford offline processing and that occur over short time scales.

Aside for the influence of other learning systems, it appears that the nature of the representations learnt by DNNs may also reduce their reliance on randomised training data. In particular, it has been suggested that Deep RL approaches that use disentangled representations (see Section 3.2.3) may be less reliant on interleaved training. The main premise behind disentangled representations is that each output unit of a DNN encodes a cardinal dimension that humans use to categorise or recognise an object; e.g. colour, shape or rotation. As a result, if the abstraction being learnt involves a single cardinal dimension, then transfer should be less dependent on interleaved training because there is less interference from other dimensions. In comparison if the representations are entangled or the abstraction being learnt involves multiple cardinal dimensions then interference will occur and perceptual similarity should be reduced in order to negate the effect of spurious correlations.

From the perspective of the brain it has been well documented that the learning of verbalisable rules along cardinal dimensions recruits different learning systems compared to the learning of statistical rules that require integration across dimensions (e.g., Ashby and Maddox, 2005). It is therefore likely that the affect of perceptual similarity on transfer depends on whether the abstraction is a simple verbalisable rule or a non-verbalisable statistical rule and the degree of entanglement in the representations.

Evidence for this interaction has come from a recent study by (Flesch et al., 2018). In this study, Deep RL models and participants were faced with two simple RL tasks that involved categorising images of trees based on their ‘branchiness’ and ‘leafiness’. Each task was identified by a different background scene. The tasks were presented in either an interleaved or blocked manner and each task had its own abstraction or rule that dictated whether reward was obtained or not. Two different sets of rules were used; one set was cardinal and operated along a single dimension (either ‘branchiness’ or ‘leafiness’) and the other set was diagonal in that it integrated across the two dimensions. Test/transfer performance was assessed using a set of interleaved test trials involving novel tree stimuli.

In the case of the human participants, when the rules were cardinal, people performed better at test if they had received the blocked training condition. This benefit of blocked training was even greater if the person had a strong prior bias to categorise the tree stimuli based on their ‘branchiness’ and ‘leafiness’. This suggests that the blocked design (1) helped participants to factorise the problem into the two cardinal rules and (2) was most beneficial when people represented the problem in a disentangled manner along the cardinal dimensions. Interestingly, even if the rules were diagonal (i.e. they required integration over the two dimensions), participants appeared to show improved learning of the decision boundary in the blocked condition despite similar overall test performance compared to the interleaved condition. These findings demonstrate that grouping experiences by perceptual similarity can confer transfer benefits within the context of an RL problem. In addition, this benefit is particularly prominent when the rule or abstraction being learnt is cardinal and verbalisable.

With respect to the Deep RL models, Flesch et al. (2018) first explored how the

performance of a standard Deep Convolutional Neural Network (DCNN) performed on the tasks. As expected the DCNN demonstrated ceiling performance under the interleaved training regime regardless of whether the rules were cardinal or diagonal. However, under the blocked regime the DCNN suffered and could only remember the last task that it was trained on. These results are a clear demonstration of how entangled representations favour interleaved and perceptually dissimilar exemplars in order to generalise and transfer successfully. Flesch et al. (2018) then explored how a DCNN with disentangled representations performed on the two training regimes. In this case, the disentangled DCNN still showed diminished performance on the blocked training regime but performance was significantly better than the DCNN with entangled representations. This demonstrates that disentangled representations can alleviate some of the dependency of neural network models on interleaved training.

Taken together, the findings of Flesch et al. (2018) help to highlight the intricate interactions between representations, types of rule and grouping of experiences based on perceptual similarity. In particular, maximising perceptual similarity appears to have a genuine benefit for human learning, at least in the case of task switching and the categorisation of novel stimuli. In contrast, blocked training appears to be detrimental to learning in neural networks, although some of this can be alleviated through the use of disentangled representations.

In summary, the training of DNNs in Deep RL systems generally predict that low perceptual similarity between experiences should promote transfer. The influence of other learning systems and the use of disentangled representations can potentially reduce the reliance of DNNs on interleaved training. However, even with these alterations, there is no Deep RL system that predicts that transfer should benefit from experiences with a high degree of perceptual similarity. This appears to be at odds with the theory of progressive alignment and also the results of Flesch et al. (2018), who reported improved transfer after blocked training on two RL tasks. The rest of this chapter explores this issue in a series of 3 studies with human adults.

7.1.4 Overview of Human Experiments

The purpose of the present chapter is to explore the effect of perceptual similarity between experiences on transfer in humans. Critically, we explore this in a naturalistic Reinforcement Learning (RL) environment, which involves both sequential decision making and motor control. To achieve this, we designed video games that could differ in terms of perceptual similarity but used the same verbalisable relational rules. We then generated sequences of these games with high or low perceptual similarity between consecutive games and investigated which sequence improved transfer performance in human participants.

As we saw in the previous section, much of the existing literature has focused largely on the effects of interleaved vs. blocked training. A blocked regime induces high levels of perceptual similarity between experiences whereas an interleaved regime induces low levels of perceptual similarity. However, these training regimes generally involve distinguishing between categories (Kornell and Bjork, 2008; Rohrer et al., 2014; Goldstone, 1996; Kurtz and Hovland, 1956; Whitman and Garner, 1963), or in the case of Flesch et al. (2018), between tasks. In the present study, rather than focusing on discriminating between concepts or tasks, we instead focus on learning a single concept or task. In other words, all of the games presented to the participants are related and relevant for learning a meaningful strategy. This is akin to learning a single conceptual category rather than learning to discriminate between several categories. We are therefore investigating how the degree of perceptual similarity within a category or a set of related tasks can affect transfer performance. In addition, our task involves playing active video games which, compared to classic concept learning and classification tasks, is a big step in the direction of naturalistic behaviour involving motor control and real-time sequential decision making. We believe that it is important to take this step away from the simple presentation of exemplars and single decisions, in order to see if the aforementioned findings generalise to more naturalistic decision-making problems.

Despite these differences between our experimental paradigm and the literature reviewed in the previous section, these studies still make several interesting predictions about whether consecutive games with high or low perceptual similarity should promote transfer. The theory of progressive alignment predicts that consec-

utive games with high perceptual similarity should help to bootstrap learning and therefore improve transfer to more dissimilar games later on. However, as previously mentioned there is likely to be a ‘sweet spot’ of perceptual similarity, whereby consecutive games are similar enough to form abstractions over but different enough to maximise the degree of abstraction. In comparison, the work of Carvalho and Goldstone (2014b) suggests that high perceptual similarity between consecutive games should encourage people to focus on the similarities between them. In our case, this predicts that consecutive games with high perceptual similarity should improve transfer because all the games are related and so highlighting the similarities will help to identify the consistent underlying rules.

In terms of the analogy between Deep RL and the brain, we have already seen that the training of DNNs is generally reliant on interleaved training, which minimises spurious correlations in the input. This suggests that consecutive games with low perceptual similarity should help participants to avoid learning strategies that involve spurious perceptual features and therefore promote transfer. Some Deep RL approaches predict that other mechanisms, such as replay or disentangled representations, may help to alleviate this reliance on interleaved training. If true, then these mechanisms predict that there should be no difference between the high or low perceptual similarity conditions. Either way, the analogy between Deep RL and the brain suggests that there should be no benefit of the high perceptual similarity condition over the low perceptual similarity condition.

One interesting property of our experimental paradigm is that the video games use underlying rules that are verbalisable and therefore involve cardinal dimensions. This is important because the work of Flesch et al. (2018) suggests that verbalisable rules benefit from blocked training when the task involves learning to switch between tasks. If this generalises to the learning of a set of related tasks then it suggests that organising the games so that the perceptual similarity is maximised, as is the case in a blocked training regime, should provide a transfer benefit to participants. Presumably this is because it will help the participants to focus on the cardinal dimensions that constitute the underlying rules.

With these conflicting predictions in mind, we ran three different experiments to investigate the effect of perceptual similarity on transfer performance. Experiment

1 tests the effect of perceptual similarity on transfer in an impoverished match-to-sample task that is more typical of classic psychology experiments. Experiment 2 uses the same stimuli and relational rules as Experiment 1, but uses them in a 2D video game, which represents a more naturalistic setting. Experiment 3 is the same as Experiment 2, but the participants are told the rules of the game before-hand so that we can obtain a measure of ceiling performance given that the participants already know what they need to transfer.

7.2 Experiment 1

For the first experiment we wanted to check: (1) whether participants were actually able to learn the types of relational rules that we planned to use in the video games, and (2) whether the degree of perceptual similarity had an effect in simple static settings. In order to answer these questions, we designed a simple match-to-sample task that used the relational rules we planned to use in the video games. In the task, participants were presented with a central object and they had to choose one of the objects on either the left or the right. The correct choice was based on one of three possible dimensions; colour, shape and texture. Experiment 1 therefore removes many of the more complicated aspects of learning associated with video games and allows participants to concentrate on learning the relational rules for transfer.

7.2.1 Methods

Match-to-Sample Task

The match-to-sample task was implemented using Unity Game Engine to allow for online web hosting. Participants were presented with a simple screen with three basic shapes on it (Figure 7.1). The central object was the reference object and participants needed to use this object to decide whether to select the left or right object. The central object had a colour, texture and shape associated with it and only one of the left or right objects matched on each of these attributes. This ensured that participants could always select an object based on either a colour, texture or shape match.

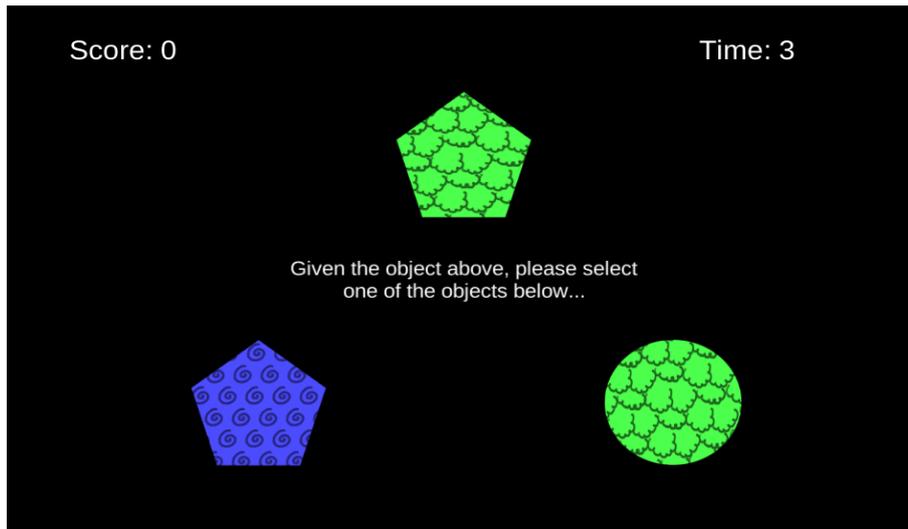


Figure 7.1: *Example screenshot of the match-to-sample task for Experiment 1. Participants had to use the central object to decide whether to click on the left or right object. Depending on the underlying rule of the trial, the correct choice may depend on matching the colour, texture or shape of the central object*

The correct choice between the left and right objects was dictated by a relational rule that corresponded to an attribute of the central object (i.e. its colour, texture or shape). For instance, if the rule was texture based then the correct answer was the object that had the same texture as the central object. These relational rules were chosen because they can potentially be applied to a wide range of different perceptual configurations. Choosing the correct object scored 1 point and participants made a choice by clicking on either the left or right object in the lower half of the screen. For each trial there was a 5 second time limit, after which the participant would score 0 points. If the time limit was reached then participants received no feedback, otherwise they were shown the number of points scored for their choice.

In order to explore the effect of perceptual similarity on transfer, we designed a principled method for generating sequences of questions that represented either high or low perceptual similarity conditions. The shapes and textures of the three objects for the first question were randomly selected from a set of possible training values (Figure 7.2). Consecutive questions were then constructed by randomly selecting an object and randomly changing either its shape, colour or texture. Different sequences of questions were generated by setting the seed of a random generator to 10 different values. Sequences of questions generated in this manner represented the high perceptual similarity condition because consecutive games were highly correlated in

terms of surface features. In comparison, the low perceptual similarity condition was generated by taking the high perceptual similarity sequence of questions and randomly shuffling them. This served to reduce the correlations in perceptual similarity between consecutive questions. Importantly, this methodology ensured that participants were exposed to exactly the same questions over the course of learning regardless of whether they received the high or low perceptual similarity condition. The only thing that differed between a participant trained on the high or low perceptual similarity conditions was the order of the questions. Upon completing the sequence of questions, participants were given a final test question that was the same regardless of whether they had been in the high or low perceptual similarity condition. The final test question used the same relational rule as the training questions but involved a random perceptual configuration that neither group of participants had seen before (Figure 7.2). The final test question therefore provided a fair way of assessing the effect of high vs. low perceptual similarity on transfer ability.

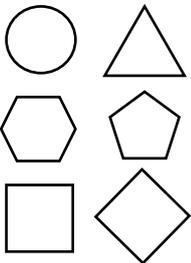
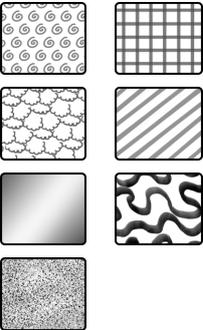
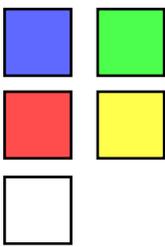
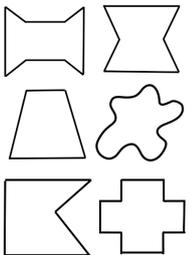
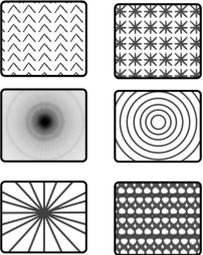
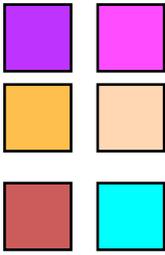
	Shapes	Textures	Colours
Training			
Test			

Figure 7.2: Table showing the shapes, patterns and colours that were sampled from in order to construct the training questions and the novel test questions.

Experimental Procedure

Participants were recruited using the online platform ‘Prolific Academic’ and were rewarded £5 for participation. To enrol on the study participants had to be aged 18-36, have normal or corrected-to-normal vision and be fluent in English. In total 80 participants were recruited (Male=46, Female=34). They were randomly assigned to either the high or low perceptual similarity conditions. Participants completed 5 blocks of 20 trials with the final trial of each block being a novel test trial. For each block a relational rule was chosen at random (repetitions were allowed) to dictate which object the participant should choose in order to score a point. There were three possible relational rules: same texture, same shape and same colour. At the end of each block the participants were warned that the rules of the task had changed.

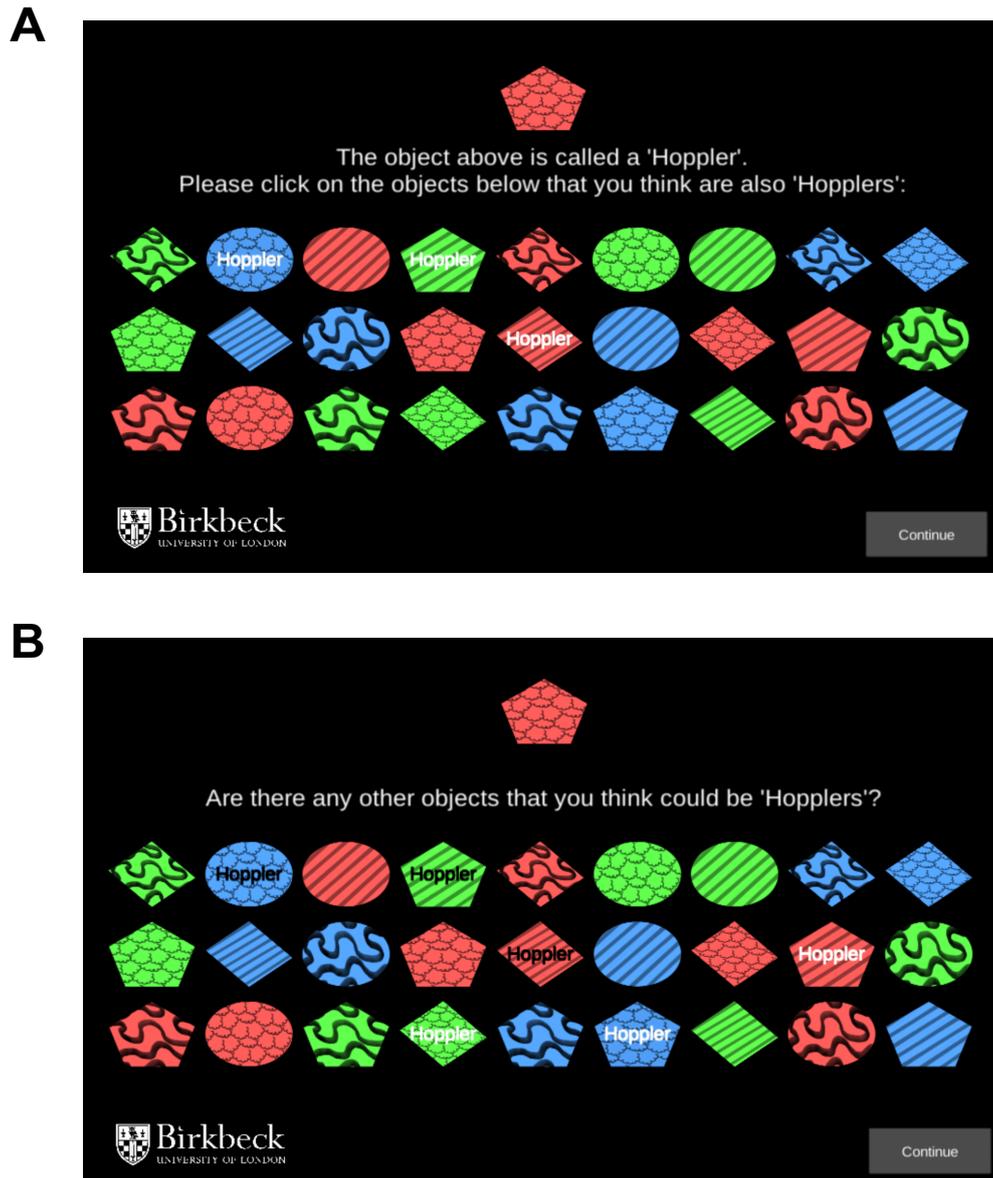


Figure 7.3: Screenshots of the free classification task given to participants before the match-to-sample task. (A) Participants were shown a prototypical object called a ‘hoppler’ and were asked to select other objects they believed to be a hoppler. Participants could select objects by clicking on them at which point they would be labelled as a hoppler. (B) Participants were given a second chance to identify more hopplers, their choices from the first round were locked in place at this point.

Before starting the task participants performed a free classification task (Figure 7.3). The purpose of this task was to evaluate each participants’ prior bias to the three feature dimensions. Each participant was presented with the same example object and were told that the object was a ‘hoppler’. They were also presented with an array of other objects and were asked to select ones that they also believed to be ‘hopplers’. The other objects covered every combination of three different

colours, textures and shapes, with the hoppers colour, texture and shape included in the permutation. After selecting other objects that they believed to be hoppers, participants were asked a second time to select hoppers from the remaining objects. This second round was used to assess how each participant ordered the importance of each of the three features.

Upon completing the free classification task, participants were led through an example match-to-sample trial so that they understood what was required (Figure 7.4). Participants were then given a final set of instructions before beginning the task.

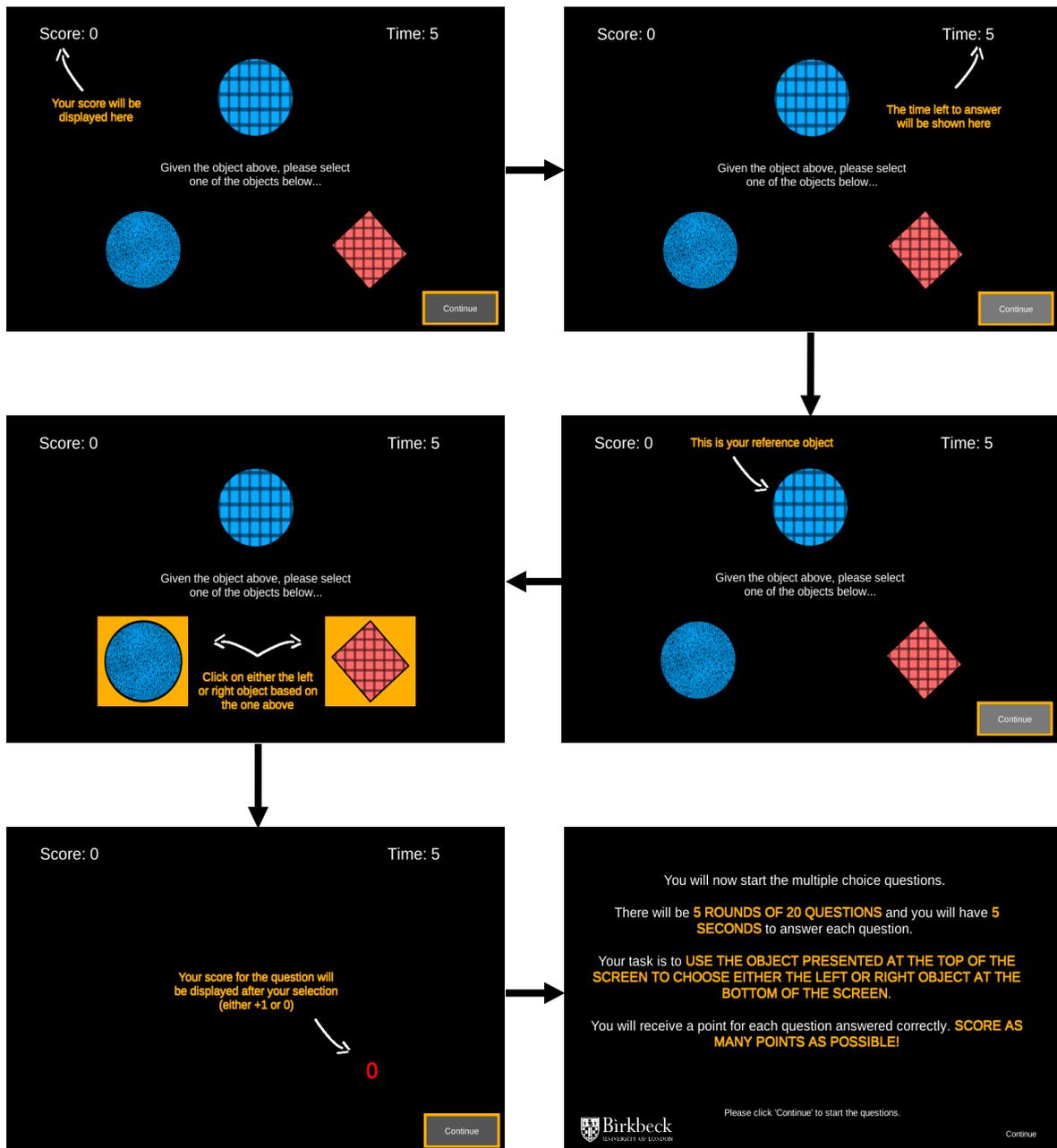


Figure 7.4: Screenshots from the example match-to-sample trial that participants were walked through. The final screenshot shows the instructions given to participants immediately before starting the match-to-sample task. Participants were reminded of design of the experiment as well as the task they had to perform.

7.2.2 Results

Overall Performance

Figure 7.5 shows the performance of the participants over the course of the task collapsed across training conditions. In general participants only required the first question in each block to infer the appropriate rule and subsequently perform above

chance. The final level of each block was completely novel and tested the transfer ability of participants. In every block, participants performed above chance on the final level. This suggests that the participants were able to easily acquire the relational rules from the first trial and use them in novel situations to select the best action.

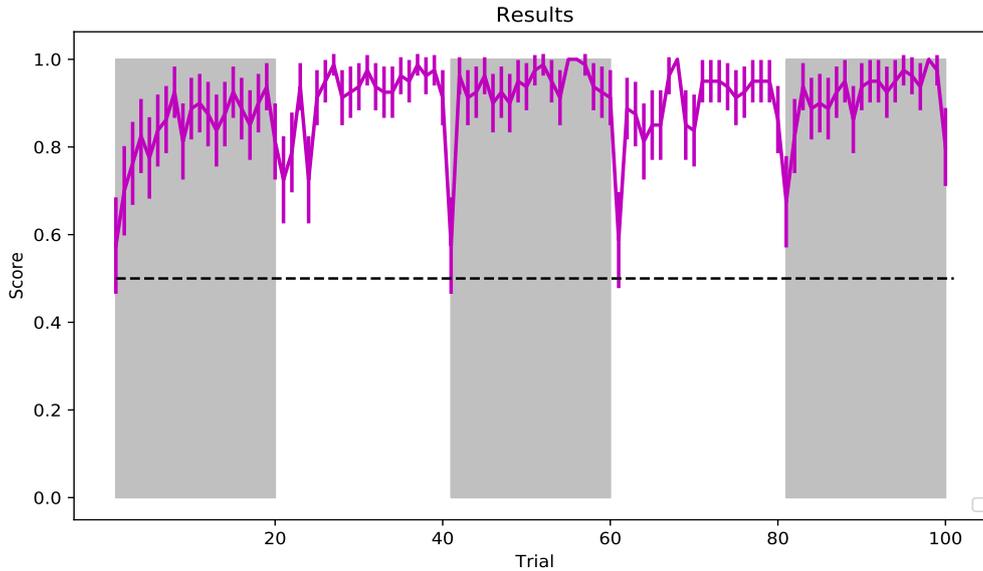


Figure 7.5: Average performance of participants over the course of the task. Dashed line represents chance and shaded regions represent different blocks. Error bars represent 95% confidence intervals.

Figure 7.6 shows the performance of the participants on the last training trial and the test trial for each block of the task. To investigate whether there was a change in performance between the last training trial and the test trial we performed a McNemar test on the change in scores. We found that participants demonstrated a ‘cost of transfer’. More specifically, participants demonstrated a significant drop in performance from the last training trial to the test trial when collapsing across all blocks ($\chi^2(1, N=400)=14.0, p<.001$). At the level of individual blocks, there was a significant drop in performance for the first ($\chi^2(1, N=80)=2.0, p=0.013$) and last ($\chi^2(1, N=80)=1.0, p=0.001$) blocks, but not the middle blocks (block 2: $\chi^2(1, N=80)=1.0, p=0.125$, block 3: $\chi^2(1, N=80)=6.0, p=1.000$, block 4: $\chi^2(1, N=80)=4.0, p=0.118$).

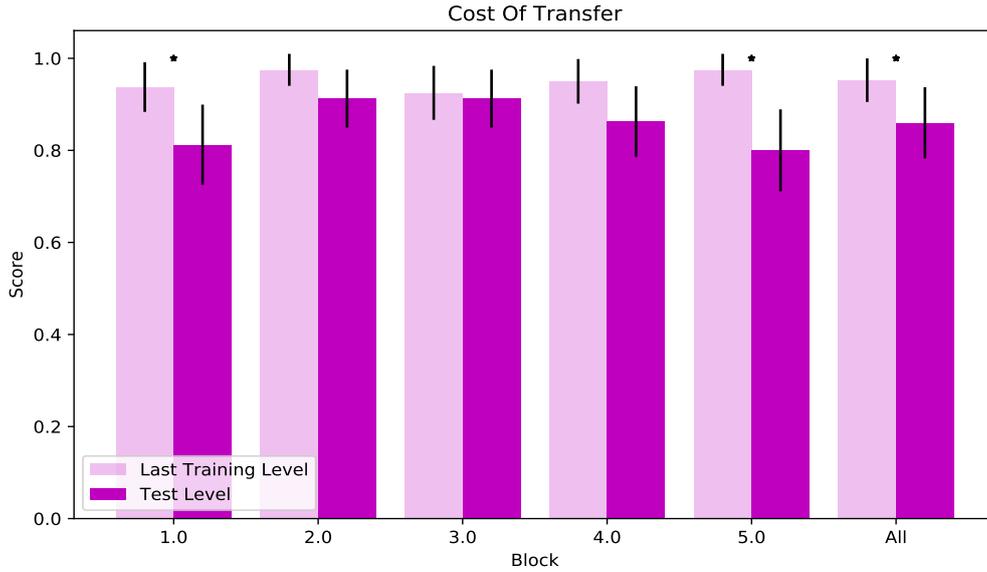


Figure 7.6: *Cost of transfer for each training block as well as overall. Error bars represent 95% confidence intervals and stars represent significant differences.*

Effect of Perceptual Similarity During Training on Test Performance

Figure 7.7 shows the performance of participants over the course of the task but split by whether participants received the low- or high-perceptual similarity training regime. Figure 7.8 shows just the test performance of the participants based on which training regime they received (see Appendix A, Figures A.1 and A.2 for count data). To explore whether there was an effect of training regime on test performance we performed a chi-squared test between the test scores for the two training regimes. We saw no difference between the two regimes with respect to test performance, suggesting that the degree of perceptual similarity between successive trials had no effect on participants' ability to acquire the relational rules and apply them to novel trials (Overall: $\chi^2(1, N=400)=0.2, p=0.640$, block 1: $\chi^2(1, N=80)=0.2, p=0.689$, block 2: $\chi^2(1, N=80)=0.2, p=0.623$, block 3: $\chi^2(1, N=80)=0.2, p=0.623$, block 4: $\chi^2(1, N=80)=0.0, p=0.980$, block 5: $\chi^2(1, N=80)=1.4, p=0.233$).

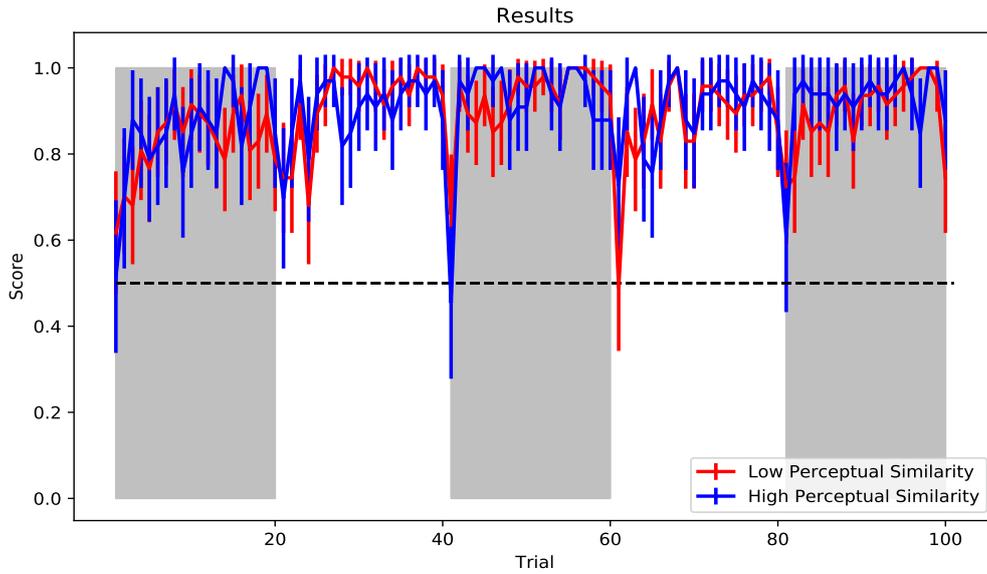


Figure 7.7: Average performance of participants over the course of the task split by perceptual similarity. Dashed line represents chance and shaded regions represent different blocks. Error bars represent 95% confidence intervals.

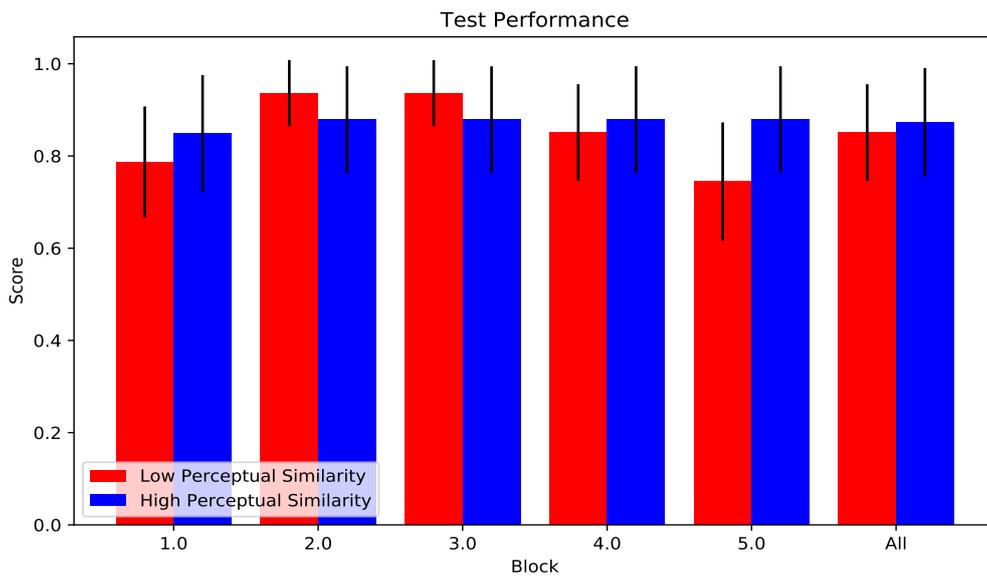


Figure 7.8: Test performance for each training block as well as overall, split by perceptual similarity. Error bars represent 95% confidence intervals and stars represent significant differences.

Effect of Rule Dimension

Figure 7.5 averages across all relational rules and so one possibility is that participants' transfer ability may be different depending on the relational rule. Figure 7.9 shows the performance of participants over the course of the task averaged across

all blocks, split by relational rule. Figure 7.10 shows just the test performance averaged over all blocks for each relational rule (see Appendix A, Figure A.3 for count data). To investigate whether there was a difference in test performance based on the relational rule we performed a chi-squared test between the test scores for each relational rule. We saw no differences in test performance across the different relational rules suggesting that participants could use them in novel trials equally well ($\chi^2(2, N=400)=2.7, p=0.260$).

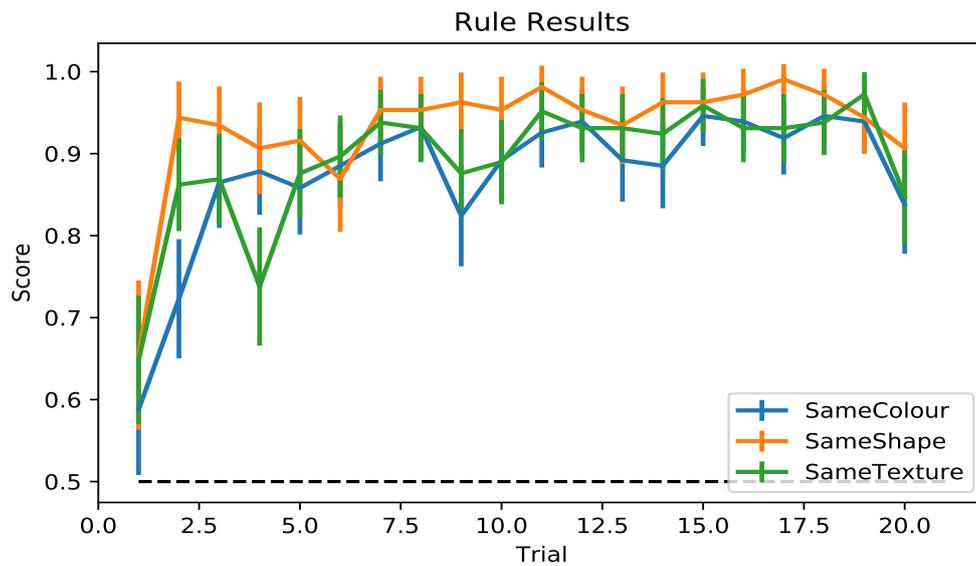


Figure 7.9: Average block performance of participants separated by the relational rule governing the block. Dashed line represents chance and error bars represent 95% confidence intervals.

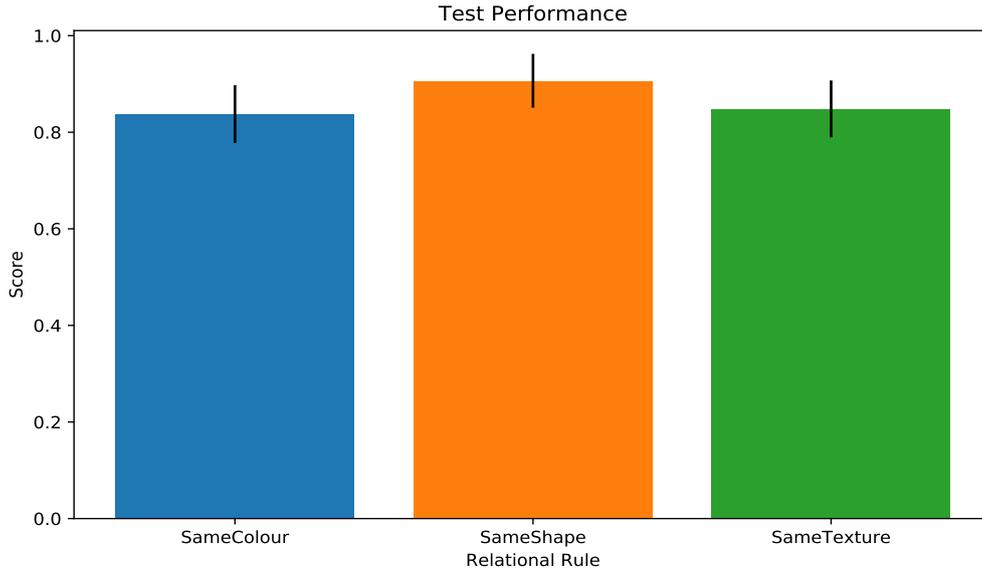


Figure 7.10: *Test performance averaged over all training blocks for each relational rule. Error bars represent 95% confidence intervals and stars represent significant differences.*

Interaction Between Perceptual Similarity During Training and Rule Dimension

The final possibility we explored was whether there was an interaction between the type of relational rule and the degree of perceptual similarity during training. Figure 7.11 shows the performance of participants over the course of the task averaged across all blocks but split by relational rule and perceptual similarity. Figure 7.12 shows just the test performance of participants averaged over all blocks, again split by relational rule and perceptual similarity (see Appendix A, Figure A.4 for count data). To test whether there was an interaction we performed a separate chi-squared test for each relational rule between the test scores of each training regime. We saw no evidence for an interaction between the type of relational rule and the degree of perceptual similarity during training in terms of test performance (Colour: $\chi^2(1, N=148)=0.0$, $p=0.870$, Shape: $\chi^2(1, N=107)=0.5$, $p=0.467$, Texture: $\chi^2(1, N=145)=3.1$, $p=0.078$).

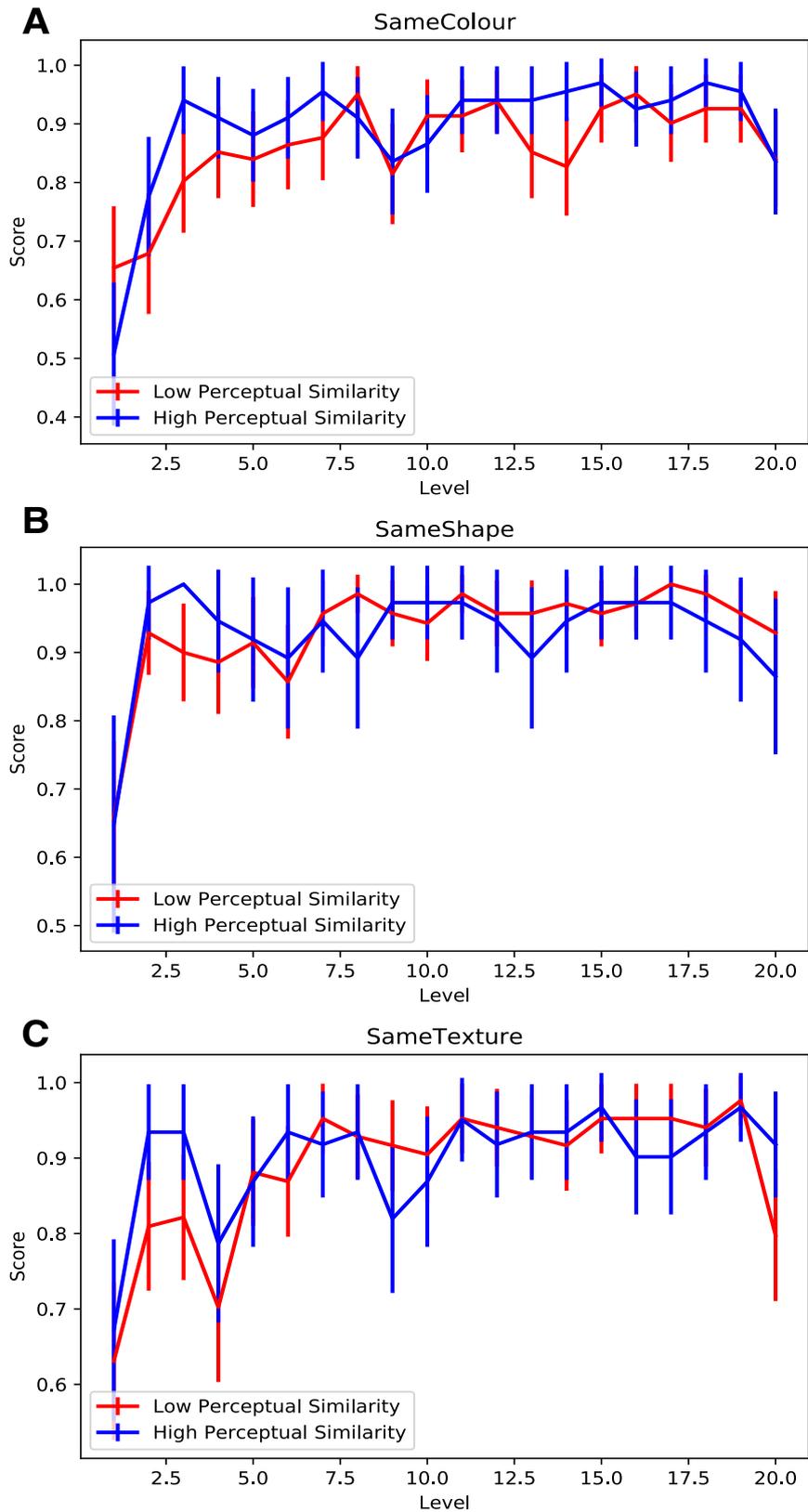


Figure 7.11: Average block performance of participants split by perceptual similarity. Each panel refers to a different relational rule governing the block: (A) Same Colour, (B) Same Shape, (C) Same Texture. Error bars represent 95% confidence intervals.

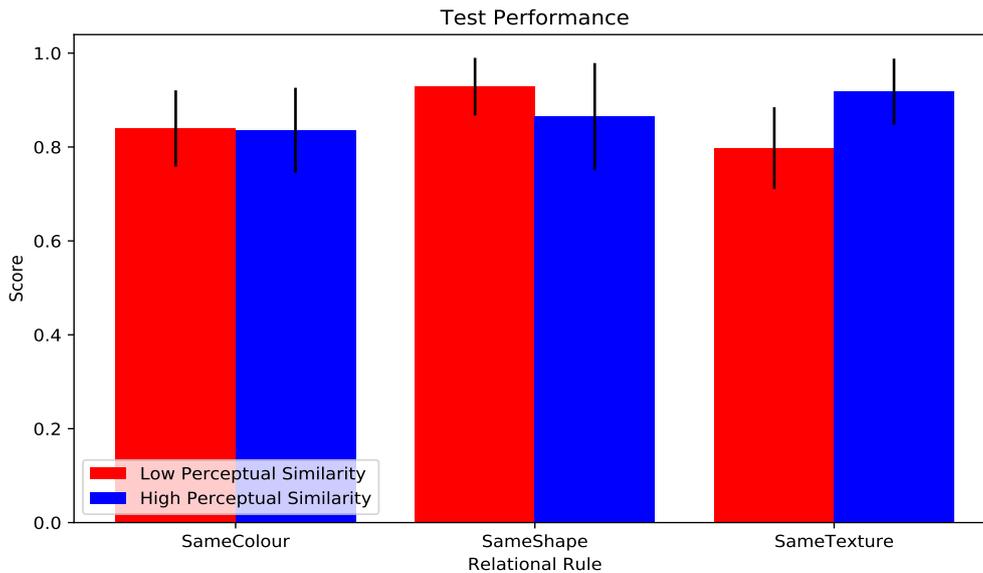


Figure 7.12: Test performance averaged over all training blocks for each relational rule and split by perceptual similarity. Error bars represent 95% confidence intervals and stars represent significant differences

Effect of *a Priori* Attention

Figure 7.13 shows the results of the free-classification task. Participants were most likely to choose objects that had the same shape as the target object (e.g., pentagon), followed by the same texture (e.g., Texture 2) and finally the same colour (e.g., Red). These results suggest that *a priori* participants categorise the stimuli primarily based on shape, followed by texture and finally colour.

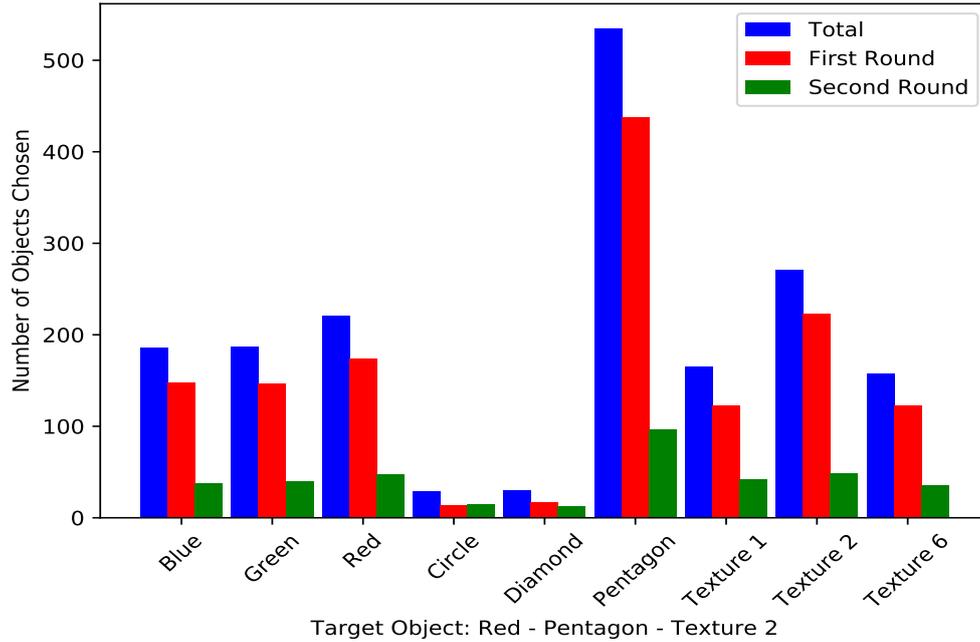


Figure 7.13: Results of the free classification task. The x-axis represents the characteristics of the chosen objects while the y-axis represents the number of objects chosen with a particular characteristic summed over all participants. Blue denotes total counts, red denotes the counts for the first round of classification and green denotes the counts for the second round of classification.

We wanted to investigate whether increased *a priori* attention to a specific dimension corresponded to improved performance on test trials involving that dimension. We therefore performed three separate logistic regressions (Table 7.1). The dependent variables were the test scores (0 or 1) for each rule type (same colour, same shape and same texture) and the independent variables were the training condition and the number of matches across each dimension in the free-classification task. In each of the logistic regressions, none of the dimension matches were significant predictors of test performance. We therefore found no evidence for the hypothesis that increased *a priori* attention to a dimension lead to better transfer of rules involving that dimension at test.

Table 7.1: *Logistic regression analysis of test scores. Each column is a separate logistic regression using only the test scores from blocks using a specific rule. Values not in brackets represent beta coefficients. Values in brackets represent 95% confidence intervals.*

	<i>Dependent variable:</i>		
	Test Score		
	Shape Blocks	Colour Blocks	Texture Blocks
Training Regime	-0.60 (-1.99, 0.79)	0.13 (-0.77, 1.03)	0.88 (-0.20, 1.96)
No. Shape Matches	0.13 (-0.11, 0.36)	0.11 (-0.09, 0.31)	-0.27 (-0.60, 0.06)
No. Colour Matches	-0.37 (-1.12, 0.38)	-0.12 (-0.78, 0.53)	0.68 (-0.43, 1.80)
No. Texture Matches	0.11 (-0.36, 0.57)	0.19 (-0.13, 0.50)	-0.29 (-0.70, 0.11)
Constant	2.39* (0.40, 4.38)	0.60 (-0.67, 1.87)	2.44** (0.61, 4.28)
Observations (N)	1(97), 0(10)	1(124), 0(24)	1(123), 0(22)
Log Likelihood	-32.03	-63.83	-57.46
Akaike Inf. Crit.	74.06	137.66	124.93
Pseudo- R^2	0.80	0.61	0.65
1 - Pearson's χ^2	0.67	0.47	0.07

Note:

* $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$

7.2.3 Discussion

The purpose of Experiment 1 was to explore whether participants could acquire relational rules and use them for transfer in a setting that is typical of behavioural psychology experiments. We found that when presented in the form of a match-to-sample task, participants could indeed rapidly infer the underlying relational rules and use them on trials involving objects they had never seen before. This suggests that both the stimuli and relational rules used in Experiment 1 can support flexible transfer.

These results also demonstrate the ability of participants to rapidly adapt their strategy in the face of changing task demands. Participants were able to quickly react to changes in the underlying rule after only a few trials in a block. One interpretation of this is that participants were able to dynamically switch their attention to different stimulus dimensions (i.e. shape, colour or texture) at the

start of each block. This is exactly the kind of behaviour that Selective Particle Attention (SPA) attempts to capture in Chapter 6. In particular, SPA predicts that participants should select the appropriate dimension at the start of each block by evaluating how accurately they predict reward.

While participants did exhibit transfer, an overall cost of transfer was also observed, particularly in the first and last blocks of the task. The cost of transfer in the first block may be due to the fact that the participants needed time to familiarise themselves with the general requirements of the task. Equally, participants had no prior expectation that they would experience a highly novel test trial on the final level of the block. The cost of transfer in the final block is surprising given the absence of a cost in the middle blocks of the task. One explanation for this may be that participant’s engagement with the task decreased as time went on. However, this seems unlikely as the training score for the final block was as high as in all previous blocks.

Comparison of performance in the low- vs high-perceptual similarity conditions indicated no difference in transfer performance between the two conditions, either when averaged or split across relational rules. One possible explanation for this is that participants were performing at ceiling for the task in both conditions. A harder task may therefore help to uncover the training benefits of one condition over the other. With this in mind, in the next experiment we explore the effect of perceptual similarity on transfer in a more naturalistic setting.

7.3 Experiment 2

Having demonstrated in Experiment 1 that participants are able to acquire and use the relational rules for transfer, we designed a more complex and naturalistic video game that incorporated the same relational rules. Participants in Experiment 1 performed close to ceiling regardless of whether the training consisted of high- or low-perceptual similarity. This suggests that the task was too easy and participants could transfer the relational rules irrespective of training regime. We therefore took the stimuli from Experiment 1 and used them in a simple 2D video game. We hypothesised that the increased difficulty of the games, compared to the match-to-

sample task, would remove the ceiling effect and reveal any differences between the low- and high-perceptual similarity conditions.

7.3.1 Methods

Games

All of the tasks in Experiment 2 were conducted using custom-made 2D video games implemented in Unity game engine. Each game consisted of a simple 2D environment with moving 2D objects. Participants were able to control one of the 2D objects using the arrow keys on a keyboard. The arrow key direction corresponded to the direction of movement of the player object and this allowed for 9 possible movement combinations (left, right, up, down, left-up, right-up, left-down, right-down, no input). Movement of the player object was additive using velocity vectors. For example, if the player object was moving directly across the screen from left to right then a press/selection of the left arrow would add a small leftward vector to the player object's current rightward movement vector. This would cause the player's object to move to the right at a slower rate. Subsequent presses/selections of the left arrow key would keep adding a leftward vector until the object started to move leftward. Importantly, choosing to press no arrow key (no input) does not stop the player object but leaves its current velocity vector unchanged and so the player object will continue on its current trajectory until an arrow key is pressed. Participants were told to use the arrow keys for the study but were not told about the correspondence between the arrow key direction on the keyboard and their effect on the 2D screen.

As with Experiment 1 we wanted the video games to utilise relational rules that allowed for different perceptual configurations while maintaining the same task structure. With this in mind, the video game consisted of four objects; the player controlled object, two moving objects with constant velocity and a static goal object. The goal of the game was to score as many points as possible on a single trial. In order to score a point the player had to collide with the static goal object. The goal object was randomly positioned along the edge of the screen and would disappear after the player-controlled object collided with it. To make the goal object re-appear for collection, the player-controlled object had to collide with the moving object that

was the same texture as itself. Conversely, if the player-controlled object collided with the moving object that was a different texture then the game would end and a new trial would start after a time-out period. The key characteristic of these rules is that they are flexible and abstract enough to allow for games of varying perceptual similarity while ensuring the rules remain constant. One can vary the shapes, colours and textures of the objects so long as one of the moving objects has the same texture as the player object. These rules represent a substrate for transfer because they provide high-level information about how to act optimally in the 2D world, regardless of the current perceptual instance of the game. A summary of the fundamental rules governing the video games can be seen in Figures 7.14 and 7.15, and a screenshot from an example game can be seen in Figure 7.16. We chose to use the texture rule over the shape and colour rules from Experiment 1 because the texture rule appeared to show the clearest trend towards a difference between high and low perceptual similarity training (Figure 7.11).

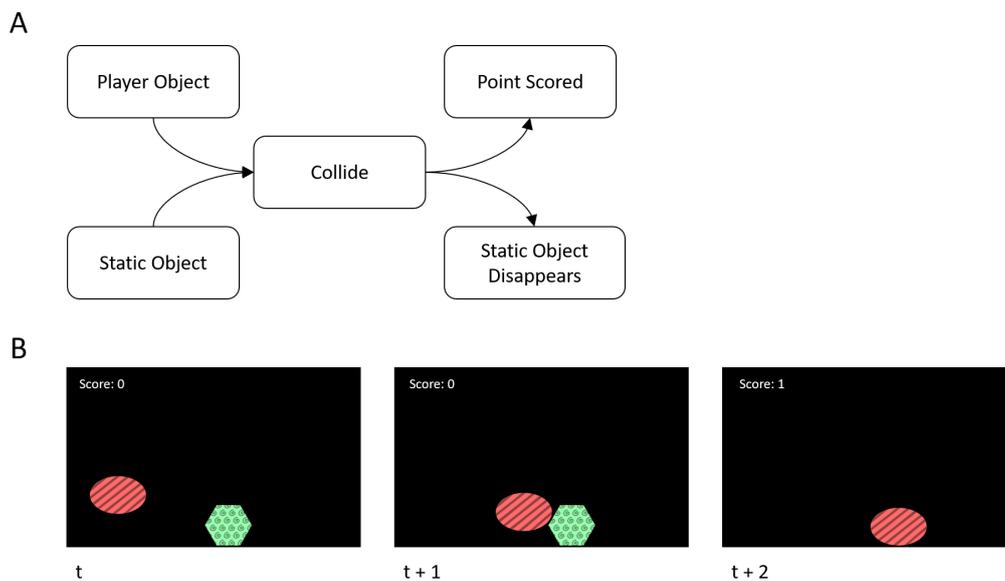


Figure 7.14: **(A)** Flow chart describing the rule that governs the scoring of points. **(B)** Diagram of how the rule looks while playing the game. If the player collides with the static object then the player scores a point. The player object is the red circle and the static object is the green hexagon.

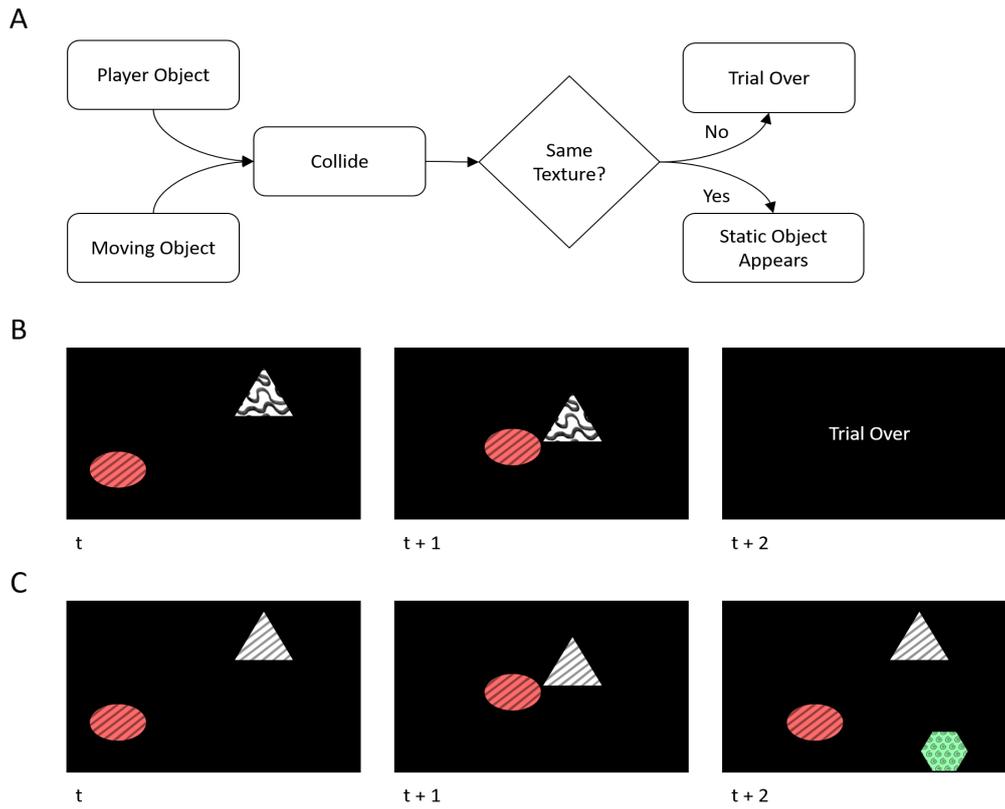


Figure 7.15: **(A)** Flow chart describing the rule that governs the ending of a trial and the generation of a static object. **(B)** If the player collides with a moving object and they are different textures then the trial ends. **(C)** If the player collides with a moving object and they are the same texture then a static object is generated. The player object is the red circle, the moving object is the white triangle and the static object is the green hexagon.

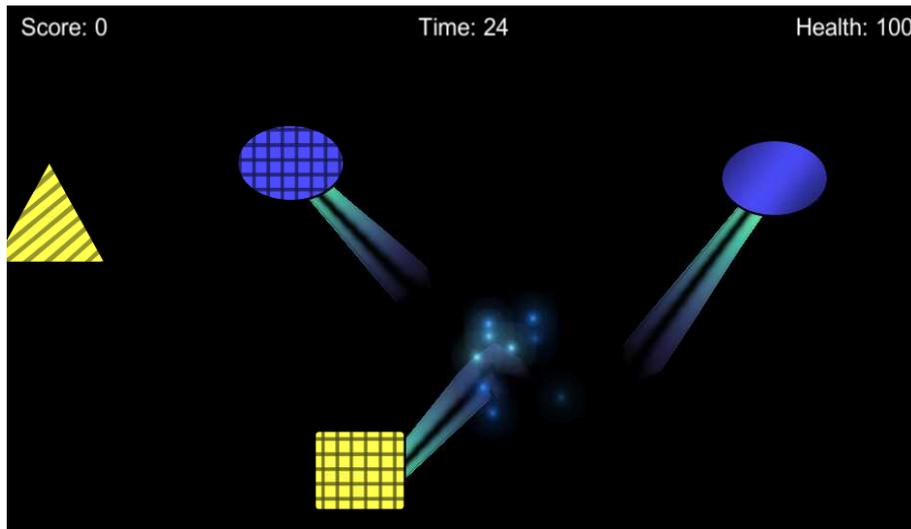


Figure 7.16: *Example screenshot of the video games for Experiment 2. Score, time remaining on the level, and health were all shown at the top of the screen. The player controlled one of the moving objects. Moving objects left a short trail indicating where they had recently been.*

The high and low perceptual similarity conditions were generated in the same way as in Experiment 1 using 10 different random seed values. The high similarity condition was generated by randomly changing one feature of a randomly chosen object between successive games. The only constraint on this was that the player texture always had to match one of the moving objects' textures. This ensured that all game rules were present for each game and consecutive games could differ by no more than two features (e.g. if the player texture was changed then a moving object texture was also changed to ensure all rules remained present). Again the low perceptual similarity condition was generated by randomly shuffling the sequence of games associated with the high perceptual similarity condition. In addition to the high and low perceptual similarity conditions, we also included a third condition for Experiment 2. In this condition all the object features were randomised between consecutive levels, while ensuring all the rules were satisfied. As a result, this condition does not control for the same training levels as the other two conditions. The purpose of this condition was to test whether some of the proposed benefits of perceptually dissimilar experiences are due to the increased variety of experiences rather than the actual temporal sequence. We refer to this third condition as the *random condition*. As before, all conditions finished with a final test level, which used perceptually novel features to obtain a true measure of transfer ability.

Figure 7.17 contains all the shapes, textures and colours used for generating the game trajectories. Example trajectories for each of the conditions can be seen in Figure 7.18. All participants completed 20 games in total with the first 19 being specific to the training condition and the last one being the novel test level (see Appendix C for the results of a shortened version with only 10 games). The duration of the 19 training games was 30 seconds and the duration of the test game was 120 seconds. The rationale for a short training game duration was to increase participant engagement by shortening the overall experiment duration. The rationale for a longer test game duration was to allow for greater differences in test performance.

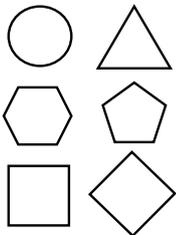
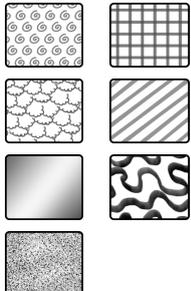
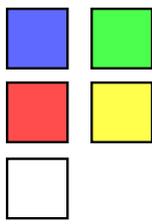
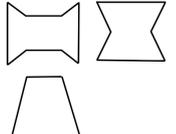
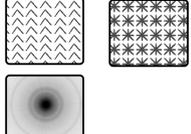
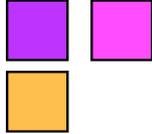
	Shapes	Textures	Colours
Training			
Test			

Figure 7.17: Table showing the shapes, textures and colours that were sampled from in order to construct the training levels and the novel test level.

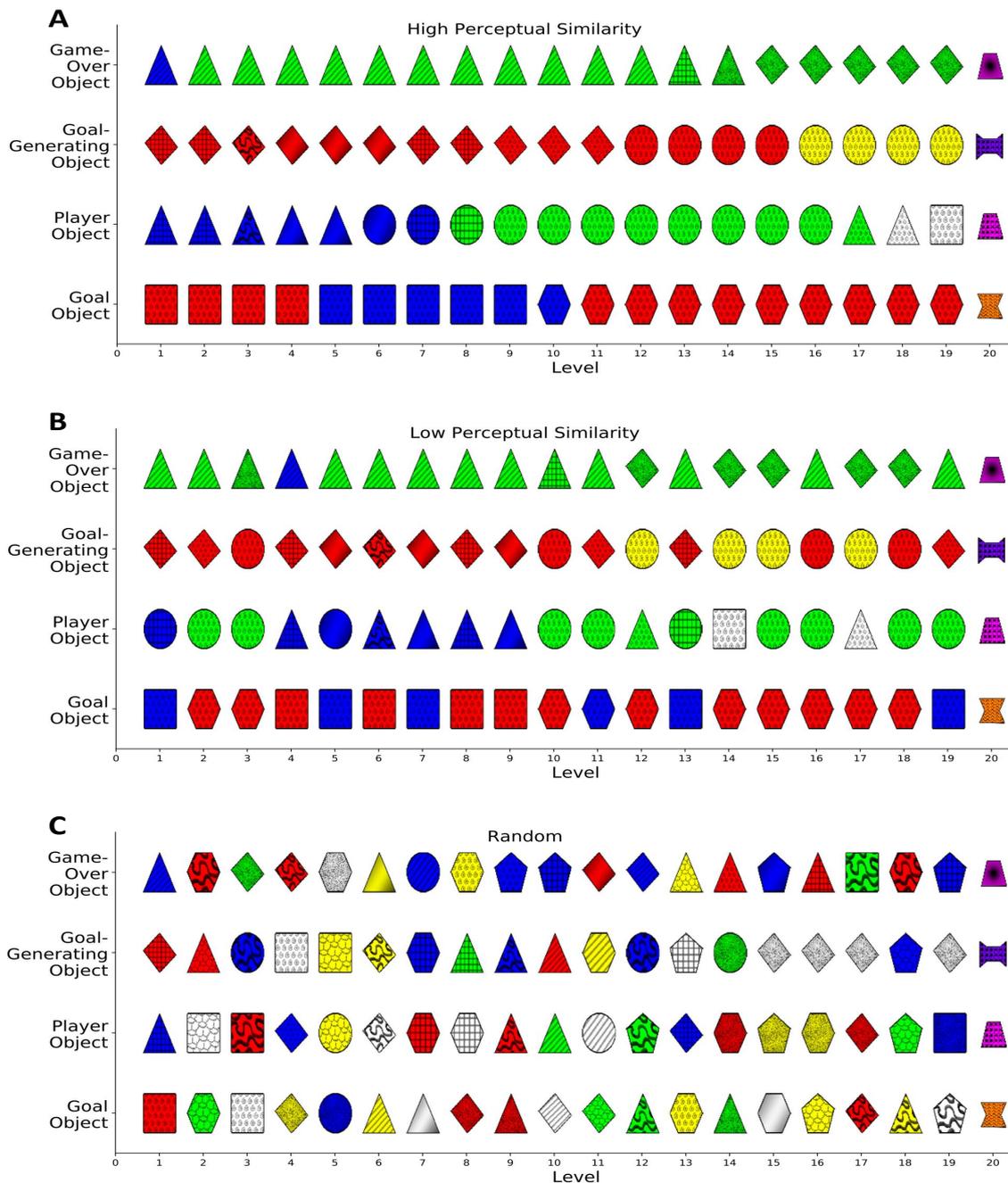


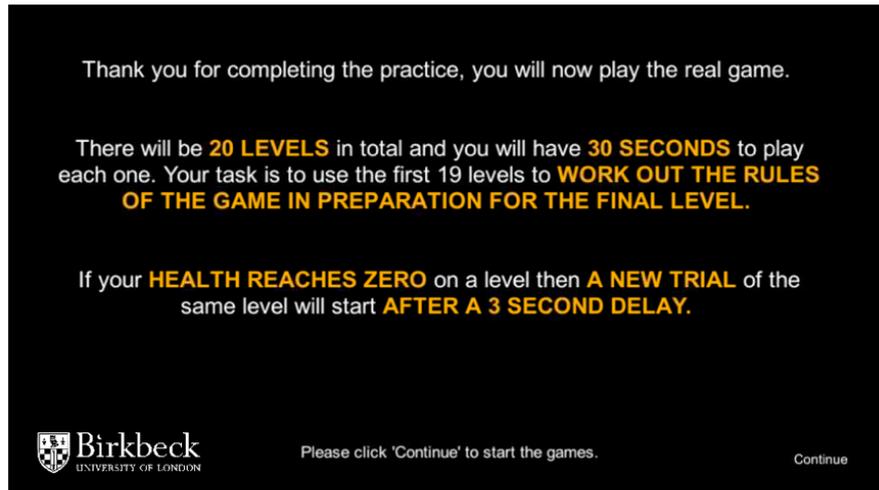
Figure 7.18: *Depiction of the different training conditions. The y-axis represents the object and the x-axis represents the level number. The final level represents the perceptually novel test level. (A) High-perceptual similarity condition. On any two consecutive levels only the shape, texture or colour of one object was changed. If the player object's or goal-generating object's texture changed then both needed to be changed to ensure that there was always a texture match. (B) Low-perceptual similarity condition. The high-perceptual similarity condition was randomly shuffled to decrease perceptual similarity but control for the games experienced. (C) Random condition. All object features were randomised between games to minimise perceptual similarity and provide additional variance compared to the other two conditions.*

Experimental Procedure

Three hundred participants were recruited (Male=182, Female=117, Undisclosed=1) using the online platform ‘Prolific Academic’ and were rewarded £5 for participation. To enrol on the study participants had to be aged 18-30, be fluent in English and be using a desktop computer. Participants were split into three groups; one receiving a high-degree of perceptual similarity between consecutive levels, one receiving a low-degree of perceptual similarity between consecutive levels, and one receiving completely random levels. Participants were removed from the analysis if the number of key presses they made for any game was 0. This led to 95 participants in the high-degree of perceptual similarity group, 100 participants in the low-degree of perceptual similarity group and 99 participants in the random group.

Before starting the games participants were given the same free-classification task as in Experiment 1. Participants were also told that the games were related and that they would need to use the training games in order to score as many points as possible on the final test game. The rationale for this was that spontaneous transfer can be difficult (Kurtz and Honke, 2020) and so we wanted to remove the added difficulty of identifying that the games were related. Example screenshots of the instructions given to the participants can be seen in Figure 7.19.

A



B

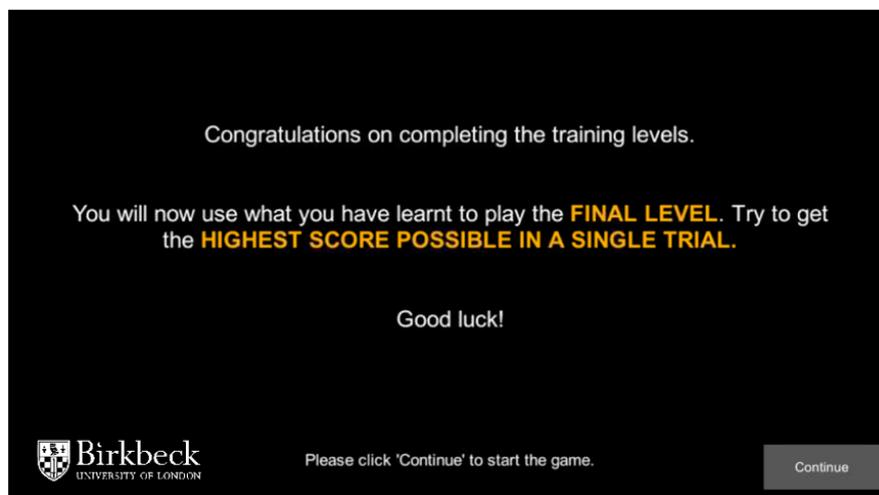


Figure 7.19: (A) Example screenshot of the instructions given to participants before playing the training games. (B) Example screenshot of the instructions given to participants before playing the test game. Participants were made aware of the fact that the games were related and that the first 19 were informative with respect to the final game

Participants were also given a task before and after the experiment to control for any improvement in using the keyboard controls. The rationale for this was that part of the improvement in performance over the course of the experiment may be due to participants becoming more familiar with using the arrow keys to control the player object. By accounting for this improvement, we hope that any performance differences observed can be attributed to the transfer of knowledge rather than to motor control improvements. A screenshot of the motor control task given to participants can be seen in Figure 7.20. The task involved controlling a white circle in order to collect a series of blue rectangles. The controls for the white

circle worked in the same way as in the main games. The time taken to collect all the blue rectangles was taken as a proxy for the participant's competence in using the arrow keys to control the player object.

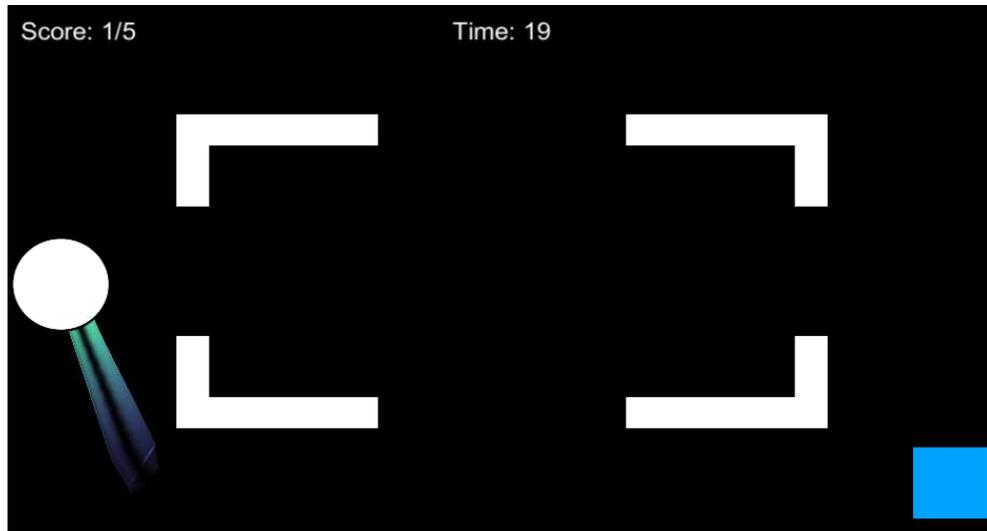


Figure 7.20: *Example screenshot of the motor control task given to participants both before and after the main games. The goal of the task was to use the white circle to collect 5 blue squares. The blue squares appeared sequentially so that once one square was collected the next one would appear. Both the number of squares collected and the time elapsed were shown at the top of the screen.*

7.3.2 Results

Free-Classification Task

The results of the free-classification task can be seen in Figure 7.21. We observed the same pattern as in Experiment 1, with participants favouring matches based on shape, followed by texture and finally by colour.

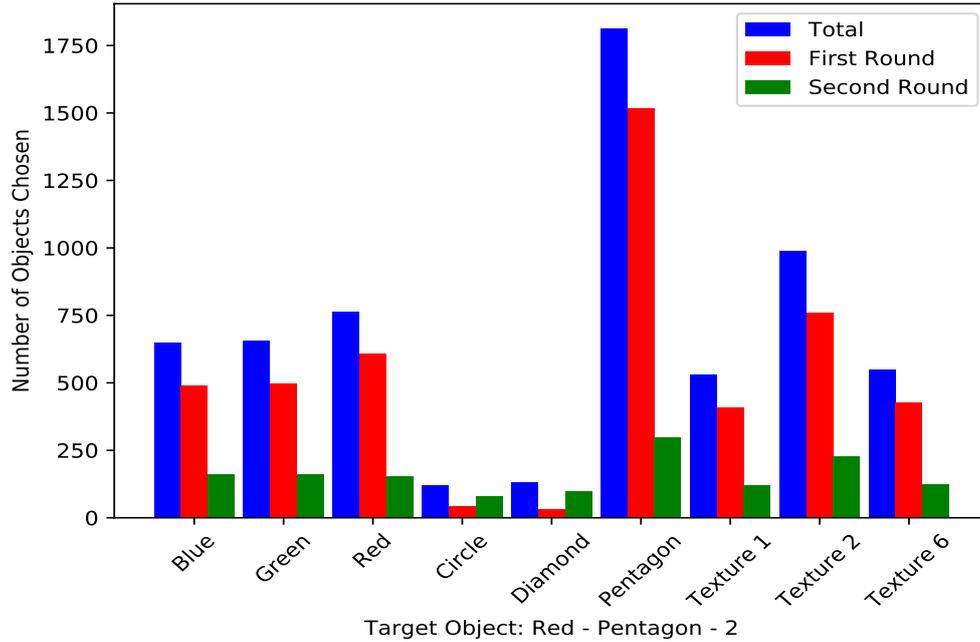


Figure 7.21: Results of the free classification task. The x-axis represents the characteristics of the chosen objects while the y-axis represents the number of objects chosen with a particular characteristic summed over all participants. Blue denotes total counts, red denotes the counts for the first round of classification and green denotes the counts for the second round of classification.

Training Performance

For our analysis of the games we chose to focus on the score achieved on the first trial of each level and the total score achieved on each level. The score on the first trial of a level is an indication of a participant’s ability to correctly infer which object to avoid and which one to interact with. The score on the first trial therefore provides a measure of a participant’s ability to transfer knowledge without feedback. In comparison, the total score on a level incorporates a participant’s ability to transfer knowledge with the use of feedback. For example, participants can use the results of the first trial to infer which object to avoid and which one to interact with, a strategy that may lead to a poor first score but a good total score.

Figure 7.22 shows the training trajectories for the three training regimes. All regimes showed a rapid increase in performance during the early levels, with performance plateauing off towards the end of training. Figure 7.23 shows the first and total scores summed over all the training levels for each training regime (see Appendix B, Figures B.1 and B.2 for histograms). The results of a Kruskal-Wallis rank sum

test indicated that there was a significant difference between the regimes in terms of the sum of first scores over training ($\chi^2(2, N=294)=12.5, p=0.002$) but not the sum of total scores over training ($\chi^2(2, N=294)=3.6, p=0.165$). Pairwise Wilcoxon rank sum tests with bonferroni corrections revealed that in the case of the sum of first scores over training, the random condition was significantly lower than both the high-perceptual similarity condition ($p=0.010$) and the low-perceptual similarity condition ($p=0.005$). There was no significant difference between the high-perceptual similarity condition and the low-perceptual similarity condition ($p=1.000$). In the case of the sum of total scores over training there were no significant differences (low vs. high: $p=1.00$, low vs. random: $p=0.360$, high vs. random: $p=0.250$).

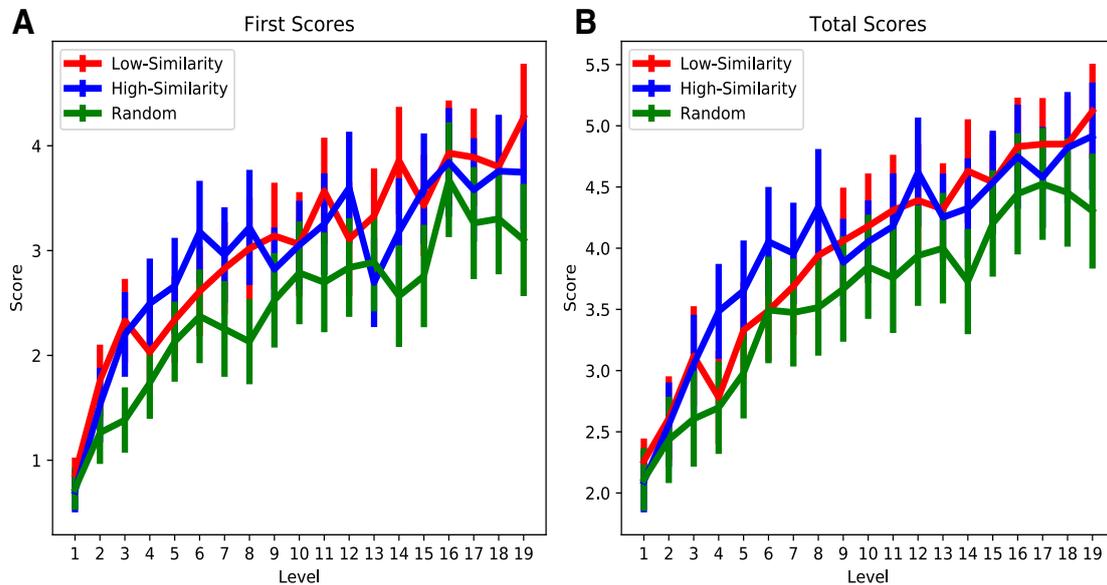


Figure 7.22: Scores for each level over the whole training trajectory for high-perceptual similarity, low-perceptual similarity and random conditions. **(A)** First trial scores **(B)** Total scores. Error bars represent 95% confidence intervals.

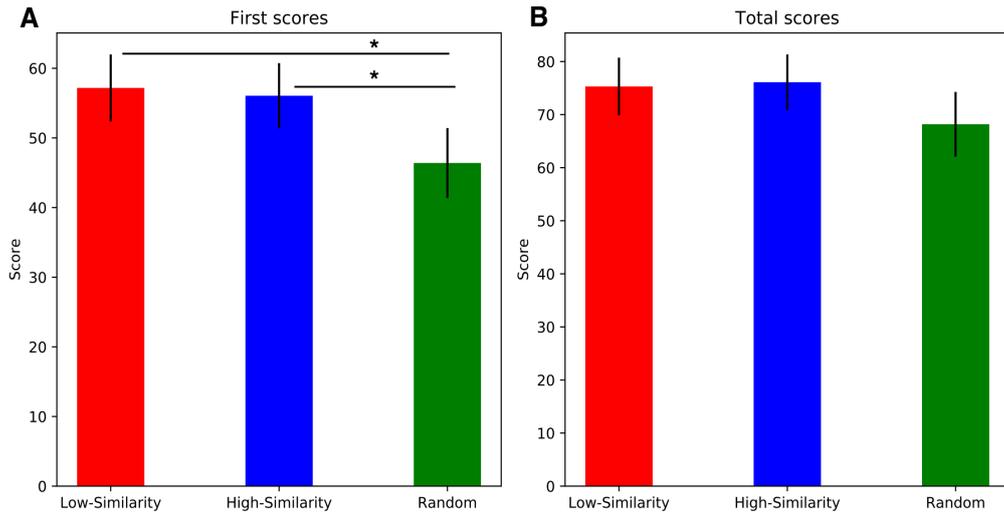


Figure 7.23: Scores summed over all training levels for high-perceptual similarity, low-perceptual similarity and random conditions. (A) First trial scores. (B) Total scores. Error bars represent 95% confidence intervals.

Test Performance

As in Experiment 1, the key measure of overall transfer performance was the score on the final test level. Figure 7.24 shows the first and total scores for the final test level for all training regimes (see Appendix B, Figures B.3 and B.4 for histograms). To test for an effect of perceptual similarity during training on test scores we performed a Kruskal-Wallis rank sum test between the test scores for each of the training regimes. For both the first ($\chi^2(2, N=294)=1.7, p=0.426$) and total ($\chi^2(2, N=294)=2.2, p=0.336$) test scores we found no significant differences. This lack of differences between the different training regimes suggests that the degree of perceptual similarity between consecutive levels did not have an effect on transfer ability.

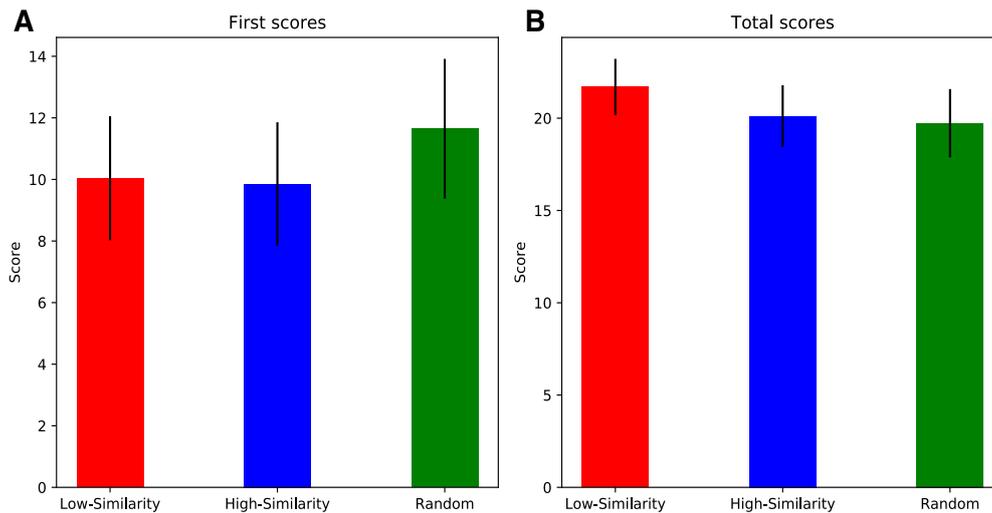


Figure 7.24: Scores on the final test game for high-perceptual similarity, low-perceptual similarity and random conditions. **(A)** Score achieved on the first trial of the final game **(B)** Total score achieved on the final game. Error bars represent 95% confidence intervals.

In order to further explore which variables contributed to performance on the final test game and therefore transfer ability we also ran a regression analysis (Table 7.2). The first scores and total scores on the final test game were used as dependent variables in separate regression models, and the training regime, the self-reported video game experience, the number of texture matches during the free classification task, the performance on the first motor control task and the difference in performance between the first and second motor control tasks were included as independent variables. The self reported video game experience was included as an independent variable in order to control for participants' differing levels of video game experience. The number of texture matches in the free classification task were included to test the hypothesis that *a priori* attention to texture would improve the participants ability to identify the rule and transfer it to novel levels. The performance on the first motor control task and the difference between the two motor control tasks were included as independent variables to control for participants' competence with using the controls and any improvements they made in using the controls over the course of the experiment.

Table 7.2 shows the results of these regression analyses. Across both regression models neither the training regime nor the number of texture matches were significant predictors of performance on the final test game. In comparison, the time

taken on the first motor task and the difference between the second and first motor tasks were significant predictors of test performance. More specifically, faster times on the first motor task led to better transfer performance. Similarly, the larger the improvement between the first and second motor tasks the better the transfer performance. This suggests that the participants' familiarity with the controls and their subsequent improvement in using these controls is a good predictor of transfer performance on the final test game.

Table 7.2: *Regression analysis of test scores. Each column is a separate regression using either the first or total score on the test level. Values not in brackets represent beta coefficients. Values in brackets represent 95% confidence intervals.*

	<i>Dependent variable:</i>	
	First Score (1)	Total Score (2)
Experience	0.47* (0.01, 0.93)	0.21 (-0.16, 0.57)
High-Perceptual Similarity	-0.06 (-2.96, 2.83)	-1.75 (-4.04, 0.54)
Random	1.51 (-1.33, 4.35)	-2.16 (-4.42, 0.09)
Motor Task 1	-1.17*** (-1.86, -0.48)	-1.13*** (-1.68, -0.58)
Motor Task 2 - Motor Task 1	-1.12** (-1.83, -0.42)	-1.01*** (-1.57, -0.45)
No. Texture Matches	-0.04 (-0.66, 0.58)	0.17 (-0.33, 0.66)
Constant	28.64*** (13.99, 43.30)	41.01*** (29.38, 52.63)
Observations	294	294
R ²	0.08	0.11
Adjusted R ²	0.07	0.09
Residual Std. Error (df = 287)	10.18	8.08
F Statistic (df = 6; 287)	4.40***	5.74***

Note: *p<0.05; **p<0.01; ***p<0.001

Cost of Transfer

Aside from transfer performance, we also wanted to check whether there was a difference in the cost of transfer between training regimes (Figure 7.25). In order to explore this, we tested whether the change in performance between the last training

level and the test level was significantly different between the training regimes using a Kruskal-Wallis rank sum test. For both the first ($\chi^2(2, N=294)=5.4, p=0.069$) and total ($\chi^2(2, N=294)=1.7, p=0.432$) test scores we found no significant difference between the training regimes with respect to the cost of transfer.

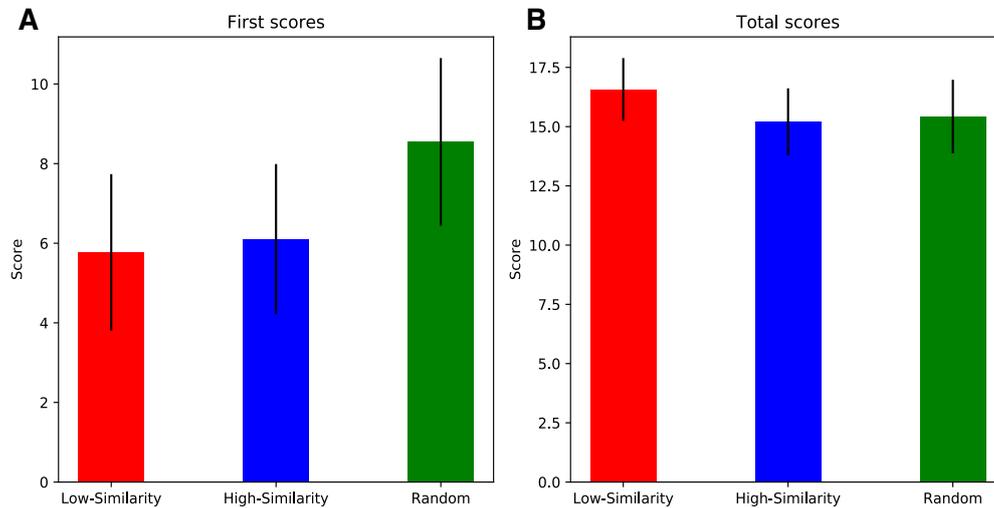


Figure 7.25: *Change in scores from the last training level to the final test level for the high-perceptual similarity, low-perceptual similarity and random conditions. (A) Change in first scores. (B) Change in total scores. Error bars represent 95% confidence intervals.*

Object Interactions

To further understand what the participants were learning over the course of the task we looked at which objects the participants interacted with over the course of the experiment. Figure 7.26 shows the proportion of participants that interacted with a particular object for each level. We analysed all object interactions using chi-squared tests. Interactions with the goal objects were excluded in order to focus on participants' ability to infer the underlying texture rule, which is dependent on interactions with the moving objects.

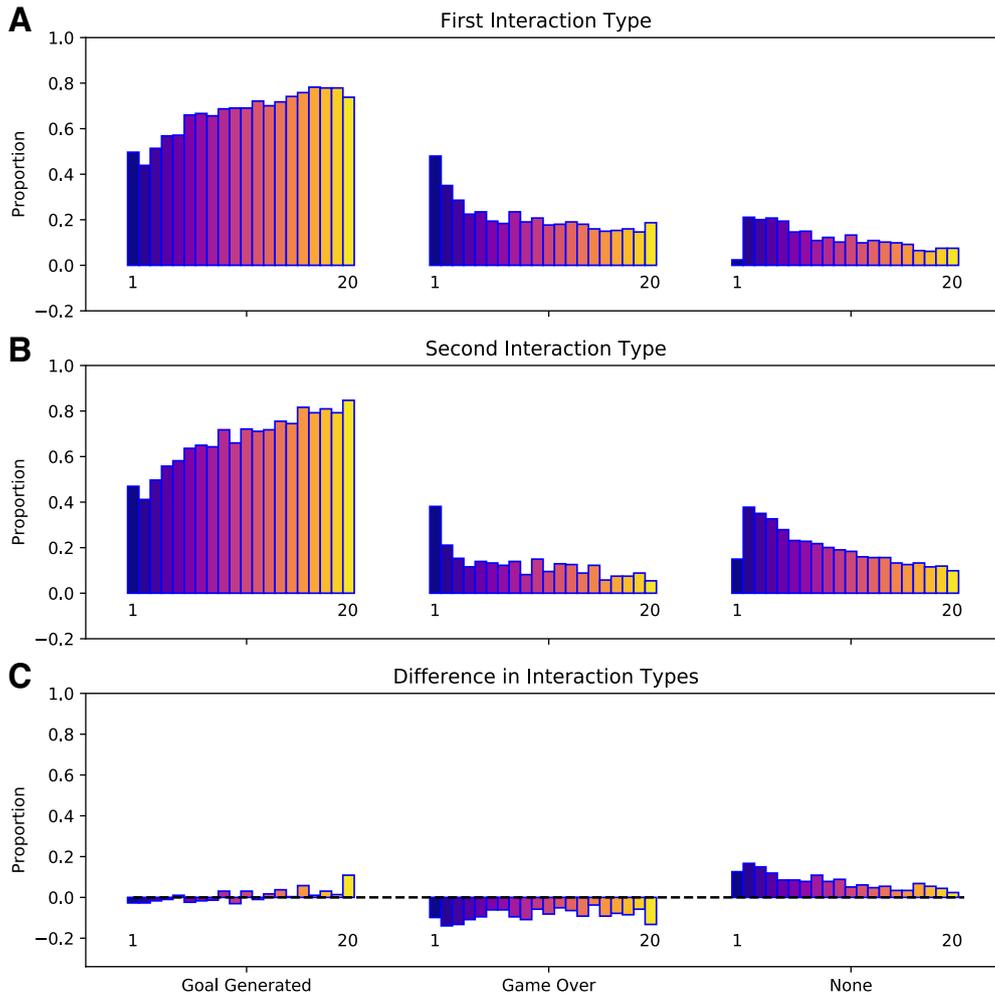


Figure 7.26: *The proportion of interaction types during learning. (A) The proportion of first interaction types for all participants. An interaction is defined as a collision with another object. The x-axis indicates the object that was first interacted with (excluding the goal object) and the colour of the bars indicates the level number (blue is first training level and yellow is final test level). (B) Same as A but for second interaction types. (C) The difference between the first and second interaction types.*

If participants successfully discovered the underlying relational rule and transferred it between levels then their first interaction should be with the goal-generating object. Over the course of training, the number of first interactions with the goal-generating object significantly increased when comparing the first training level to the final test level ($\chi^2(1, N=588)=35.3, p<0.001$). In comparison, the number of first interactions with the game-over object decreased from the first training level to the final test level ($\chi^2(1, N=588)=55.3, p<0.001$). This suggests that participants were indeed learning which object to interact with first; i.e. that of the same tex-

ture as the player object. Further analyses of the final test level showed that the number of goal-generating first interactions was significantly larger than the number of game-ending first interactions ($\chi^2(1, N=588)=177.3.0, p<0.001$). This suggests that the majority of participants were able to use the underlying texture rule to infer which object to first interact with on the final test level.

Interestingly, the difference between goal-generating and game-ending interactions was even more exaggerated for the second interaction of each level. This effect was mainly driven by an increased aversion to the game-over object on the second interaction. In particular, on the final test level the number of goal-generating second interactions was significantly larger than the number of goal-generating first interactions ($\chi^2(1, N=588)=9.9, p=0.002$). Similarly the number of game-over second interactions was significantly lower than the number of game-over first interactions ($\chi^2(1, N=588)=23.1, p<0.001$). This suggests that some participants may be employing a ‘one-shot’ strategy, whereby they use the first interaction to infer which subsequent objects to interact with. Such a strategy would not require any understanding of the underlying relational texture rule, only that one object is always goal-generating and one object is always game-ending.

Figure 7.27 also demonstrates the proportion of different object interactions but split based on training regime. Interestingly, from visual inspection it appeared that in the random condition participants were more likely to first interact with the game-ending object and then interact with no other objects. This suggests that the random condition potentially reduced exploratory behaviour during training and made it harder for the participants to discover that they could generate more goal objects.

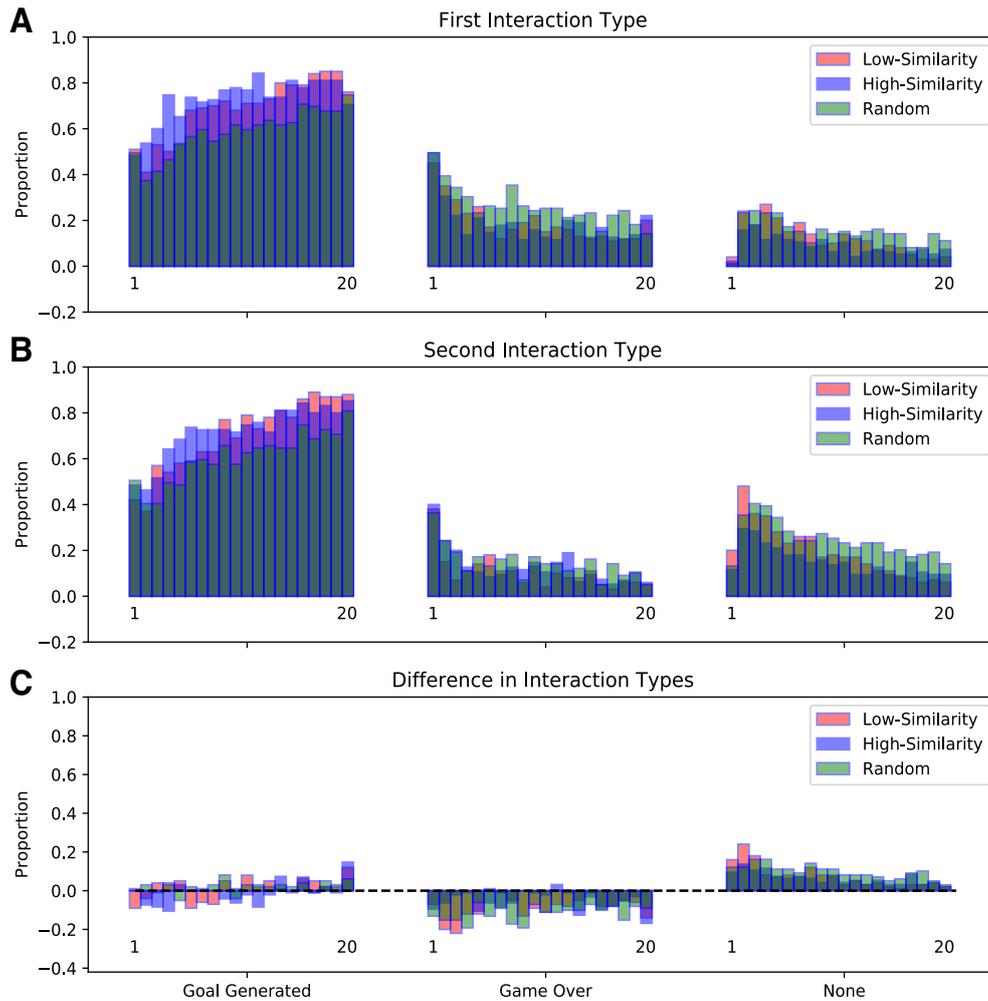


Figure 7.27: *The proportion of interaction types during learning split by training regime. (A) The proportion of first interaction types for all participants. An interaction is defined as a collision with another object. The x-axis indicates the object that was first interacted with (excluding the goal object) for each level of the experiment (1-20). The colour represents the training regime. (B) Same as A but for second interaction types. (C) The difference between the first and second interaction types.*

Figure 7.28 shows the proportion of interaction types on just the test level for the different training regimes. The results of a chi-squared test revealed no significant difference between training regimes for both the first ($\chi^2(4, N=294)=5.4, p=0.252$) and second ($\chi^2(4, N=294)=3.8, p=0.439$) interaction types. This is consistent with the lack of significant differences in test score between the different training regimes and further suggests that the training regime had no impact on transfer performance.

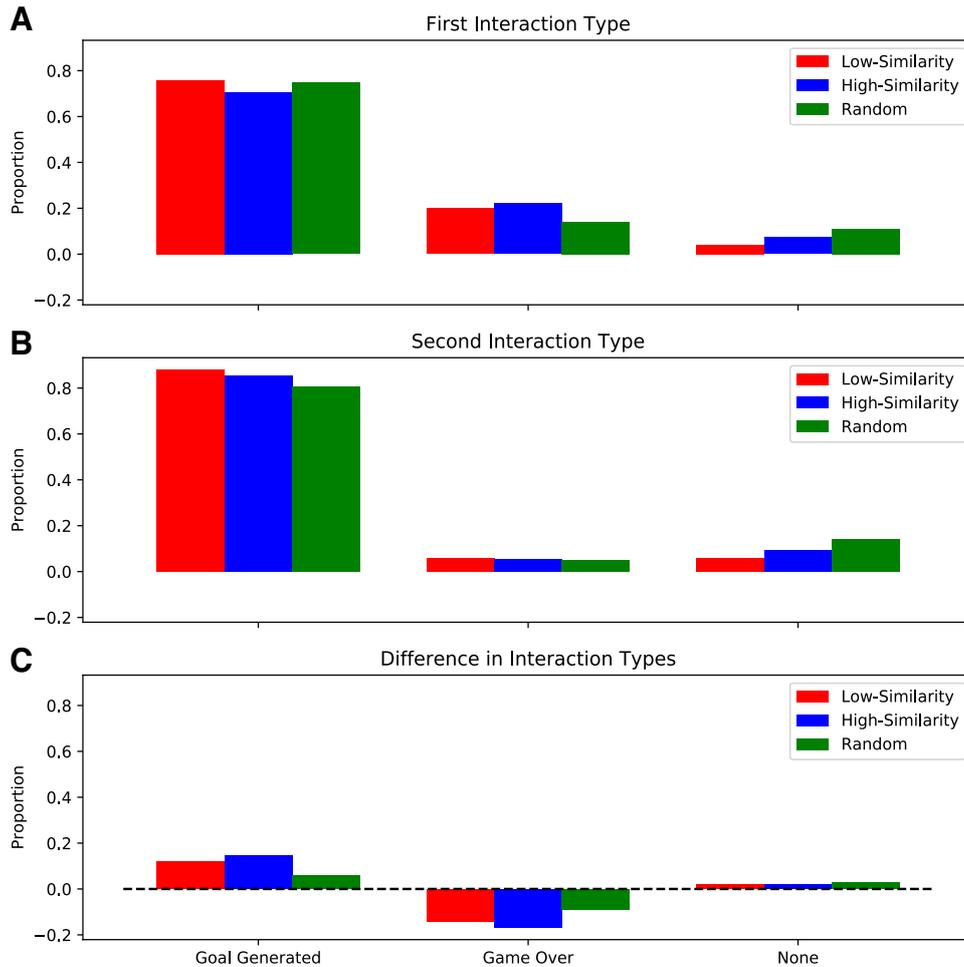


Figure 7.28: *The number of interaction types for the final test level. (A) The number of first interaction types summed over all participants. An interaction is defined as a collision with another object. The x-axis indicates the object that was first interacted with (excluding the goal object) and the colour of the bars indicates the training regime. (B) Same as A but for second interaction types. (C) The difference between A and B.*

7.3.3 Discussion

In Experiment 2 we explored how the degree of perceptual similarity between consecutive 2D video games affected transfer performance. The 2D video games all shared the same underlying relational rule, which allowed the perceptual features of the games to vary while maintaining the same task structure. Participants received 19 training levels followed by a final test level, which involved completely novel perceptual features¹. In total, we explored the effect of three different training regimes.

¹See Appendix C for the results of a shortened version with only 9 training levels. Our overall results remained the same, with the perceptual similarity between consecutive games having no

In the high-perceptual similarity condition consecutive games differed by a single feature; an object was chosen at random and either the colour, texture or shape was changed. In the low-perceptual similarity condition the training games from the high-perceptual similarity condition were taken and shuffled, which decreased the perceptual similarity between successive games but kept the training games constant. Finally, in the random condition, all the features of the objects were chosen at random for each training game. This not only lead to low-perceptual similarity between consecutive games but also greatly increased the amount of information available during training.

Analysis of the training scores across the groups indicated that participants in the random group scored significantly worse on the first trial of each level. This suggests that the increased randomness in the random condition made it harder for participants to transfer strategies between levels that enabled them to do ‘zero-shot’ transfer i.e. infer the best policy without the need for feedback. Interestingly when looking at the total score for each level during training there were no significant differences between training regimes. One possible explanation for this is that participants in the random condition were able to recover from their deficit in first score by using ‘one-shot’ strategies to achieve a good total score.

Our key measure of transfer ability was performance on the final test level, which involved completely novel features. Analysis of these test scores indicated no significant differences between training regimes. This suggests that the training benefit of the high- or low- perceptual similarity condition over the random condition was not present at test. One potential explanation for this is that any beneficial strategies that were being used in the high or low perceptual similarity conditions during training were just exploiting the spurious correlations present in the training games. Such strategies could not be utilised in the test game because it involved completely novel perceptual features and so the underlying relational rule was critical for inferring which object to interact with on the first trial.

The results of the regression analysis further highlighted that there were no differences between training regimes with respect to test performance. Indeed, the main predictors of test performance appeared to be participants’ familiarity with

effect on transfer ability.

the controls before the games began, and also their subsequent improvement in using these controls. This highlights that a substantial source of difficulty in the games was controlling the player object. We also tested the hypothesis that prior attention to texture in a free-classification task would improve participants' ability to identify the underlying relational texture rule and therefore improve transfer performance. However we also found that the number of texture matches made in the free-classification task was not a significant predictor of performance on the test level.

To try to understand participants' decisions during the task, and the strategies that they were using, we analysed which objects they interacted with on each level. More specifically, we looked at their first and second interactions on each level (not including the goal object). This analysis gave us an insight into whether participants were performing 'zero-shot' or 'one-shot' transfer. For example, if participants understood the underlying relational rule then they could infer which object to interact with on the test level without the need for feedback ('zero-shot'). In comparison if participants understood that there was always one object that generated a goal and one object that ended the game, then they could use the first trial to infer which is which and then perform optimally ('one-shot' transfer). This approach shares many similarities with Harlow (1949)'s work on meta-learning, whereby primates would learn from a single trial which object contained a food item and then use this information to perform optimally.

Across all training regimes, the frequency with which participants correctly first interacted with the goal-generating object increased over the course of training. This corresponded to a decrease in the number of game-ending first interactions. This provides evidence that participants were indeed learning how to perform 'zero-shot' transfer across training games. When looking at second interactions, the number of interactions with the goal-generating object was even larger and the number of interactions with the game-ending object was even lower compared to the first interactions. This suggests that participants were also employing 'one-shot' learning strategies to gain points during training. Investigation of object interactions during training for individual training regimes revealed that participants in the random condition appeared to be worse at differentiating between the goal-generating and

game-ending objects in their first interaction. As a result they were then less likely to interact with another object after their first interaction. This suggests that the increased differences between consecutive games in the random condition may have reduced the level of exploration of the participants. This is consistent with the significantly worse first training scores reported in the random condition.

Analyses of first and second interactions on the final test level revealed that participants were able to transfer both ‘zero-shot’ and ‘one-shot’ strategies to a completely novel game. Across all groups the first interaction type was most likely to be the goal-generating object indicating that participants were able to use the underlying relational texture rule to infer which object to interact with. Similarly, the second interaction type was even more likely to be the goal-generating object suggesting that participants could use the result of the first interaction to further infer which object to interact with. We found no evidence for differences in object interactions between the three training regimes. This further supports the idea that any benefit of the high- or low-perceptual similarity conditions over the random condition during training did not hold when a truly novel game was presented.

In the next experiment we obtain a measure of ceiling performance by telling the participants the rules of the games beforehand. This helps us to understand how participants perform on the games given perfect knowledge of what they need to transfer.

7.4 Experiment 3

In Experiment 2 we saw evidence of ‘zero-shot’ and ‘one-shot’ transfer but with no differences in test level performance between the different training regimes. One explanation, as in Experiment 1, for why there were no differences between the training regimes was that participants were performing at near ceiling. This raised the question of how participants would perform on the task if they were told the rules of the game before-hand; i.e. if they had perfect knowledge of the games. In addition, it is unclear how knowledge of the game translates into transfer performance and whether the training regime can aid the use of explicit knowledge for transfer. The purpose of Experiment 3 was to explore these questions and so we

repeated Experiment 2 but told the participants the rules of the games before-hand. We therefore refer to Experiment 3 as the control condition and Experiment 2 as the non-control condition.

7.4.1 Methods

Both the games and experimental procedure were the same as Experiment 2. The only difference was that participants were told the rules of the game before starting the training levels. Figure 7.29 shows a screenshot of how the rules were presented to the participants. In total 156 participants were recruited for the control condition (Male=98, Female=56, Undisclosed=2). As before, participants had to be aged 18-30, be fluent in English and be using a desktop computer. Participants were removed from the analysis if they had zero key presses on any of the levels. As a result there were 55 participants in the high-degree of perceptual similarity group, 42 participants in the low-degree of perceptual similarity group and 47 participants in the random group.

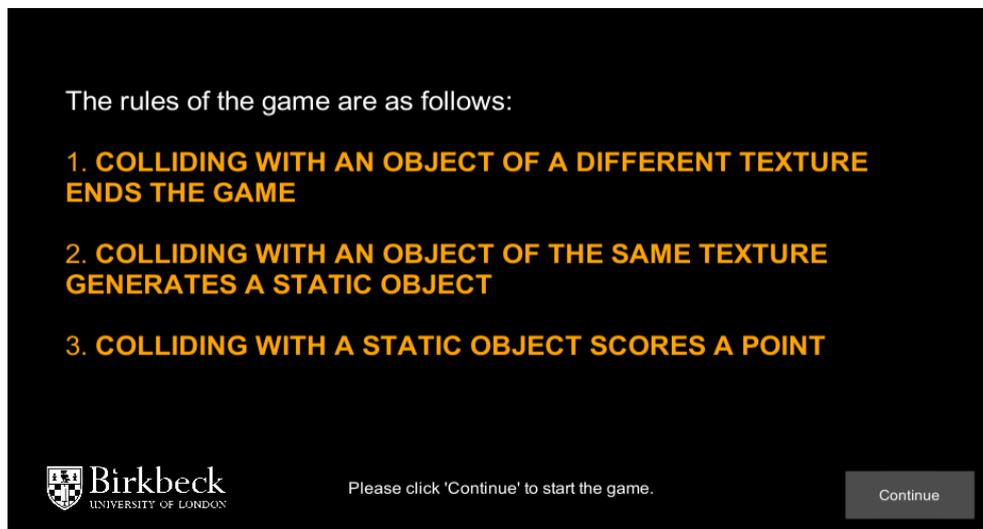


Figure 7.29: *Screenshot of how the game rules were presented to the participants before they started playing the games.*

7.4.2 Results

Training Performance

Figure 7.30 shows the scores over training for the control condition split by training regime. As before, across all training regimes there was a marked increase in per-

formance early on followed by a steady improvement for the remainder of training. Figure 7.31 shows the first and total scores summed over all the training levels for the control condition (see Appendix D, Figures D.1 and D.2 for histograms). The results of a Kruskal-Wallis rank sum test indicated that the summed first scores over training were significantly different between training regimes ($\chi^2(2, N=144)=7.9, p=0.019$) but that the summed total scores were not ($\chi^2(2, N=144)=5.9, p=0.053$). Pairwise comparisons using Wilcoxon rank sum tests with bonferroni corrections revealed that for both first scores ($p=0.011$) and total scores ($p=0.048$) the random condition was significantly lower than the high-perceptual similarity condition. No other pairwise differences were found between the different training regimes (first scores: low vs high: $p=0.639$, low vs. random: $p=0.671$, total scores: low vs. high: $p=0.719$, low vs. random: $p=0.791$).

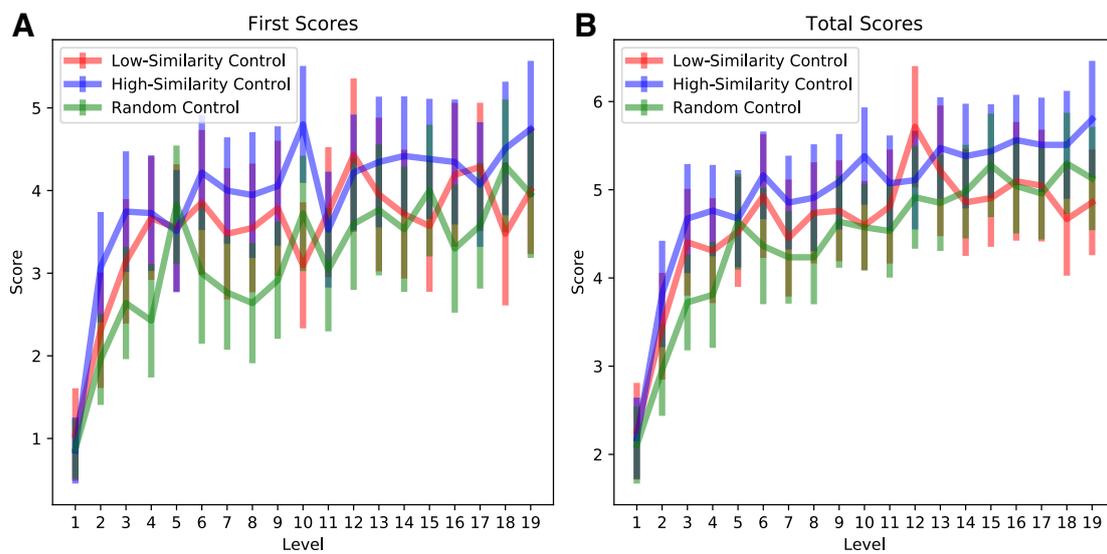


Figure 7.30: Scores for the control condition over the whole training trajectory for high-perceptual similarity, low-perceptual similarity and random conditions. (A) First trial scores. (B) Total scores. Error bars represent 95% confidence intervals.

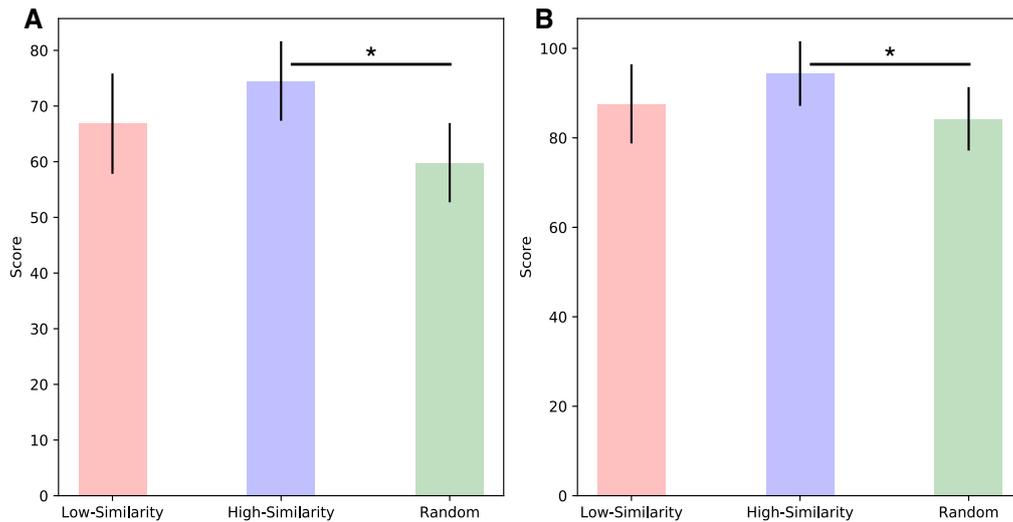


Figure 7.31: Scores for the control condition summed over all training levels for high-perceptual similarity, low-perceptual similarity and random conditions. **(A)** First trial scores. **(B)** Total scores. Error bars represent 95% confidence intervals.

Figure 7.32 shows the scores over training for both the control and non-control conditions. Similarly, 7.33 shows scores summed over training for both the control and non-control conditions. The results of a Wilcoxon rank sum test between the control and non-control conditions revealed that participants scored significantly higher first scores for each training regime when they were told the rules beforehand, except for the low-perceptual similarity regime (low-perceptual similarity: $W=2519.5$, $p=0.061$, high-perceptual similarity: $W=3691.5$, $p<0.001$, random: $W=3064.5$, $p=0.002$). In terms of total scores, performance was significantly higher for each training regime when they were told the rules beforehand (low-perceptual similarity: $W=2673.5$, $p=0.010$, high-perceptual similarity: $W=3689.5$, $p<0.001$, random: $W=3052$, $p=0.002$).

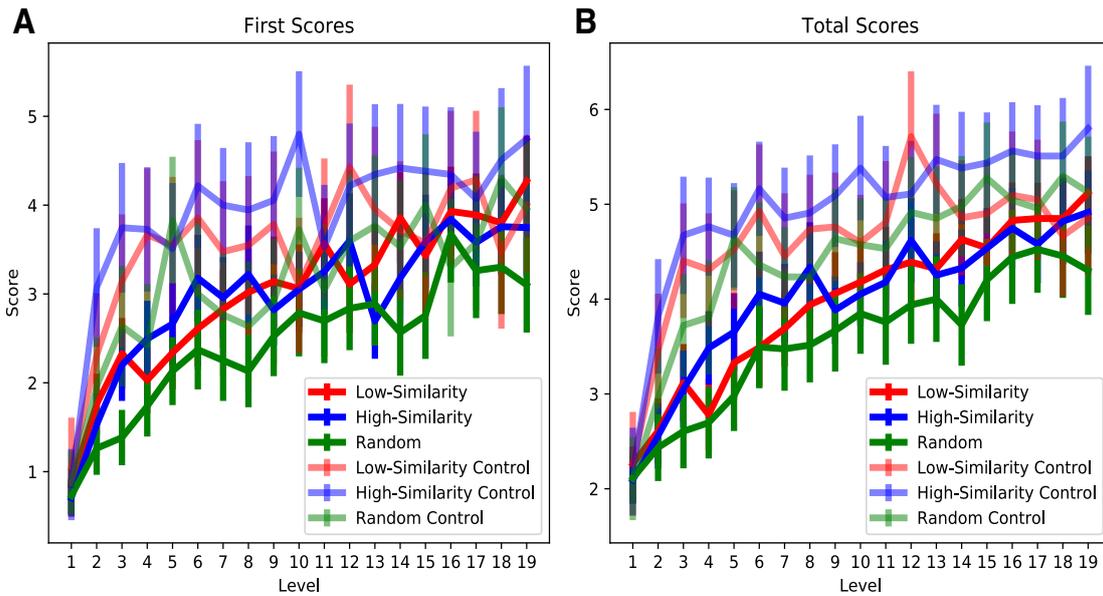


Figure 7.32: Scores for both the control and non-control conditions over the whole training trajectory for high-perceptual similarity, low-perceptual similarity and random conditions. Transparent lines represent the control condition and solid lines represent the non-control condition. (A) First trial scores. (B) Total scores. Error bars represent 95% confidence intervals.

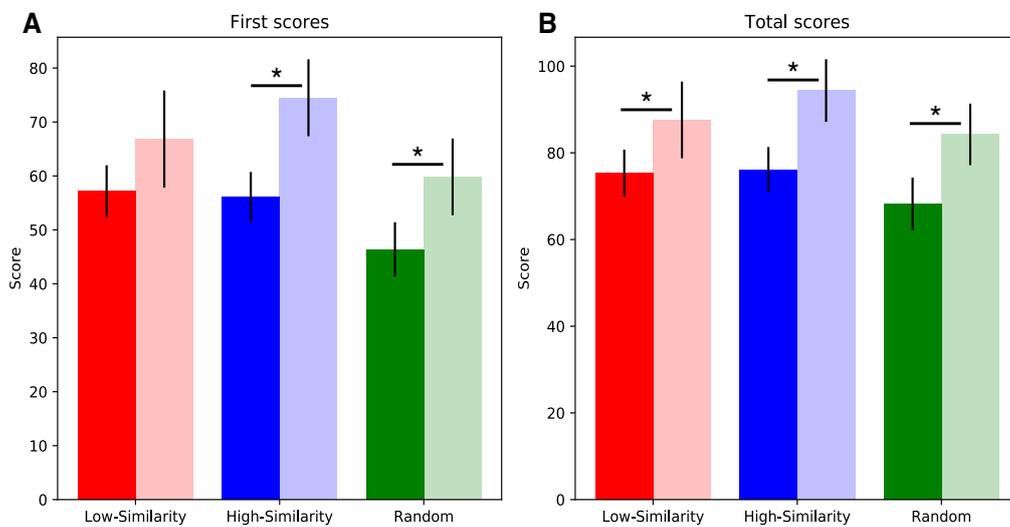


Figure 7.33: Scores for the control and non-control conditions summed over all training levels for high-perceptual similarity, low-perceptual similarity and random conditions. Transparent bars represent the control condition and solid bars represent the non-control condition. (A) First trial scores. (B) Total scores. Error bars represent 95% confidence intervals.

Test Performance

Figure 7.34 shows the first and total scores for the final test game in the control condition (see Appendix D, Figures D.3 and D.4 for histograms). The results of a Kruskal-Wallis rank sum test indicated that both the first scores ($\chi^2(2, N=144)=7.1$, $p=0.029$) and the total scores ($\chi^2(2, N=144)=10.8$, $p=0.005$) on the final game were significantly different between training regimes. Pairwise Wilcoxon rank sum tests with bonferroni corrections showed that in the case of the total scores, the high-perceptual similarity group was significantly higher than the random group ($p=0.0025$). All other pairwise comparisons were non-significant (first scores: low vs. high: $p=0.068$, low vs. random: $p=1.000$, high vs. random: $p=0.079$, total scores: low vs. high: $p=0.627$, low vs. random: $p=0.278$)

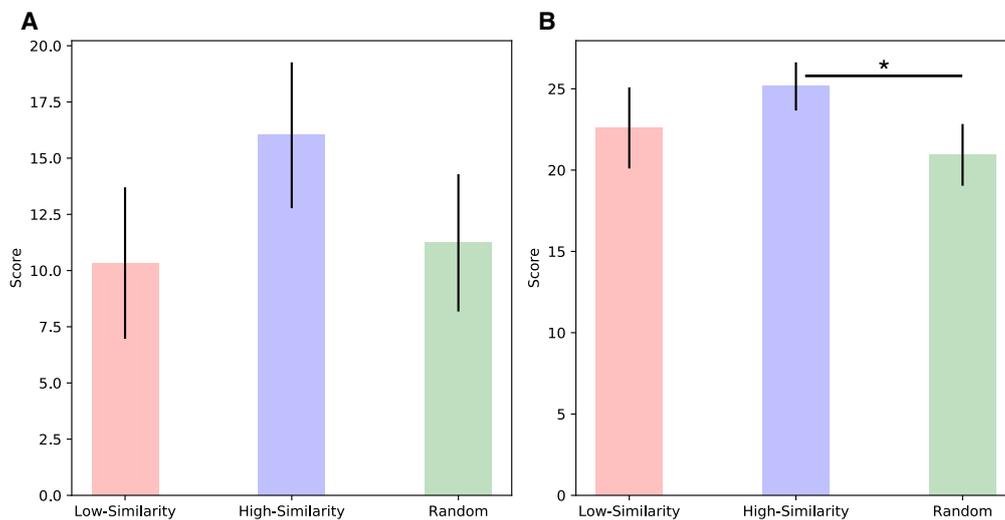


Figure 7.34: Scores for the control condition on the final test game for high-perceptual similarity, low-perceptual similarity and random conditions. **(A)** Score achieved on the first trial of the final game **(B)** Total score achieved on the final game. Error bars represent 95% confidence intervals.

As in the non-control condition, we used the results on the test level to perform a regression analysis to control for self-reported video game experience, familiarity with the controls and improvements in using the controls over the course of the experiment. Table 7.3 shows the results of the regression analysis. In line with the non-control condition, we saw that the performance on the first motor control task, and the improvement between the first and second motor control tasks, were both significant predictors of the first and total score on the test level. However, unlike

in the non-control condition, we saw that the high-perceptual similarity group alone was a significant predictor of test level performance. The coefficients were positive indicating that participants receiving the high-perceptual similarity training regime were likely to perform better on the test level than if they received the low-similarity training regime, controlling for all other variables.

Table 7.3: Regression analysis of test scores. Each column is a separate regression using either the first or total score on the test level. Values not in brackets represent beta coefficients. Values in brackets represent 95% confidence intervals.

	<i>Dependent variable:</i>	
	First Score (1)	Total Score (2)
Experience	0.41 (-0.25, 1.07)	0.47* (0.09, 0.86)
High-Perceptual Similarity	6.74** (2.66, 10.83)	3.10* (0.74, 5.47)
Random	2.99 (-1.32, 7.29)	-0.16 (-2.66, 2.33)
Motor Task 1	-2.14*** (-3.11, -1.17)	-1.11*** (-1.67, -0.55)
Motor Task 2 - Motor Task 1	-2.09*** (-3.12, -1.06)	-1.06*** (-1.66, -0.46)
No. Texture Matches	-0.44 (-1.32, 0.43)	0.43 (-0.07, 0.94)
Constant	47.71*** (27.35, 68.08)	37.72*** (25.94, 49.51)
Observations	144	144
R ²	0.26	0.30
Adjusted R ²	0.22	0.27
Residual Std. Error (df = 137)	10.13	5.86
F Statistic (df = 6; 137)	7.82***	10.01***
<i>Note:</i>	*p<0.05; **p<0.01; ***p<0.001	

Figure 7.35 shows the first and total scores for the final test level for both the control and non-control conditions. Wilcoxon rank sum tests indicated that in the high-perceptual similarity training regime the first score ($W=3497$, $p<0.001$) and the total score ($W=3628.5$, $p<0.001$) was significantly higher in the control condition compared to the non-control condition. No significant differences were found between the control and non-control conditions for the other training regimes (first scores: low: $W=2113.5$, $p=0.953$, random: $W=2271.0$, $p=0.817$, total scores: low:

W=2390.0, p=0.195, random: W=2420.5, p=0.695).

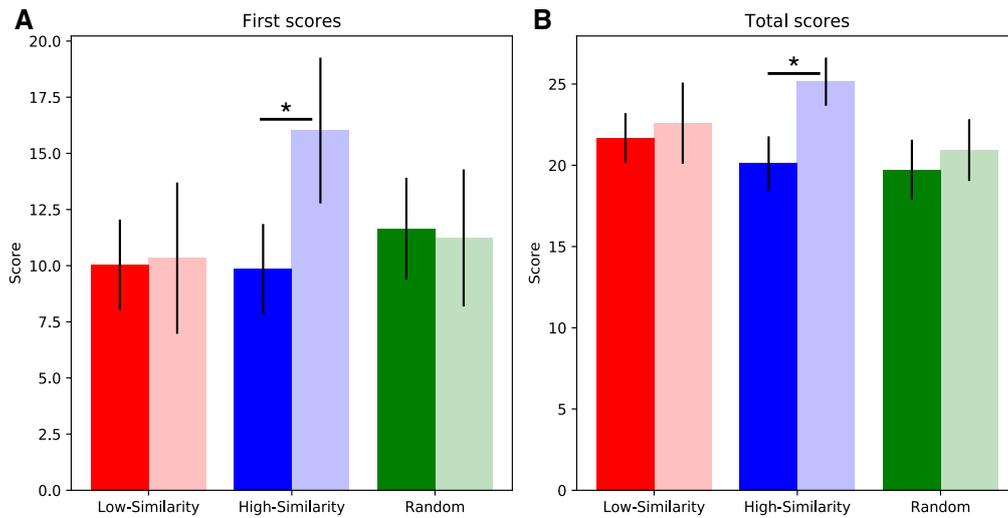


Figure 7.35: Scores for the control and non-control conditions on the final test game for high-perceptual similarity, low-perceptual similarity and random conditions. Transparent bars represent the control condition and solid bars represent the non-control condition. (A) Score achieved on the first trial of the final game (B) Total score achieved on the final game. Error bars represent 95% confidence intervals.

7.4.2.1 Object Interactions

Finally, we looked at the different object interactions the participants made in the control condition using a series of chi-squared tests. Figure 7.36 shows the proportion of interaction types over the course of the experiment collapsed across all training regimes. As in the non-control condition, the proportion of goal-generating first interactions was larger on the final test level compared to the first training level ($\chi^2(1, N=288)=17.9, p<0.001$) and the proportion of game-over first interactions was lower ($\chi^2(1, N=288)=16.2, p<0.001$). The proportion of goal-generating first interactions on the final test level was significantly larger than the proportion of game-over first interactions ($\chi^2(1, N=288)=141.8, p<0.001$). This again suggests that the vast majority of participants were able to perform ‘zero-shot’ transfer by using the underlying relational rule to infer which object to interact with on the final test level. However, unlike in the non-control condition, on the test level the proportion of goal-generating and game-over second interactions was not significantly different from the proportion of goal-generating and game-over first interactions (goal-generating interactions: $\chi^2(1, N=288)=0.1, p=0.741$, game-over interactions:

$\chi^2(1, N=288)=0.3, p=0.584$). This suggests that in the control condition participants were less likely to use a ‘one-shot’ strategy compared to the non-control condition.

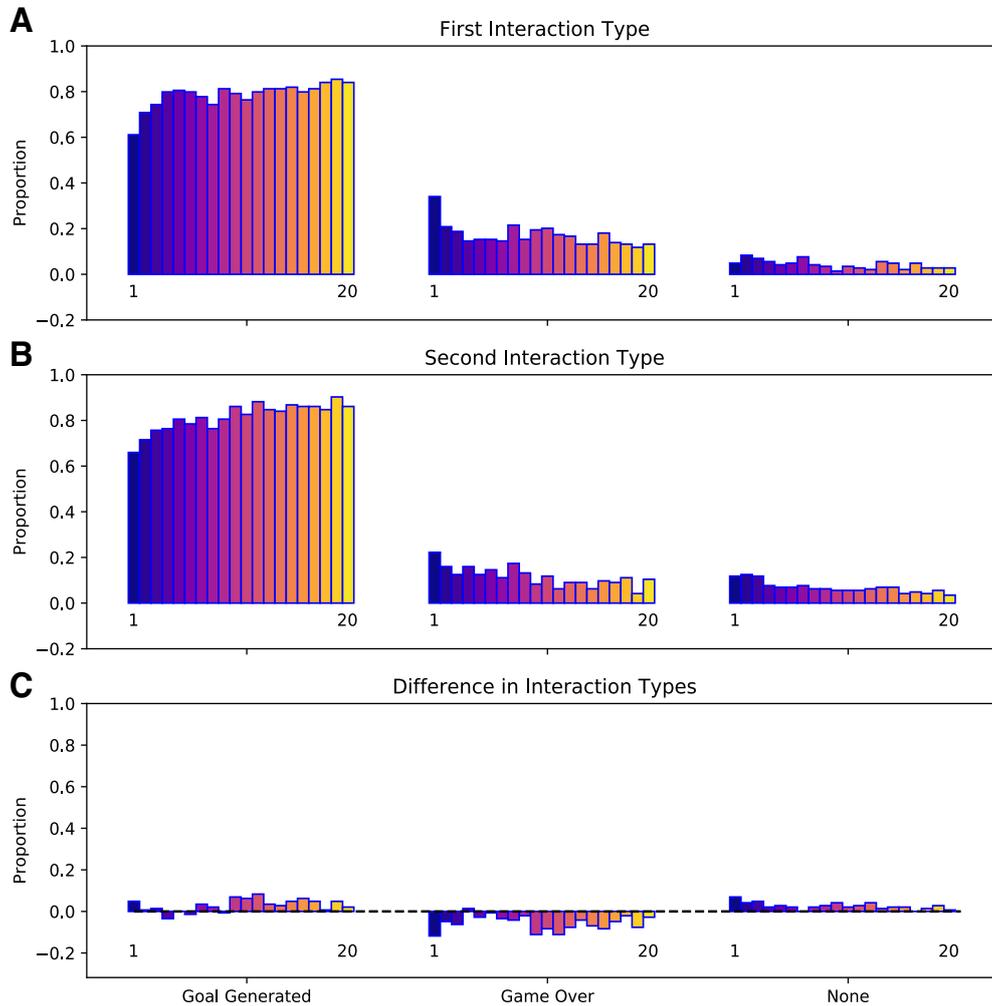


Figure 7.36: *The number of interaction types during learning. (A) The proportion of first interaction types for all participants. An interaction is defined as a collision with another object. The x-axis indicates the object that was first interacted with (excluding the goal object) and the colour of the bars indicates the level number (blue is first training level and yellow is final test level). (B) Same as A but for second interaction types. (C) The difference between the first and second interaction types.*

Figure 7.37 shows the same as Figure 7.36 but spilt by training regime. Figure 7.38 shows just the test level object interactions spilt by training regime. The results of a chi-squared test revealed that on the test level there were no significant differences between the training regimes in terms of first ($\chi^2(4, N=144)=5.2, p=0.271$) or second ($\chi^2(4, N=144)=6.8, p=0.145$) object interactions.

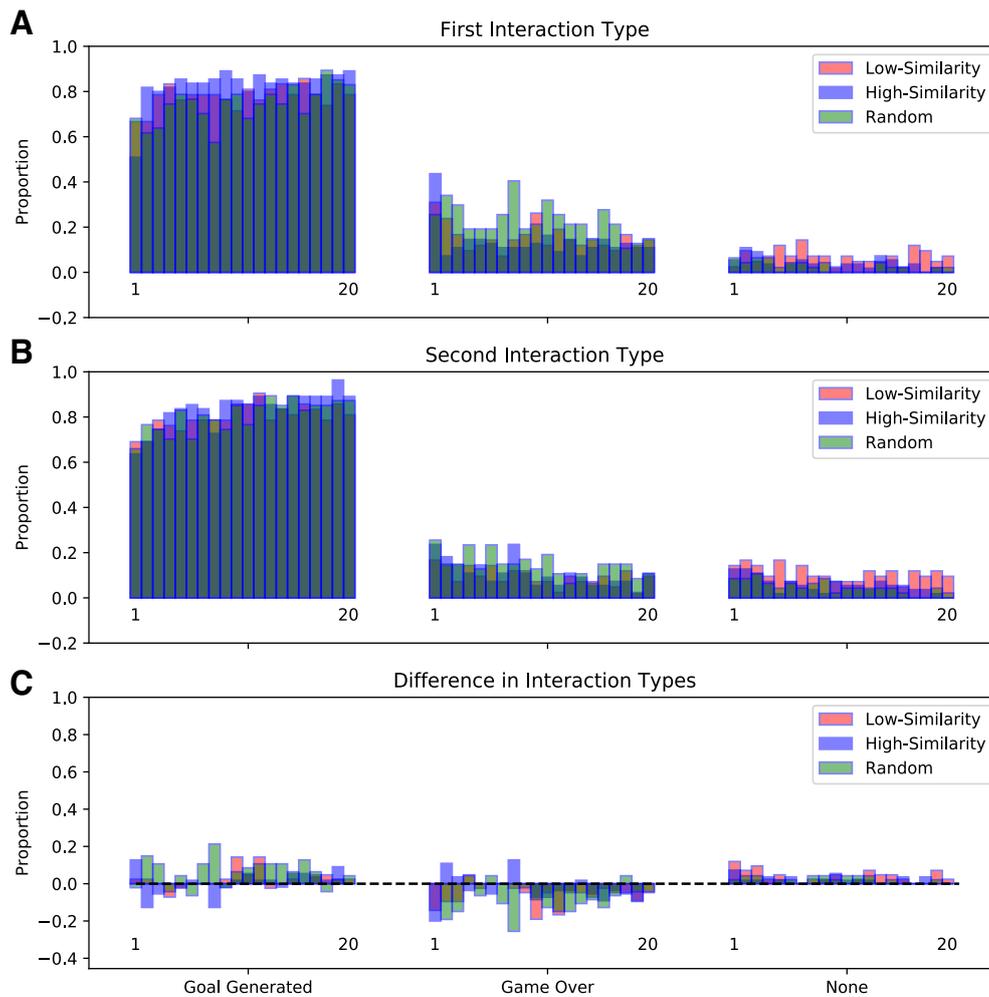


Figure 7.37: *The proportion of interaction types during learning split by training regime. (A) The proportion of first interaction types for all participants. An interaction is defined as a collision with another object. The x-axis indicates the object that was first interacted with (excluding the goal object) for each level of the experiment (1-20). The colour represents the training regime. (B) Same as A but for second interaction types. (C) The difference between the first and second interaction types.*

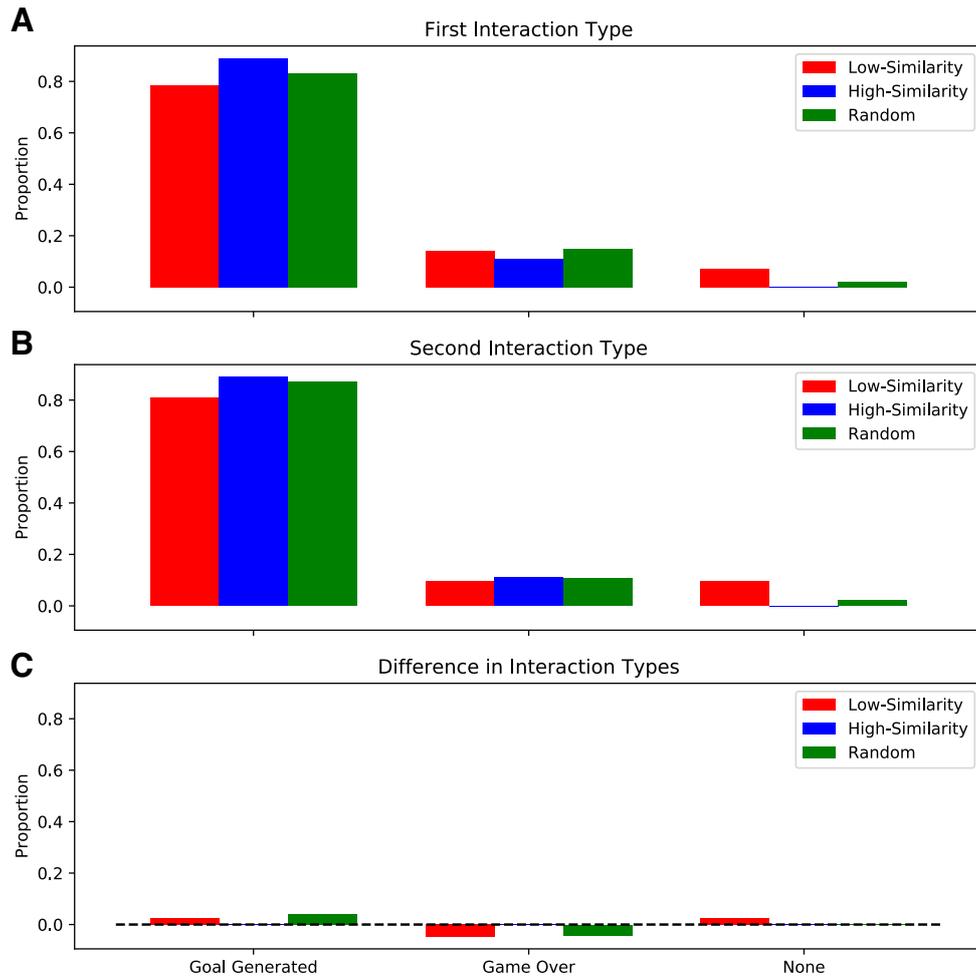


Figure 7.38: *The number of interaction types for the final test level. (A) The number of first interaction types summed over all participants. An interaction is defined as a collision with another object. The x-axis indicates the object that was first interacted with (excluding the goal object) and the colour of the bars indicates the experience group. (B) Same as A but for second interaction types. (C) The difference between A and B.*

7.4.3 Discussion

In Experiment 3 we repeated Experiment 2 but told the participants the rules of the games before-hand, which resulted in a control and non-control condition respectively. The purpose of this was (1) to explore how people performed with perfect knowledge of the games and (2) to investigate whether the degree of perceptual similarity between consecutive games had an impact on people’s ability to use explicit knowledge for transfer.

With respect to training performance in the control condition, the results were

similar to non-control condition in that participants in the random group performed significantly worse than those in the high-perceptual similarity group. This suggests that even if participants have knowledge of the underlying games rules, the large variation in perceptual features between games and the increased combination of features increases the difficulty of learning. Indeed, this may explain why in the non-control condition we saw no difference between training regimes. If the key factor affecting transfer is the degree of perceptual similarity and novelty of the features then all regimes saw a completely novel test level and so transfer was equally impacted. As expected when comparing training performance between the non-control and control conditions, participants predominantly performed better if they were told the rules before-hand. This indicates that participants were able to understand the rules and utilise them during training to speed up learning.

In the non-control condition we saw no differences between training regimes on the final test level. In contrast, there was a significant effect of training regime on test performance in the control condition. In particular, participants in the high-perceptual similarity condition appeared to perform better than those in the other two training regimes. This suggests that the training regime may interact with the use of explicit knowledge for transfer. Why might high-perceptual similarity training allow for better use of explicit knowledge for transfer? One possibility is that consecutive games with a high-degree of perceptual similarity makes it easier for participants to integrate their explicit knowledge with the feedback they are getting from the games. Our hypothesis is that if perceptual features are changing gradually between successive games then it makes it easier to test the explicit knowledge provided. In comparison, if many perceptual features are changing between games then it takes time to process the new features and align them with the explicit knowledge provided.

When comparing test performance between the control and non-control conditions, performance was only better in the control condition for the high-perceptual similarity training regime. This suggests that ceiling performance (i.e. average performance given knowledge of the rules) is not the same for the different training regimes. More specifically, performance on the test level for both the low-perceptual similarity and random conditions was the same regardless of whether the partici-

pants were told the rules of the games beforehand or not. This suggests that for these two training regimes the participants in the non-control condition were performing as well as if you had told them the rules before-hand. Explicitly telling participants the rules of the games only seemed to have a positive impact on performance when consecutive games had a high-degree of perceptual similarity.

Analysis of the object interactions in the control condition showed that participants learnt how to perform ‘zero-shot’ transfer over the course of the experiment. In addition, further analyses of the final test level indicated that unlike in the non-control condition, participants in general did not appear to rely on ‘one-shot’ transfer techniques. This makes sense given that participants were provided with the rules of the game, which allowed for better performance than ‘one-shot’ transfer techniques. Analyses of the object interactions on the final test level in the control condition did not reveal any significant differences between the different training regimes. If participants in the high-perceptual similarity group were indeed better at using the explicit knowledge for transfer then you would expect to see more goal-generating first interactions and less game-over first interactions compared to the other training regimes. While no significant effect was observed, a trend could be observed in this direction with the high-perceptual similarity group having the highest proportion of goal-generating first interactions and the lowest proportion of game-over interactions.

7.5 General Discussion

How people acquire knowledge and transfer it to novel situations is still an open question. Many believe that it relies upon the acquisition of relational knowledge that is independent of specific perceptual features (Gentner, 1988; Wilson et al., 1985; Cook and Wasserman, 2007; Torrey, 2009; Holyoak, 2012). One factor that is known to impact the ability to transfer knowledge to novel situations is the ordering of training examples (Kotovsky and Gentner, 1996; Gentner et al., 2007; Carvalho and Goldstone, 2014a,b; Rohrer et al., 2014; Kornell and Bjork, 2008). More specifically, the degree of perceptual similarity between training examples appears to have an affect on how people perform on a novel test example. In the field of

analogy, a high-degree of perceptual similarity between examples has been proposed to improve the acquisition of knowledge for transfer (Kotovsky and Gentner, 1996; Gentner et al., 2007). In comparison, in the field of concept learning, a low-degree of perceptual similarity has been proposed to improve the acquisition of knowledge for transfer (Rohrer et al., 2014; Kornell and Bjork, 2008). The experimental design of analogy and concept learning tasks have several differences. For example, analogy tasks typically involve aligning one domain with another based on the underlying relational structure whereas concept learning tasks typically involve discriminating between concepts based on perceptual features. Transfer in these cases may therefore involve different cognitive processes that are differentially affected by the degree of perceptual similarity between consecutive examples. Understanding how the degree of perceptual similarity between consecutive experiences affects transfer in these tasks is important because it can elucidate the mechanisms via which knowledge is being acquired and applied. In addition, it can inform constraints on computational models of transfer.

In this chapter we investigated how the degree of perceptual similarity between experiences affects transfer from a Reinforcement Learning (RL) perspective. Feedback from tasks in the analogy and concept learning literature may engage RL machinery but they are far from naturalistic. Naturalistic RL tasks involve sequential decision making and motor control. We therefore aimed to investigate how the degree of perceptual similarity between experiences affects transfer in a more naturalistic RL setting using simple 2D video games. The video games consisted of several 2D objects that could vary in terms of shape, colour and texture. One object was controlled by the player, one was critical for scoring points and the other two dictated whether the game ended or more point scoring objects were generated. The perceptual features of the objects were allowed to vary between games but the same underlying relational rule was present in all of the games; colliding with the object that had the same texture as the one you controlled generated an object that could be collected to score a point. High-perceptual similarity games were produced by changing one feature at a time between games whereas low-perceptual similarity games were produced by shuffling the order of the high-perceptual similarity games. This allowed for differing degrees of perceptual similarity between consecutive games

but controlled for the individual games experienced during training. A third condition was also explored where the features of the games were randomised on each level to minimise perceptual similarity and provide additional variation in the perceptual features. At the end of all conditions a final test game was administered, which involved a completely novel set of perceptual features and served as a measure of transfer performance.

Before investigating the effects of perceptual similarity on transfer in the 2D video games, we first checked that the underlying relational rules could be used to perform transfer in a simpler setting (Experiment 1). Stimuli from the 2D video games were used in a match-to-sample task, with the degree of perceptual similarity being manipulated in the same way as the full 2D video games. We found that participants could easily acquire the relational rules and apply them to novel test examples, with no differences between high-perceptual similarity and low-perceptual similarity conditions. While no differences between the two training regimes were observed we hypothesised that this may be due to the simple nature of the task. Having confirmed that the relational rule could be acquired and used for transfer we moved on to the more naturalistic 2D video games (Experiment 2).

In Experiment 2 we observed differences during training between the different training regimes. The random group demonstrated decreased training performance compared to the other two training regimes. This suggests that the low perceptual similarity between the levels and the large degree of variation in the perceptual features made it harder for participants to learn point-scoring strategies. One interpretation of this is that in the case of the low- and high-perceptual similarity training regimes there was some degree of progressive alignment occurring (Kotovsky and Gentner, 1996; Gentner et al., 2007). The reduced variation in features and greater perceptual similarity in these two conditions meant that it was easier for participants to align the games and score more points during training. In contrast, the random condition potentially introduced too many differences between the games, which made it harder for participants to align them. From looking at the object interactions made by participants, it appears that the increased difficulty of the random condition made participants less likely to explore during the training games. Participants in the random condition were more likely to have a negative

first interaction and to refrain from interacting with a second object. This suggests that the degree of perceptual similarity between consecutive games may have affected the explore-exploit trade-off that is characteristic of RL tasks (Sutton and Barto, 1998).

Despite there being differences during training between the different training regimes, we observed no differences on the final test level. This suggests that the degree of perceptual similarity between consecutive games had no effect on participants' ability to transfer the underlying relational rules. Therefore, while some evidence of progressive alignment may have existed during training, it appeared to have no benefit on the novel test level. One potential explanation for this is that the difference between the last training level and the perceptually novel test level was too large for participants to align. Indeed, progressive alignment predicts that there is a 'sweet spot' for alignment, whereby experiences are dissimilar enough to create a useful abstraction but similar enough to identify the similarities between them. One interesting point to note is that the degree of perceptual similarity in each of the conditions stayed relatively stable over learning. This is slightly at odds with the theory of progressive alignment, which suggests that experiences should get gradually dissimilar over learning so that easier abstractions bootstrap the learning of more complex abstractions. Future work should therefore include a training condition where the level of perceptual dissimilarity gradually increases over the course of training, which is consistent with the theory of progressive alignment.

In addition to progressive alignment, the literature on analogical reasoning also describes a phenomenon known as the 'relational shift' (Gentner and Hoyos, 2017). The relational shift states that children learn to focus on relational content rather than surface features over the course of development. This has some interesting implications for our results because our video games rely on relational rules. It is possible that we saw no differences between training regimes because we tested adult participants who already have the ability to focus on relational information. In comparison, testing on children may unveil differences between the training regimes because they are still in the process of learning how to focus on relational information. We hypothesize that this critical period in development could be substantially more susceptible to the effect of perceptual similarity on the ability to identify re-

lational knowledge for transfer.

With respect to the concept learning literature, our test results showed that the random training regime did not confer any benefit on the novel test game. This indicates that the predictions from the concept learning literature may not apply to our naturalistic RL task. For example, Carvalho and Goldstone (2014b)’s hypothesis that high-perceptual similarity promotes individuals to look at the similarities between exemplars, predicts that the structured training regime should help participants to discover the underlying similarities between the games. However we saw no evidence of this effect on the novel test game. One potential explanation for this contradictory finding is that the concept learning paradigms performed by (Carvalho and Goldstone, 2014b) relied on categorising different objects based on perceptual features. In comparison, our 2D video games do not rely on the identification of specific perceptual features but instead rely on the identification of a relational rule that is independent of specific perceptual features. This is perhaps more representative of analogical reasoning tasks, which also rely on the use of relational information.

What do the results of Experiment 2 tell us about the analogy between Deep RL and the brain? The lack of differences between training conditions on the final test level raises several different hypotheses about the internal computations underlying efficient RL in the brain. Throughout this thesis we have argued that the Deep Neural Networks (DNNs) used in Deep RL approaches mimic the properties of semantic memory and the neocortex. During the training of DNNs, the input data is often randomised to reduce spurious correlations that could lead to over-fitting. Classic Deep RL approaches therefore predict that a high degree of perceptual similarity should lead to the best training performance but the worse test performance because the network will over-fit to perceptual similarities between consecutive games. Interestingly, part of this prediction appeared to be true in our results. More specifically, the high- and low-perceptual similarity conditions performed significantly better than the random group during training but this benefit did not apply to the novel test game. This suggests that whatever strategies were being transferred during training in these two conditions were sub-optimal and likely exploited spurious similarities in perceptual features, as predicted by DNNs.

While the training results appear to be consistent with the predictions made

by Deep RL systems and the over-fitting of DNNs, the test results do not. Over-fitting in DNNs is typically associated with poor test performance and good training performance. DNNs would therefore predict that the test performance should be worse in the high perceptual similarity condition, however we saw no significant differences between the training conditions. One simplistic interpretation of these null results is that DNNs are a poor model of semantic memory in humans, and that the training of DNNs is fundamentally different to how humans learn during the task.

While it is easy to jump to the conclusion that DNNs are a poor model of semantic memory, there are many other reasons why this discrepancy may exist. Firstly, in order for DNNs to strongly overfit to spurious correlations in the input data they need to perform enough spurious weight updates. This means that the DNN would need to be trained for long enough on each level that its weights would move to a different part of the parameter space based on the training condition. As a result, if the levels are short enough then the DNN may not have enough time to fit the spurious correlations because the input data is changing too quickly. The weights will therefore stay in the same region of the parameter space regardless of the training condition. Future work should therefore explore whether increasing the time spent on each level leads to a benefit of low perceptual similarity training at test, as would be predicted by the over-fitting behaviour of DNNs.

One of the main themes of this thesis has been the need to consider multiple learning systems interacting with each other in order to understand the efficiency of RL in the brain. This suggests that an alternative reason for why we do not see the effects predicted by DNNs is that other learning systems are involved that compensate for the over-fitting. In particular, Deep RL systems typically avoid the problem of over-fitting to temporal correlations by relying on a hippocampal learning system. This system stores past experiences and randomly samples from them for training. This process mimics that of biological ‘replay’ and suggests that replay might be the reason why we did not see an effect of perceptual similarity on transfer performance in our experiments. In essence, the brain may be using the hippocampus to break up any correlations in the input so that the overall effect is the same regardless of whether the training involves high or low perceptual similarity.

In addition to the influence of other learning systems, it has been suggested that disentangled representations may help to alleviate DNNs reliance on interleaved training. It is largely believed that the brain represents information in a distributed fashion, a property that it shares with Deep-RL models. These distributed representations can either be disentangled (a single unit represents a cardinal dimension) or entangled (a cardinal dimension is represented as a pattern across several units). If a task requires learning based on a single cardinal dimension then disentangled representations do not suffer from interference whereas entangled representations do. Entangled representations therefore favour training on low-perceptual similarity examples in order to reduce the level of interference whereas disentangled representations are indifferent to the degree of perceptual similarity. If a task requires learning over multiple cardinal dimensions then both disentangled and entangled representations suffer from interference and both should favour low-perceptual similarity training. The fact that we saw no effect of perceptual similarity on transfer performance may suggest that the knowledge being learnt may be acting on a single cardinal dimension that is represented in a disentangled manner.

A final explanation for why we saw no effect of perceptual similarity at test is that the participants in our experiments were not relying on semantic learning at all. Indeed, in this thesis we have argued that classical DNNs are best used to represent sensory cortices that learn slowly over many experiences. However, the duration of our experimental task was around 20 minutes, which is potentially too short to engage semantic learning. Our results also remained the same when we tried a shortened version of the task that consisted of only 10 levels and lasted around 10 minutes (see Appendix C). Over these short time-scales, learning in other systems such as working memory in the Pre-Frontal Cortex (PFC) may be more responsible for the effects seen in the experiments (see Section 8.4.6 for further discussion). This would also be in line with the idea that the temporal order of exemplars in concept learning appears to effect attentional processes (Carvalho and Goldstone, 2014b), which are associated with the PFC.

One Deep-RL model that has been proposed to model transfer learning in humans is Meta-RL (see Section 5.2.1). Meta-RL works by using one RL algorithm to train another RL algorithm to become better at learning on a set of related tasks.

This form of learning known as meta-learning can lead to ‘one-shot’ transfer as the second RL algorithm improves at using feedback from novel environments until it only requires one piece of feedback to infer a suitable policy. Interestingly we saw evidence of such behaviour in Experiment 2 whereby participants were increasingly likely to behave optimally on the second interaction compared to the first interaction on the novel test level. This suggests that meta-learning in RL is indeed a powerful tool and allows for the transfer of strategies to novel environments. This being said, a key component of Meta-RL is the random selection of tasks from a collection of related tasks for training. As with standard DNNs, this helps to reduce spurious correlations between training updates. However, as already discussed, we saw no negative effect of high perceptual similarity training on test performance. This suggests that the training regime for Meta-RL may be unrealistic in its current format.

Experiment 3 was the same as Experiment 2 with one crucial difference: participants were told the rules of the games before training. The results of this experiment revealed that the performance of the low-perceptual similarity and the random groups on the final test level were the same as Experiment 2. On the other hand, the high-perceptual similarity group performed significantly better in Experiment 3 compared to Experiment 2. This is surprising because it suggests that in the low-perceptual similarity and random groups, the participants were performing at ceiling in Experiment 2. In contrast, the ceiling of the high-perceptual similarity group appears to be higher than the other groups. One possibility is that given enough training levels all other conditions would eventually reach the level of performance demonstrated by the high-perceptual similarity group that were told the rules before-hand.

We propose that these results can be explained by the fact that even when participants are provided with the rules of the game they still have to align those rules with the game-play itself. When consecutive games have a high-degree of perceptual similarity the games remain more consistent making it easier to project the rules onto consecutive games as many of the objects are the same as before. In comparison, when the degree of perceptual similarity is low it takes more time to identify how the objects relate to the rules when faced with a new level. As

a results participants that have already been told the rules in the high-degree of perceptual similarity condition learn to align the rules with the game-play quicker and so are better placed to transfer their knowledge when a novel test game is presented. It appears that knowledge of the game rules is not enough for perfect transfer, participants also have to practice using them before they can be utilised for transfer, and this practice is influenced by the degree of perceptual similarity between games.

The fact that we did see a benefit of high perceptual similarity training when participants were told the rules of the game before-hand has interesting implications for Deep RL models of human RL. Currently, it is still an open question how explicit instructions in written form can be provided to Deep RL systems. However, our results suggest that future work in this direction should account for the fact that a period of learning is still required after the explicit instructions have been provided. In addition, this learning period should be sensitive to the degree of perceptual similarity between experiences and that this ultimately will have an effect on transfer performance.

In the final chapter of this thesis we will discuss the general implications of the work we have conducted throughout the thesis. We hope that this will help to distill the contributions we have made to our understanding of efficient RL in both humans and machines. We will also highlight key limitations of the work and promising areas for future research. With respect to the work conducted in this chapter, Sections 8.4.5 and 8.4.6 are particularly relevant as they discuss the difficulties of defining perceptual similarity and of investigating learning in naturalistic tasks.

Chapter 8

Discussion

Overview

In this chapter we discuss the results of our work and their broader implications for the research community. We start by re-iterating the goal of this thesis (Section 8.1) and summarise the key computational (Section 8.2) and empirical (Section 8.3) findings of our work with respect to this goal. We then move on to discuss the limitations of the approaches used, and highlight promising avenues of future research that could address these limitations (Section 8.4). This is followed by a discussion of some of the practical lessons that have been learnt over the course of this thesis (Section 8.5) and our concluding remarks (Section 8.6).

8.1 Goal of The Thesis

Humans demonstrate rapid learning and inference in novel situations throughout their life. Often these situations involve sparse reward signals and are thought to engage Reinforcement Learning (RL) mechanisms in the brain, which transform perceptual input into action. Recent advances in Deep RL have opened up promising avenues for modelling this transformation; however, it appears to lack the efficiency and flexibility of human learning and behaviour. The purpose of this thesis was to build an analogy between the brain and Deep RL in order to understand this discrepancy and investigate the internal computations that support efficient RL in the brain and the external factors that influence them. To achieve this we used a

combination of both computational modelling and empirical approaches.

8.2 Summary of Computational Findings

Chapters 3-6 of this thesis investigated the computational properties of different learning systems in the brain and how they contribute to fast and efficient RL. Central to this investigation was *Complementary Learning Systems (CLS) theory*, which we used to organise past literature and inform our own research. CLS theory posits that the brain consists of two key learning systems: the neocortex and the hippocampus. These two systems have complementary properties in that the neocortex slowly learns over-lapping representations while the hippocampus rapidly learns pattern-separated representations. In Chapter 4 we proposed a novel algorithm called *Complementary Temporal Difference Learning (CTDL)*, which utilised these complementary properties to improve the performance, stability, efficiency and flexibility of well known Deep RL algorithms. Central to the performance of CTDL were two key properties: (1) both the neocortical and hippocampal learning system directly contributed to action selection, and (2) the errors produced by the neocortical learning system were used to guide the learning of the hippocampal learning system. Importantly, CTDL predicts that the hippocampus should be biased towards storing experiences that the neocortex is poor at evaluating and that the hippocampus is most beneficial when the task involves rare but highly salient events.

While CLS theory makes a dissociation between the neocortex and the hippocampus, many sub-divisions exist within these two learning systems. In Chapter 5 we proposed that the sub-division between the Pre-Frontal Cortex (PFC) and sensory cortices may be critical for fast and efficient RL. The PFC has long been associated with executive control and a growing body of research suggests that it has an important influence during RL. One particular function of the PFC is selective feature-based attention, which is thought to guide RL towards important features based on the task at hand. This is thought to improve the efficiency of RL in the brain by reducing the dimensionality of state representations. With this in mind, in Chapter 6 we proposed a novel algorithm called *Selective Particle At-*

tention (SPA), which imbued a Deep RL agent with the ability to perform visual feature-based attention. SPA used a particle filter to model the PFC and incorporated both bottom-up and top-down attention to guide feature selection. Our results showed that SPA greatly improved the performance, flexibility and efficiency of Deep RL algorithms. Crucially, SPA was able to quickly and dynamically change the features being attended to based on the task at hand. This not only reduced the dimensionality of the problem but also flexibly re-purposed existing representations without the need to learn new ones end-to-end. From a biological perspective, SPA demonstrates that selective feature attention may be a viable mechanism for flexibly re-purposing existing representations in sensory cortices for efficient RL.

In the case of both CTDL and SPA, the proposed algorithms highlight the importance of considering the brain as a group of interacting learning systems each with their own set of computational properties. It appears unlikely that a single learning system can support the kinds of flexible behaviour shown by humans. It is therefore critical that we try to understand important demarcations between learning systems in the brain and what their distinct computational properties are. Here we highlight the slow learning of over-lapping representations in the neocortex, the fast learning of pattern separated representations in the hippocampus, and the dynamic, top-down attention of the PFC as key computational properties. We also emphasise the importance of the interactions between these different learning systems. In particular, we propose that the errors generated by the neocortex are used to bias the content of the hippocampus and error signals generated by the PFC are used to re-purpose representations in sensory cortices. These are just a selection of the many possible interactions between these learning systems and it is likely that a plethora of different interactions are required to fully capture the complexity of human RL.

8.3 Summary of Empirical Findings

Chapter 7 moved away from the internal computations supporting efficient RL in the brain and instead focused on the effect of external factors in the environment. The investigation of external factors is important because it helps to constrain com-

putational theories by making predictions about when RL in the brain should be more or less efficient. While many external factors may influence RL in the brain, we chose to focus on how the degree of perceptual similarity between successive experiences affects the ability to transfer knowledge between situations.

The ability to transfer knowledge between situations is critical for efficient RL because it allows humans to evaluate new states and actions using knowledge learnt in previous situations. Several theories make competing predictions about how transfer should be affected by the perceptual similarity between consecutive experiences. In order to address these conflicts we designed a 2D video game that consisted of a set of underlying relational rules. These relational rules allowed the perceptual features of the game to vary between levels while maintaining the same task structure across all levels. A key benefit of using this 2D video game was that it allowed us to explore the effect of perceptual similarity on transfer in a more naturalistic setting compared to previous work. The game involved making sequential decisions and performing fine motor control in a complex environment, which are characteristics that are often lacking in typical experimental psychology tasks.

Participants were split into three experimental conditions; high perceptual similarity, low perceptual similarity and random. The high perceptual similarity condition was generated by changing a single feature (colour, shape or texture) of a single object between consecutive levels. The low perceptual similarity condition was then generated by randomly shuffling the sequence of levels generated by the high perceptual similarity condition, which reduced the degree of perceptual similarity between consecutive levels. This method meant that the participants were exposed to the same levels in each condition but experienced them in a different order. Finally, in the random condition, the features of each object were randomised on every level. This led to an even lower degree of perceptual similarity between levels and a greater variety of levels compared to the high and low perceptual similarity conditions. After training on these conditions, all participants were given the same entirely novel test level to assess their ability to perform transfer.

Our results provided no evidence for an effect of perceptual similarity on the ability to perform transfer. The performance of participants on the final test level was the same regardless of the condition they were assigned to for training. Across

all conditions we saw evidence of both ‘zero-shot’ and ‘one-shot’ transfer suggesting participants transferred a range of strategies between games. Our primary hypothesis for why we did not see any differences in transfer ability between the training conditions was that the task was too easy and participants were performing at ceiling. If this were the case then the benefits of one training condition over another may not become apparent.

In order to test this possibility, we repeated the experiment again but told the participants the rules of the game before training began. The rationale for this was that it would give us a measure of ceiling performance as participants would have perfect knowledge of the underlying rules that they needed to transfer. Interestingly, participants performed significantly better in the high-perceptual similarity condition compared to the other training conditions. These results suggest that the degree of perceptual similarity between consecutive experiences may have an effect on the utilisation of explicit knowledge for transfer. Even when provided with the knowledge required for transfer, participants still need to map that knowledge onto the perceptual input provided by the games. This process appears to be facilitated by a high degree of perceptual similarity between consecutive levels. The performance of the participants in the high perceptual similarity condition was also significantly higher than the performance of the corresponding participants who were not told the rules of the game. This indicates that the lack of differences in the previous experiment were not due to ceiling effects.

8.4 Limitations and Future Work

This section covers some of the limitations of the approaches used in this thesis. It also highlights promising avenues of future research that may be able to address some of these limitations.

8.4.1 Transfer vs. Rapid Learning

Throughout this thesis we have attributed efficient RL in the brain to two main processes: (1) the transfer of past knowledge and (2) the rapid learning of new information. These processes are often hard to distinguish between, as learning is

naturally influenced by existing knowledge and existing knowledge is produced via previous learning. Both of the algorithms proposed in this thesis, *Complementary Temporal Difference Learning (CTDL)* and *Selective Particle Attention (SPA)*, attempt to explore the mechanisms needed for efficient RL in the brain. At their heart both algorithms improve the ability of Deep RL systems to rapidly learn from reward signals. CTDL achieves rapid learning by utilising a model of the hippocampus that uses a large learning rate. In contrast SPA achieves rapid learning by using a model of the Pre-Frontal Cortex (PFC) that uses approximate Bayesian inference. While these rapid learning mechanisms are explicit in both CTDL and SPA, what evidence is there for them also having the ability to transfer past knowledge to the current task?

In the case of both CTDL and SPA we explored the effect of changes in the reward function on performance. Transfer in this case is ultimately dependent on feedback in the form of a reward signal. The agents are unaware of any changes to the environment until Reward Prediction Errors (RPEs) are generated. RPEs are a crucial component of both CTDL and SPA as they are used to update both the hippocampal and PFC learning systems respectively. Both CTDL and SPA demonstrated evidence of transfer in these reward revaluation experiments. For example, in the case of both CTDL and SPA, changes to the reward function did not reduce the agents to the same level of performance as when learning first started and both algorithms were able to quickly recover. This suggests that at least some of the knowledge learnt by the algorithms was re-used when the reward function changed.

How might this transfer be occurring in CTDL and SPA? In the case of CTDL, the Self-Organizing Map (SOM) was able to quickly encode changes to the environment based on the RPE produced by the Deep Neural Network (DNN). We hypothesise that this may have helped to protect the knowledge stored in the DNN by re-adjusting the content of the SOM to account for new areas of weakness. In comparison, the Deep Q Network (DQN) appeared to fail catastrophically and demonstrated evidence of negative transfer, whereby learning on the previous task actually hindered learning of the new task. In the case of SPA, the selective attention mechanism naturally implements a form of transfer because it dynamically selects from

existing features based on their ability to predict reward. Changes in the reward function will therefore lead to the transfer of new down-stream features. In addition to down-stream features, SPA also transfers up-stream knowledge stored in the Deep RL network. The features selected by the particle filter depend on the function being approximated by the Deep RL network. As a result, when the reward function changes, the particle filter will attempt to find the best combination of features for transferring the already learnt knowledge in the Deep RL network. This may explain why the attention vector weights were different in the multiple choice task each time a particular image category was set as the target. The attention vector weights do not stay the same for a particular image category because it has to re-adjust to changes in the knowledge of the Deep RL component produced by other image categories being set as the target.

Changes to the reward function are responsible for feedback-driven transfer, however we are often faced with novel situations where the environment changes and reward feedback is not available. For example, we may have to decide whether to take a new job offer or not without the hindsight of taking similar job offers before. This kind of ‘zero-shot’ transfer relies exclusively on inference mechanisms rather than efficient learning mechanisms. The problem of ‘zero-shot’ transfer is a hard one and it relies on the successful transfer of knowledge between situations without the use of feedback. Changes to the state space provides the opportunity for ‘zero-shot’ transfer. When an agent is faced with a change in state space, they can use their knowledge of previous state spaces’ to infer the best actions without having experienced a reward in the new state space. While humans appear to be able to deal with changes in state space regularly, classic Deep RL algorithms appear to lack such an ability. It is for this reason that DQN had to be re-trained on each video game because it could not handle the changes in images between games. We explored the effect of a change in state space on the performance of SPA using the object collection game in Chapter 5. SPA demonstrated impressive performance, with performance only dropping to intermediate levels and rapidly recovering to previous levels of performance. However this is still likely to be due to reward-based feedback over several episodes rather than ‘zero-shot’ transfer. To get a measure of ‘zero-shot’ transfer in this case we would need to analyse the accuracy of SPA’s first

initial decision before any reward-based feedback is given (i.e. how long does it take to achieve the first point?).

This raises the question of what mechanisms support ‘zero-shot’ transfer in the brain? In Chapter 3, we saw how disentangled representations could support transfer when the state space changed to represent a visual scene that the agent had never seen before. By directing learning on to individual factors of variation, the knowledge learnt is invariant to changes in the other factors and therefore supports ‘zero-shot’ transfer. In Chapter 3 we also saw how relational representations formed in the hippocampus can be used to form novel inferences about experiences or transitions that have never been experienced before. In the case of both disentangled and relational representations, the representations learnt are more invariant to changes in the state space and so support action selection even in unseen state spaces. This notion of invariance is extremely important as human concepts are highly invariant to changes in perceptual input (e.g. we can recognise a dog from a multitude of different angles and contexts). It is by using these concepts to parse novel scenes that we are able to make inferences without the use of feedback. In addition, these concepts are likely to support efficient model-based RL and planning as we can make predictions about their behaviour over time. We therefore believe that matching the concepts learnt by humans and the representations learnt by Deep RL systems represents a promising avenue for achieving ‘zero-shot’ transfer. Whether this can be achieved using innate inductive biases or large amounts of rich perceptual data remains unclear. In reality it is likely that a combination of these two approaches is necessary and that different forms of transfer will require different mixes of the two. For example, innate spatial invariance in convolutional neural networks has greatly improved their ability to generalize, as has the creation of large natural image data sets for training.

8.4.2 Efficient Reinforcement Learning as a Combination of Mechanisms

In Chapters 3 and 5 we reviewed several avenues of research that attempted to understand the computations responsible for the efficiency of human Reinforcement Learning (RL). These chapters highlighted how several interacting phenomena are

likely to underlie the complex and intelligent behaviour of humans in novel situations. Unfortunately, the fact that many interacting mechanisms are likely to be responsible for efficient RL raises several difficult questions.

Firstly, there is the problem of distilling the fundamental computational properties of each learning system based on the different theoretical approaches highlighted in this thesis. In some cases it seems relatively straightforward to combine the approaches reviewed in Chapters 3 and 5. For example, in Chapter 3 we saw that the hippocampus may be responsible for learning both relational and successor representations. In this case the two phenomena seem to rely on the same computational property; namely the ability to form predictive representations. It therefore seems plausible that such a property could sustain both mechanisms or that the successor representation is just an instantiation of a cognitive map. Similarly, in Chapter 5 we saw how the Pre-Frontal Cortex (PFC) appears to be responsible for selective attention and concept formation. Again, in both these cases there is a shared computational property; namely the production of attentional signals that act on features of the input. These shared computational properties are appealing because they allow us to simplify the key mechanisms in each learning system and reconcile different avenues of research. However, while these approaches appear simple to reconcile there are many which do not. For example, we saw in Chapter 5 that the PFC may be responsible for meta-learning as well as producing attentional signals. Do these mechanisms rely on fundamentally different properties of the PFC or do they share a similar underlying mechanism? Future work must explore further which of the approaches outlined in this thesis share the same underlying mechanisms and which ones are computationally distinct. This will help to improve our understanding of the fundamental computational properties of each learning system and how they support efficient RL.

Even with the fundamental computational properties of each learning system identified, it remains a challenging task to understand how they depend on each other to produce efficient RL. For example, the neocortex provides input to the hippocampus and so the content of episodic memory is dependent on the content of semantic memory. This means that approaches such as Complementary Temporal Difference Learning (CTDL) in Chapter 4, are reliant upon research trying

to understand how Deep Neural Networks (DNNs) can more faithfully represent semantic memory; e.g. through the use of disentangled representations or the reduction of catastrophic forgetting. Indeed, it may be that the true potential of such mechanisms can only be realised once other mechanisms have been elucidated. For instance, an understanding of how the PFC and the hippocampus produce conceptual representations may improve the efficiency of models that investigate predictive representations in the hippocampus. This is because conceptual representations allow for inferences that generalise to unseen perceptual instances. Future work will therefore need to investigate how different computational mechanisms depend on each other and how advancements in our understanding of one mechanism might affect the utility of other mechanisms. If some computational mechanisms are dependent on others then this suggests that a hierarchy exists, whereby some mechanisms contribute more to the efficiency of RL than others. Identifying the mechanisms at the top of the hierarchy would therefore offer the biggest step towards reconciling the efficiency of Deep RL with human RL.

8.4.3 Further Demarcation of Learning Systems in the Brain

In this thesis we have argued that in order to understand the efficiency of human RL, one must consider the brain as a collection of learning systems working together. More specifically, in Chapters 3 and 4 we started by focusing on the striatum, neocortex and hippocampus as key learning systems for efficient RL. In Chapters 5 and 6 we then argued that the neocortex should be divided into sensory cortices and the Pre-Frontal Cortex (PFC) in order to explain phenomena such as meta-learning, concept formation and selective attention. This therefore raises the question of whether further divisions of sensory cortices should be made in order to capture efficient RL.

The neocortex has long been divided based on functional significance. For example, regions of the neocortex are often labelled based on the perceptual modality that they appear to process. However, these divisions are only approximate and increasing evidence suggests that they are overlapping. For example, multimodal signals are often reported in areas commonly associated with a single modality (Schroeder et al., 2001; Kayser et al., 2007; Bizley et al., 2007; Bizley and King, 2008). In ad-

dition, patients who lose the ability to perceive a particular modality demonstrate processing of other modalities in the region associated with the lost modality. For example, blind human patients and rodents have demonstrated event-related potentials in visual cortex produced by auditory stimuli (Alho et al., 1993; Piche et al., 2007). Conversely, deaf human patients have shown activation in auditory cortex in response to visual stimuli (Finney et al., 2001). Furthermore, deaf cats appear to have superior visual localization and motion detection abilities, which subsequently disappear when their auditory cortex is deactivated (Lomber et al., 2010). These findings suggest that sensory processing areas in the neocortex are actually highly similar and may just differ based on the inputs that they receive (Westermann et al., 2007). In line with this hypothesis, it has been suggested that the neocortex relies on discrete units of computation known as cortical columns, which share a stereotypical connectivity profile (Douglas et al., 1989; Amorim Da Costa and Martin, 2010; DeFelipe et al., 2012; Lodato and Arlotta, 2015). These columns are thought to carry out the same canonical neural computations and learning rules regardless of their location in the neocortex (Miller, 2016).

We therefore believe there is merit in considering sensory cortices as a whole, as we have done in this thesis, rather than further dividing them into separate learning systems. Indeed, the main assumption of this thesis is that sensory cortical areas share similarities with Deep Neural Networks (DNNs) in that they slowly learn overlapping representations over many experiences. The aforementioned findings support this idea of a general representation learning mechanism across perceptual modalities. Furthermore, DNNs have been found to produce similar representations to different sensory cortical areas, such as visual (Güçlü and van Gerven, 2015; Yamins and DiCarlo, 2016) and auditory (Kell et al., 2018) cortex, based on the modality that they are trained on. Future work should therefore explore multi-sensory processing and whether multi-sensory inputs to DNNs can lead to the kind of functional specification seen in the brain.

8.4.4 Issues With Comparing Deep Reinforcement Learning to Human Learning

This thesis has relied on an analogy between Deep Reinforcement Learning (RL) and human RL to understand how humans are able to efficiently respond to novel environments. In Chapters 4 and 6 we proposed two novel algorithms that used mechanisms inspired by the brain to improve the efficiency of Deep RL systems. We took this as evidence that these mechanisms may be responsible for the efficiency of human RL. However, further empirical evidence is needed to support these claims in the form of behavioural and neural evidence. Unfortunately, directly comparing the learning of Deep RL systems to human RL can be difficult for several reasons.

Firstly, by the very nature of studying transfer, Deep RL systems do not start a task with access to the same knowledge as human participants. Indeed, Deep RL systems are often trained on a task with no prior knowledge at all and even if they have been pre-trained, they lack the mechanisms to effectively use that knowledge; the very phenomenon we are trying to study. This means that Deep RL systems display very different learning trajectories to humans and this makes comparing metrics such as learning curves uninformative. Classical computational models of RL typically hand-code knowledge or task structure into the state representation of the model to circumvent this problem. However, the purpose of Deep RL is to try and learn state representations from raw perceptual input and so this negates the primary benefit of Deep RL.

The fact that Deep RL systems learn their own state representations raises another problem with comparing the learning curves between Deep RL systems and human RL. The nature and content of the learnt representations has a huge impact upon the behaviour of the overall Deep RL system. This means that until a Deep RL system is capable of learning representations that are the same as those in the human brain the behaviour of that system will be different to that of a human. This is an issue when other components of the system may faithfully mirror mechanisms found in the brain. For example the efficiency of Selective Particle Attention (SPA) in Chapter 6 is still notably lower than that of human learning. However this may simply be due to the nature of the representations the particle filter is acting on and not to do with the proposed attention mechanism. This problem is closely related

to the one outlined in section 8.4.2 where the effectiveness of a mechanism is likely dependent on other mechanisms, which may still require further research.

Despite these difficulties of directly comparing the learning profiles of Deep RL systems to human RL, both CTDL and SPA make concrete predictions that can be tested empirically. For example, CTDL makes the explicit prediction that episodic memory content should be biased by reward prediction errors generated by semantic memory. This could be tested by recording mid-brain dopamine neurons during a reward-based task that relies upon semantic knowledge, followed by an episodic memory test to check whether items from the reward-based task were best remembered when they were associated with large reward prediction errors. Predictions can also work in the opposite direction, with predictions from the brain being used to validate Deep RL algorithms. For example, in the brain the strength of top-down attention appears to get weaker as you move back through the visual stream, supposedly because it is less useful in earlier regions (Lindsay and Miller, 2018; Baluch and Itti, 2011). This can be tested in SPA by moving the position of the attention layer to earlier parts of the Deep Convolutional Neural Network (DCNN) and exploring the effect this has on performance. Therefore while direct comparisons between the learning profiles of Deep RL and human RL appears futile, qualitative predictions about the underlying mechanisms can still be used to produce valuable empirical evidence.

It is worth noting that in order to test such predictions, a high level of interdisciplinary expertise is required. Researchers need to be able to develop Deep RL algorithms, conduct well designed behavioural experiments and perform complex neural recordings and analysis. Collaboration is therefore critical if the analogy between Deep RL and human RL is to be taken further and the mechanisms underlying efficient RL in the brain are to be elucidated.

8.4.5 The Notion of Perceptual Similarity

One limitation of the empirical work in this thesis is the notion of ‘perceptual similarity’. Throughout all experiments we have defined similarity as the number of object features that differ between two levels of a 2D video game. This definition is intuitive because it is quantitative and each feature represents a cardinal property

of an object e.g. colour, shape and texture. It is therefore easy to understand and seems to reflect how we reason about similarity in the real world. Nevertheless, there are a large number of alternative definitions of perceptual similarity. For example, a mathematical definition could be the Euclidean distance between the vectors of pixels produced by two different levels. Equally, a neuroscience definition could be the similarity of neural responses in primary visual cortex between two different levels. This serves to highlight the issue of exploring the effect of perceptual similarity between consecutive experiences on transfer; how do we define what is perceptually similar?

One potential way to circumvent this problem is to have people subjectively rate the perceptual similarity of items before they are used for training. For example, one could show participants screen-shots of the different games they are going to play and have them organise the screenshots into a line based on similarity. Items next to each other would then be deemed similar based on that participants perception. A similar approach was used in Flesch et al. (2018), whereby the authors asked participants to organise stimuli into a 2D grid before they were used in a Reinforcement Learning (RL) task. This helped the authors to infer the dimensions along which participants categorised the stimuli before the RL task began.

This approach would eliminate the need to pre-define the concept of perceptual similarity as the participants would do it implicitly. It would also account for any individual differences in judgements of perceptual similarity. The results of the free-classification task in our experiments highlight why this might be important. Not all of the feature dimensions were treated equally in the task and some participants chose to weight features differently when deciding whether objects belonged to the same category or not. Therefore some participants may rate the similarity between two levels differently depending on the feature that has changed. If each participant arranged the levels based on similarity before-hand, then these individual differences would be accounted for.

8.4.6 Learning in Naturalistic Tasks

One of the attributes that made our experiments novel was the naturalistic nature of the task. Video games involve complex sequential decision making and dynamic

motor control, which are important elements of everyday behaviour. In comparison, classic behavioural paradigms such as match-to-sample tasks typically involve individual decisions and little to no motor learning. However this increase in complexity comes at a cost, as the behavioural output will contain a mix of different cognitive processes that can hide or negate the effects one is trying to investigate.

For example, we tried to account for the influence of motor learning by including a test for motor capabilities before and after the main video game. The improvement in performance on these motor tests was a strong predictor of performance on the final test level of the video game. This suggests that motor learning was a large contributor to our measure of transfer performance. It also demonstrates that learning the underlying relational rules of the video game is not the only learning problem faced by the participants. Indeed, it is possible that learning to use the controls takes priority over the learning of the underlying relational rules for many participants. Furthermore, the variation in participants' familiarity with the controls may have introduced noise into our measure of transfer performance and therefore decreased statistical power. Participants with a poor grasp of the controls may have had knowledge of the relational rules for transfer but were unable to control the object sufficiently to utilise them on the final test level. Future work should therefore explicitly test participants' knowledge of the rules after training by asking them to describe what they thought the rules were.

In addition to motor learning, the use of a naturalistic task also opened the door for a greater number of strategies or policies. This is evident from the fact that participants appeared to demonstrate signs of both 'zero-shot' and 'one-shot' learning on the final test game. Indeed, many other strategies may have existed such as 'scoring a single point and avoiding all other objects'. Which strategy a participant ultimately used may have depended on a range of factors other than just the perceptual similarity between games. For example, it requires a degree of exploration to realise that more than one point can be scored on a level. If some participants viewed the cost of an exploratory interaction as lower than other participants then they may have been more likely to employ a 'one-shot' strategy. Future work should therefore aim to constrain the games further so that only one strategy is present. For instance, the final test level could consist of a single trial

so that a ‘one-shot’ strategy is unavailable. The effect of perceptual similarity on transfer ability could then be investigated with respect to a ‘zero-shot’ strategy rather than a mixture of strategies.

Another complexity with exploring behaviour on more naturalistic tasks is the increase in cognitive resources required to complete them. With increased task demands comes the need for the allocation of cognitive resources. As we have seen, this process is often referred to as ‘attention’ and is commonly prescribed to the Pre-Frontal Cortex (PFC) (Miller and Cohen, 2001). The recruitment of attentional processes can greatly increase the complexity of behavioural analysis because it can be unclear what elements of the task participants have processed and incorporated into their decisions. For example, in our games participants may have chosen to focus on the motor controls for the first few training games rather than on the perceptual features of the objects. This is an issue because it means that the degree of perceptual similarity between training games is lost for those games. In addition, participants with high performance on the first motor control task may have had to allocate fewer cognitive resources to using the controls compared to those with poor motor control performance. As a result, they may have had more cognitive resources to focus on the perceptual features of objects and learn the underlying relational rules for transfer. This may explain why some participants were able to learn a ‘zero-shot’ strategy, while others learnt a less cognitively demanding ‘one-shot’ strategy. Future work should therefore try to understand what elements of the task participants are focusing on over the course of training and how this is effected by cognitive load. This will require the use of additional techniques such as the combination of eye-tracking and computational modelling (see Leong et al., 2017).

8.5 Lessons For The Future

Throughout the course of this thesis many practical lessons have been learnt that will be useful for future research endeavors. This section outlines some of these lessons in the hope that others can also learn from them.

8.5.1 Starting With Simple Problems

From the perspective of computational modelling, one of the key lessons has been the importance of prototyping ideas using simple problems. By starting with simple problems, such as the Grid Worlds in Chapter 4 or the multiple choice task in Chapter 6, one gains several benefits over starting with complex tasks, such as video games. The first benefit is that the computational cost of running simulations is greatly reduced, which invariably means that simulations take less time to run. This is particularly pertinent in the domain of Deep Reinforcement Learning (RL) where, as we have seen in this thesis, approaches require large amounts of data to converge to a solution. Ultimately, research is an iterative process and so reduced computational cost helps to improve research efficiency. For example, given a set amount of time and computational resources, reduced computational cost allows one to explore more architectural ideas, parameter settings, random seeds and task variations. In addition to reduced computational cost, simpler tasks are also less costly to design and implement from a human perspective. This reduced human cost also greatly improves the efficiency of the iterative research process. That said, the increasing prevalence of open source software containing more complex tasks (e.g. OpenAI Gym, Brockman et al., 2016) is helping to negate this human cost by providing a simple Application Programming Interface (API) for researchers.

Reduced computational and human costs aside, simpler tasks also have the added benefit of being more interpretable than complex tasks. By reducing the complexity of the task, the number of possible strategies that can be learnt in order to solve the problem are reduced. In the case of RL, this makes it easier to understand the final policy learnt by the agent and allows for the investigation of specific behaviours. In addition, simple tasks often involve low-dimensional state representations, which are significantly easier to interpret. For example, in the case of the Grid Worlds in Chapter 4 the state representation was the position of the agent in a 2-dimensional grid. This meant that we could visually inspect the states encoded by the Self-Organising Map (SOM) because they corresponded to positions in space. Importantly, this helped us to understand how the SOM encode violations of the generalizations made by the Deep Neural Network (DNN). Similarly, with the multiple choice task in Chapter 6, we could analyse the features produced

by individual images and investigate how the properties of those features affected the performance of the agent. This would have been much more difficult in the subsequent video games because they involved complex sequences of images.

Experimental psychology tasks also take advantage of the increased interpretability of simple task designs. This is important because the results of computational studies can be compared to their experimental counterparts when simple tasks are used. For example, by using grid worlds in Chapter 4 we could compare the positions encoded by the SOM to those encoded by the hippocampus of rodents during maze tasks. Similarly, we know that for the multiple choice task in Chapter 6, our aim was to capture the kind of ‘one-shot’ learning demonstrated by humans and animals in similar tasks (see for example Experiment 1 in Chapter 7). In comparison, the behaviour of humans and animals on more complex tasks, such as the cart-pole task in Chapter 4 or the object collection game in Chapter 6, is much more poorly characterised and so the target behaviour is less clear. Indeed, the results of Experiments 2 and 3 in Chapter 7 serve to show the wide range of strategies that humans can learn in more complex and naturalistic tasks.

Another benefit of starting with simple problems is that it also allows for simpler computational models. This is particularly true for Deep RL where one of the biggest challenges is learning useful state representations. By using a simple task, the state representation can be provided in a more direct form such as the x and y position in a Grid World, as opposed to a more complex form such as the pixel values of a frame from a video game. By reducing the complexity of the state representation, one can focus on aspects of the approach other than representation learning. For example, in the case of *Complementary Temporal Difference Learning (CTDL)* in Chapter 4, we wanted to focus on the interactions between the DNN and the SOM. By using tasks that provided useful low-dimensional state representations we could feed the states directly to the SOM. However, for more complex tasks such as using raw pixels as input, the SOM would require latent representations to deal with the high-dimensionality of the input. We were therefore able to circumvent this representation problem and instead highlighted it as a key avenue for future research.

While these benefits of simple tasks all help to identify interesting computational ideas, the ability to scale up to complex tasks is still of importance for understanding

intelligence. This is particularly true in the domain of Deep RL, which primarily concerns itself with mapping high-dimensional and continuous states to actions based on reward feedback. It can then therefore be argued that simple tasks with simple state representations fail to address the main benefit of a Deep RL approach. Nevertheless, even if a computational idea does not scale up to a more complex task the results can still be highly informative. For example, in the case of SPA in Chapter 6, initial tests suggested that the approach would not scale up to the object collection game. However by comparing the multiple choice task to the object collection game we were able to identify that several time steps were needed to efficiently evaluate a particular hypothesis. As a result, we updated the attention vector periodically and this turned out to be critical for scaling up SPA to the object collection game. This not only taught us something about the nature of our approach but the results of the simpler multiple choice task also gave us the conviction to persevere with our approach.

8.5.2 The Intersection between Artificial Intelligence and Cognitive Science

By forming an analogy between Deep Reinforcement Learning (RL) and the brain, this thesis has operated at the intersection between Artificial Intelligence (AI) and cognitive science. While an extremely rewarding inter-disciplinary approach, it has become apparent that these two fields tend to have fundamentally different goals. In particular, research in AI often puts more emphasis on state-of-the-art performance whereas cognitive science focuses on understanding the processes underlying performance. From an AI perspective, this can often lead to using cognitive science as an inspiration for new ideas but then adding non-cognitive components to further improve performance. Within cognitive science, this is often referred to as over-engineering, whereby design decisions are made with the sole purpose of solving the problem at hand. For example, it can be tempting to hand-code heuristics into a model or provide the model with information that a human learner may not have access to in order to improve performance. While these interventions may lead to better results they can be detrimental to the models ability to tell us something about cognition. Fundamentally, the aim of this thesis was to provide insights into

cognition and so over-engineering was an issue that needed to be avoided.

One obvious remedy to the issue of over-engineering is to ask yourself for each design decision whether a cognitive analogue exists in the brain. However, some decisions are not always this clear cut because our knowledge of cognition is imperfect and so practical assumptions have to be made. When this is the case, one solution is to assess the computational model on a range of tasks with different properties. If over-engineering has occurred, then the approach is likely to perform well on one task but poorly on others. Indeed, cognitive science is often interested in computational principles that are domain-general rather than hyper-specialised to a specific task. For example, in Chapter 4, we assessed CTDL on tasks that ranged from fine motor control (Cart-Pole) to spatial navigation (Grid Worlds), and that involved a combination of discrete and continuous state and action spaces. The fact that CTDL performed well on this range of tasks without the need for architectural changes suggests that the amount of over-engineering was small and that it represents a domain-general mechanism. That said, some variation in performance on different tasks can be informative. For instance, the fact that CTDL showed reduced benefits on the Cart-Pole task was useful for understanding that the SOM was most beneficial for tasks that involved rare, highly salient events.

The risk of over-engineering can also occur during the selection of hyper-parameter values. More specifically, a high sensitivity to hyper-parameters values can often be a sign of over-engineering, especially if the values have to be re-tuned for each task. That said, sometimes the tuning of hyper-parameter values can also be informative. For example, in the case of CTDL in Chapter 4, an informal parameter search assigned a high learning rate to the updating of values stored in the SOM. This was interesting because it highlighted the functional importance of fast learning in a hippocampal learning system.

Despite cognitive science appearing to be restricted by biological plausibility and less concerned with state-of-the-art performance, we believe that the field of AI should take more inspiration from a cognitive science approach. The investigation of domain-general principles that underlie general intelligence is a fundamental goal of AI research and an over-emphasis on state-of-the-art results on specific tasks is a potential barrier to this goal. With a more cognitive science approach, the AI field

can reward research that does not claim state-of-the-art results but instead focuses on detailing fundamental mechanisms underlying intelligence. This may mean using simpler tasks for increased interpretability (see the previous section), or removing specific components of a model (e.g. Chapter 4) in order to more concretely understand its behaviour. In the long-term this will lead to greater innovation and open more research avenues than simply tuning models to perform better on a single task by a few percentage points. Our hope is that the work in this thesis stands as a testament to this and demonstrates that both AI and cognitive science researchers can benefit from such research. While neither CTDL nor SPA claim state-of-the-art results on the tasks that they are evaluated on, they do describe domain general principles that both tell us something about cognition and simultaneously improve upon canonical AI approaches. Unfortunately, presenting such work to an interdisciplinary audience can often be difficult as it relies on AI techniques, which are not always widely accepted in cognitive science, and does not claim state-of-the-art results, which is a revered property in current AI research.

8.5.3 The Importance of Developmental Studies

Throughout this thesis we have focused on the ability of humans to perform efficient Reinforcement Learning (RL), typically with a focus on adult performance on tasks that operate over short time-scales. However, the importance of considering infant development when trying to understand efficient RL in adults has become increasingly apparent over the course of this thesis. We believe more work needs to be done to reconcile Deep RL with developmental psychology and that infants represent ideal subjects for investigating efficient RL for a multitude of reasons.

One of the largest differences between adults and infants is that infants enter a task with much less prior knowledge compared to adults. This is important because it means that infants are less likely to transfer knowledge from outside of the task setting. This makes it easier to control for the knowledge that is being transferred because one can be more confident that it is from the task itself. For example, trying to understand what is being transferred between different video games is difficult in adults when they already transfer vast amounts of knowledge from outside of the video games (Dubey et al., 2018).

The fact that infants have less prior knowledge also makes them more comparable to Deep RL algorithms, which also start off with limited prior knowledge. For example, Deep RL algorithms that use Deep Convolutional Neural Networks (DCNNs) to learn policies based on raw pixel inputs are effectively learning to ‘see’ for the first time. This is one of the first tasks faced by a new born infant. Indeed, this is why in Chapter 6, we focus on trying to learn a task based on a set of pre-learned visual features, as we believe this more faithfully replicates the problem faced by an adult learner when presented with a new task.

Another benefit of young children’s reduced prior knowledge is that it makes them useful for understanding whether a behaviour is learnt or innate. This is particularly important for Deep RL because it has often been criticised for lacking the necessary developmental ‘start-up software’ to be data efficient (Lake et al., 2017). For example, it has been proposed that children have the ability to reason about basic physics from a young age. However, it is still under debate how much either innate wiring in the brain or learning after birth contributes to this ability. Understanding their contributions is vital for Deep RL because it can guide researchers towards approaches that either have intuitive physics built in or that learn it from interactions with the world in a similar way to infants. The ability to perform intuitive physics in a Deep RL system is of high importance because it is likely to be a crucial component of model-based RL.

The development of infants also highlights the importance of considering different time-scales during the study of RL. Most research on Deep RL concerns itself with performing a task that in the real world would be performed for a matter of minutes or hours. However, very little work in Deep RL concerns itself with trying to capture how behaviours emerge over the course of many years. There are obviously inherent difficulties in modelling such a long-term process as the complexity of experiences increases dramatically. However, one notable example of how RL models have been used to model developmental processes is in the emergence of eye-gaze following (Triesch et al., 2006; Blakeman and Mareschal, 2017). This skill is supposed to emerge in infants over several months and helps adults to indicate objects of interest in the environment. This helps the infant to learn about its environment and is a pre-cursor to more complex forms of communication such as pointing and request

behaviour. Crucially, RL models have been proposed that are able to explain the natural emergence of eye-gaze following through learning, which suggests that it is not an innate ability. This highlights the importance of developmental RL work because humans develop complex skills in a compound and ordered way such that learning bootstraps further learning. Research into curriculum learning is beginning to touch on this topic and will be extremely informative for understanding how knowledge is consolidated and transferred between experiences.

Deep RL aside, the results of the empirical work in this thesis also highlight the importance of considering cognitive development. Ultimately the video games we explored involved learning and exploiting relational information. Critically, it has been suggested that the ability to reason about relations develops during childhood and that it bootstraps further relational learning (Gentner and Hoyos, 2017). This is important because it suggests that maybe the perceptual similarity between our video games may have had more of an effect on children. This is because their ability to reason about relations is less established and may be more susceptible to environmental factors.

8.5.4 Proposing a New Experimental Paradigm

The empirical work in this thesis involved the proposal of a new experimental paradigm for exploring the effect of perceptual similarity on transfer. The novel task involved a series of video games that reflected several hallmarks of naturalistic behaviour: continuous sequential decision-making and fine motor control. The process of proposing a new experimental paradigm as opposed to using a well-established paradigm has provided several important lessons.

Firstly, it has provided an appreciation for the number of free parameters in an experimental design that can potentially influence the effect of interest. For example, the number of video games, the duration of each video game, the nature of the instructions provided, the number of different object features and the amount of monetary reward (etc.) could all have an impact on how the training regime affects transfer performance. Typically, in well established experimental paradigms, the values of such variables are standardised based on previous studies so that researchers know the expected effects. Furthermore, they also have a rough estimate

of the effect size. In our case, we had very little idea of the expected effects of perceptual similarity on transfer ability, which made it difficult to perform a power analysis in order to calculate the necessary number of participants.

Another source of uncertainty in proposing a new experimental paradigm is from the lack of directly relevant past literature. Well established paradigms have a host of past data that can be used to form predictions about a proposed manipulation. In contrast, the relevant past literature for our study spanned many different domains from category learning to analogical reasoning. This made it difficult to formulate strong predictions and meant that we had to take a more exploratory approach. Nevertheless, it is these differences from the past literature that make the proposed experimental paradigm of interest to researchers. There is therefore a fine balance to be made between the degree of novelty and the strength of the foundations provided by past research.

8.6 Concluding Remarks

The focus of this thesis has been on understanding how rapid learning and the transfer of past knowledge can improve the efficiency of Reinforcement Learning (RL) in the brain. To help inform this investigation we have used Deep RL as an analogy to the brain. This has allowed us to explore the key computational properties of different learning systems in the brain that support efficient RL. In particular, we have taken inspiration from Complementary Learning Systems (CLS) theory and highlighted four learning systems that we believe to be particularly important for efficient RL; sensory cortices, the Pre-Frontal Cortex (PFC), the hippocampus and the striatum. Upon reviewing recent work involving these different learning systems, we proposed two novel Deep RL algorithms that aim to capture two of the key mechanisms needed for efficient RL in the brain: episodic memory and selective attention.

In addition to this computational work, we also explored how the degree of perceptual similarity between consecutive experiences can affect people's ability to transfer knowledge. We used a 2D video game as our experimental paradigm because it consists of many of the properties found in everyday tasks: sequential decision

making and fine motor control. We found that the degree of perceptual similarity between consecutive experiences had no effect on an adult’s ability to perform transfer. However, when participants were told the rules of the game before-hand, those who were trained on consecutive levels with a high degree of perceptual similarity demonstrated improved transfer. This suggests that perceptual similarity has an impact on the ability of adults to utilise explicit knowledge for transfer.

We believe that these empirical findings can be used to constrain computational theories of transfer and inform the analogy between the brain and Deep RL. For example, the Deep Neural Networks (DNNs) used in Deep RL are susceptible to spurious perceptual similarities in their input during training. This predicts that the low perceptual similarity condition should improve transfer in the 2D video game because it reduces the correlations between perceptual features. The lack of an effect of perceptual similarity on transfer when participants were not told the rules of the game therefore highlights how the learning of DNNs differs from that of human learning. Conversely, many of the participants demonstrated evidence of ‘one-shot’ learning as predicted by a Deep RL model of the PFC known as meta-RL. The fact that participants were better at transfer in the high perceptual similarity condition when they were told the rules of the games before-hand also has interesting implications for the analogy between the brain and Deep RL. In particular, it highlights that any mechanism for explicitly providing information to Deep RL models should account for the fact that transfer performance improves when the degree of perceptual similarity between consecutive experiences is high.

We hope that the analogy between the brain and Deep RL outlined in this thesis can serve as a basis for further research into efficient RL. In order to reconcile the differences in efficiency between the brain and Deep RL, we emphasise the need for Deep RL approaches that include learning systems that represent the key computational properties of the hippocampus and the PFC. Future work should also address our empirical findings, as humans appear to be able to circumvent, or even utilise, the perceptual similarity between experiences to perform transfer.

Acronyms

A2C	Advantage Actor-Critic.
AC	Attention at Choice.
ACL	Attention at Choice and Learning.
AI	Artificial Intelligence.
AL	Attention at Learning.
ANNs	Artificial Neural Networks.
CLS	Complementary Learning Systems.
CTDL	Complementary Temporal Difference Learning.
DA	Dopamine.
DARLA	Disentangled Representation Learning Agent.
DBNs	Deep Belief Networks.
DCNN	Deep Convolutional Neural Network.
DNNs	Deep Neural Networks.
DQN	Deep Q-Network.
fMRI	functional Magnetic Resonance Imaging.
LEC	Lateral Entorhinal Cortex.
LSTM	Long Short-Term Memory.
MEC	Medial Entorhinal Cortex.

PFC	Pre-Frontal Cortex.
RBM	Restricted Boltzmann Machine.
REMERGE	Recurrency and Episodic Memory Results in Generalization.
RL	Reinforcement Learning.
RPEs	Reward-Prediction Errors.
SOM	Self-Organising Map.
SPA	Selective Particle Attention.
SR	Successor Representation.
SUSTAIN	Supervised and Unsupervised STRatified Adaptive Incremental Network.
TD	Temporal Difference.
TEM	Tolman-Eichenbaum Machine.
UA	Uniform Attention.
VAE	Variational AutoEncoder.

Chapter 9

Bibliography

- Adams, C. D. (1982). Variations in the sensitivity of instrumental responding to reinforcer devaluation. *The Quarterly Journal of Experimental Psychology Section B*, 34(2b):77–98.
- Adcock, R. A., Thangavel, A., Whitfield-gabrieli, S., Knutson, B., and Gabrieli, J. D. E. (2006). Reward-Motivated Learning : Mesolimbic Activation Precedes Memory Formation. *Neuron*, 50(3):507–517.
- Addis, D. R., Wong, A. T., and Schacter, D. L. (2007). Remembering the past and imagining the future: common and distinct neural substrates during event construction and elaboration. *Neuropsychologia*, 45(7):1363–1377.
- Aerts, H., Fias, W., Caeyenberghs, K., and Marinazzo, D. (2016). Brain networks under attack: robustness properties and the impact of lesions. *Brain*, 139(12):3063–3083.
- Alho, K., Kujala, T., Paavilainen, P., Summala, H., and Näätänen, R. (1993). Auditory processing in visual brain areas of the early blind: evidence from event-related potentials. *Electroencephalography and clinical neurophysiology*, 86(6):418–427.
- Alvernhe, A., Save, E., and Poucet, B. (2011). Local remapping of place cell firing in the tolmán detour task. *European Journal of Neuroscience*, 33(9):1696–1705.
- Amorim Da Costa, N. M. M. and Martin, K. (2010). Whose cortical column would that be? *Frontiers in neuroanatomy*, 4:16.

- Anderson, D. R., Choi, H. P., and Lorch, E. P. (1987). Attentional inertia reduces distractibility during young children’s tv viewing. *Child Development*, pages 798–806.
- Aronov, D., Nevers, R., and Tank, D. W. (2017). Mapping of a non-spatial dimension by the hippocampal–entorhinal circuit. *Nature*, 543(7647):719–722.
- Ashby, F. G. and Maddox, W. T. (2005). Human category learning. *Annu. Rev. Psychol.*, 56:149–178.
- Ballard, I. C., Wagner, A. D., and McClure, S. M. (2019). Hippocampal pattern separation supports reinforcement learning. *Nature Communications*, 10(1).
- Baluch, F. and Itti, L. (2011). Mechanisms of top-down attention. *Trends in neurosciences*, 34(4):210–224.
- Barraclough, D. J., Conroy, M. L., and Lee, D. (2004). Prefrontal cortex and decision making in a mixed-strategy game. *Nature neuroscience*, 7(4):404–410.
- Bayer, H. M. and Glimcher, P. W. (2005). Midbrain dopamine neurons encode a quantitative reward prediction error signal. *Neuron*, 47(1):129–141.
- Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. (2013). The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279.
- Bengio, Y. (2012). Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 17–36.
- Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.
- Benna, M. K. and Fusi, S. (2016). Computational principles of synaptic memory consolidation. *Nature neuroscience*, 19(12):1697–1706.
- Bichot, N. P., Heard, M. T., DeGennaro, E. M., and Desimone, R. (2015). A source for feature-based attention in the prefrontal cortex. *Neuron*, 88(4):832–844.

- Biederman, I. and Shiffrar, M. M. (1987). Sexing day-old chicks: A case study and expert systems analysis of a difficult perceptual-learning task. *Journal of Experimental Psychology: Learning, memory, and cognition*, 13(4):640.
- Birrell, J. M. and Brown, V. J. (2000). Medial frontal cortex mediates perceptual attentional set shifting in the rat. *Journal of Neuroscience*, 20(11):4320–4324.
- Bizley, J. K. and King, A. J. (2008). Visual–auditory spatial processing in auditory cortical neurons. *Brain research*, 1242:24–36.
- Bizley, J. K., Nodal, F. R., Bajo, V. M., Nelken, I., and King, A. J. (2007). Physiological and anatomical evidence for multisensory interactions in auditory cortex. *Cerebral cortex*, 17(9):2172–2189.
- Blakeman, S. and Mareschal, D. (2017). Narrowing of the cone-of-direct gaze through reinforcement learning. In *CogSci*.
- Blakeman, S. and Mareschal, D. (2020a). A complementary learning systems approach to temporal difference learning. *Neural Networks*, 122:218–230.
- Blakeman, S. and Mareschal, D. (2020b). Selective particle attention: Visual feature-based attention in deep reinforcement learning. *arXiv preprint arXiv:2008.11491*.
- Bloom, L. and Lahey, M. (1978). Language development and language disorders.
- Blundell, C., Uria, B., Pritzel, A., Li, Y., Ruderman, A., Leibo, J. Z., Rae, J., Wierstra, D., and Hassabis, D. (2016). Model-free episodic control. *arXiv preprint arXiv:1606.04460*.
- Botvinick, M., Ritter, S., Wang, J. X., Kurth-Nelson, Z., Blundell, C., and Hassabis, D. (2019). Reinforcement learning, fast and slow. *Trends in cognitive sciences*, 23(5):408–422.
- Bramlage, L. and Cortese, A. (2020). Attention or memory? neurointerpretable agents in space and time. *arXiv preprint arXiv:2007.04862*.
- Bray, S. and Doherty, J. O. (2007). Neural Coding of Reward-Prediction Error Signals During Classical Conditioning With Attractive Faces. *Journal of Neurophysiology*, 97(4):3036–3045.

- Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). OpenAI Gym. *arXiv*, pages 1–4.
- Brown, A. L. (1990). Domain-specific principles affect learning and transfer in children. *Cognitive science*, 14(1):107–133.
- Brown, R. W. (1957). Linguistic determinism and the part of speech. *The Journal of Abnormal and Social Psychology*, 55(1):1.
- Burgess, N., Maguire, E. A., and O’Keefe, J. (2002). The human hippocampus and spatial and episodic memory. *Neuron*, 35(4):625–641.
- Burns, J. J. and Anderson, D. R. (1993). Attentional inertia and recognition memory in adult television viewing. *Communication Research*, 20(6):777–799.
- Cadiou, C. F., Hong, H., Yamins, D. L., Pinto, N., Ardila, D., Solomon, E. A., Majaj, N. J., and DiCarlo, J. J. (2014). Deep neural networks rival the representation of primate it cortex for core visual object recognition. *PLoS Comput Biol*, 10(12):e1003963.
- Carey, S. and Bartlett, E. (1978). Acquiring a single new word.
- Carvalho, P. F. and Goldstone, R. L. (2014a). Effects of interleaved and blocked study on delayed test of category learning generalization. *Frontiers in psychology*, 5:936.
- Carvalho, P. F. and Goldstone, R. L. (2014b). Putting category learning in order: Category structure and temporal arrangement affect the benefit of interleaved over blocked study. *Memory & cognition*, 42(3):481–495.
- Cash, S. and Yuste, R. (1999). Linear summation of excitatory inputs by cal pyramidal neurons. *Neuron*, 22(2):383–394.
- Casler, K. and Kelemen, D. (2005). Young children’s rapid learning about artifacts. *Developmental Science*, 8(6):472–480.
- Chang, C. Y., Gardner, M., Di Tillio, M. G., and Schoenbaum, G. (2017). Optogenetic blockade of dopamine transients prevents learning induced by changes in reward features. *Current Biology*, 27(22):3480–3486.

- Chen, Z. and Siegler, R. S. (2013). Young children’s analogical problem solving: Gaining insights from video displays. *Journal of experimental child psychology*, 116(4):904–913.
- Childers, J. B. and Tomasello, M. (2003). Children extend both words and non-verbal actions to novel exemplars. *Developmental Science*, 6(2):185–190.
- Choung, O.-h., Lee, S. W., and Jeong, Y. (2017). Exploring feature dimensions to learn a new policy in an uninformed reinforcement learning task. *Scientific reports*, 7(1):1–12.
- Cohen, J. D. and O’Reilly, R. C. (1996). A preliminary theory of the interactions between prefrontal cortex and hippocampus that contribute to planning and prospective memory. *Prospective memory: Theory and applications*, pages 267–295.
- Constantinescu, A. O., O’Reilly, J. X., and Behrens, T. E. (2016). Organizing conceptual knowledge in humans with a gridlike code. *Science*, 352(6292):1464–1468.
- Cook, R. G. and Wasserman, E. A. (2007). Learning and transfer of relational matching-to-sample by pigeons. *Psychonomic Bulletin & Review*, 14(6):1107–1114.
- Daw, N. D., Niv, Y., and Dayan, P. (2005). Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nature neuroscience*, 8(12):1704–1711.
- Dayan, P. (1993). Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624.
- Dayan, P., Hinton, G. E., Neal, R. M., and Zemel, R. S. (1995). The helmholtz machine. *Neural computation*, 7(5):889–904.
- Dayan, P., Kakade, S., and Montague, P. R. (2000). Learning and selective attention. *Nature neuroscience*, 3(11):1218–1223.

- Debiec, J., Ledoux, J. E., and Nader, K. (2002). Cellular and Systems Reconsolidation in the Hippocampus. *Neuron*, 36:527–538.
- DeFelipe, J., Markram, H., and Rockland, K. S. (2012). The neocortical column. *Frontiers in Neuroanatomy*, 6:22.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.
- Desimone, R. and Duncan, J. (1995). Neural mechanisms of selective visual attention. *Annual review of neuroscience*, 18(1):193–222.
- Dias, R., Robbins, T., and Roberts, A. (1996). Dissociation in prefrontal cortex of affective and attentional shifts. *Nature*, 380(6569):69–72.
- Dobbins, I. G., Foley, H., Schacter, D. L., and Wagner, A. D. (2002). Executive control during episodic retrieval: multiple prefrontal processes subserve source memory. *Neuron*, 35(5):989–996.
- Doherty, J. P. O., Dayan, P., Friston, K., Critchley, H., and Dolan, R. J. (2003). Temporal Difference Models and Reward-Related Learning in the Human Brain. *Neuron*, 38(2):329–337.
- Dong, Z., Bai, Y., Wu, X., Li, H., Gong, B., Howland, J. G., Huang, Y., He, W., Li, T., and Wang, Y. T. (2013). Neuropharmacology Hippocampal long-term depression mediates spatial reversal learning in the Morris water maze. *Neuropharmacology*, 64:65–73.
- Douglas, R. J., Martin, K. A., and Whitteridge, D. (1989). A canonical microcircuit for neocortex. *Neural computation*, 1(4):480–488.
- Dragoi, G. and Tonegawa, S. (2011). Preplay of future place cell sequences by hippocampal cellular assemblies. *Nature*, 469(7330):397–401.
- Dragoi, G. and Tonegawa, S. (2013). Distinct preplay of multiple novel spatial experiences in the rat. *Proceedings of the National Academy of Sciences*, 110(22):9100–9105.

- Dubey, R., Agrawal, P., Pathak, D., Griffiths, T. L., and Efros, A. A. (2018). Investigating human priors for playing video games. *arXiv preprint arXiv:1802.10217*.
- Duncan, K., Doll, B. B., Daw, N. D., and Shohamy, D. (2018). More Than the Sum of Its Parts : A Role for the Hippocampus in Configural Reinforcement Learning. *Neuron*, 98(3):645–657.
- Džeroski, S., De Raedt, L., and Driessens, K. (2001). Relational reinforcement learning. *Machine learning*, 43(1-2):7–52.
- Eichenbaum, H. (2017). Prefrontal–hippocampal interactions in episodic memory. *Nature Reviews Neuroscience*, 18(9):547–558.
- Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99.
- Esber, G. R. and Haselgrove, M. (2011). Reconciling the influence of predictiveness and uncertainty on stimulus salience: a model of attention in associative learning. *Proceedings of the Royal Society B: Biological Sciences*, 278(1718):2553–2561.
- Exton-mcguinness, M. T. J., Lee, J. L. C., and Reichelt, A. C. (2015). Updating memories — The role of prediction errors in memory reconsolidation. *Behavioural Brain Research*, 278:375–384.
- Farashahi, S., Rowe, K., Aslami, Z., Lee, D., and Soltani, A. (2017). Feature-based learning improves adaptability without compromising precision. *Nature communications*, 8(1):1–16.
- Fei-Fei, L., Fergus, R., and Perona, P. (2006). One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611.
- Finney, E. M., Fine, I., and Dobkins, K. R. (2001). Visual stimuli activate auditory cortex in the deaf. *Nature neuroscience*, 4(12):1171–1173.
- FitzGerald, T. H., Seymour, B., and Dolan, R. J. (2009). The role of human orbitofrontal cortex in value comparison for incommensurable objects. *Journal of Neuroscience*, 29(26):8388–8395.

- Flesch, T., Balaguer, J., Dekker, R., Nili, H., and Summerfield, C. (2018). Comparing continual task learning in minds and machines. *Proceedings of the National Academy of Sciences*, 115(44):E10313–E10322.
- Foster, D. J. and Wilson, M. A. (2006). Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature*, 440(7084):680–683.
- François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., and Pineau, J. (2018). An introduction to deep reinforcement learning. *arXiv preprint arXiv:1811.12560*.
- French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.
- Gallahue, D. L., Ozmun, J. C., and Goodway, J. (2006). *Understanding motor development: Infants, children, adolescents, adults*. McGraw-hill Boston, MA.
- Garg, A. K., Li, P., Rashid, M. S., and Callaway, E. M. (2019). Color and orientation are jointly coded and spatially organized in primate primary visual cortex. *Science*, 364(6447):1275–1279.
- Gauthier, J. L. and Tank, D. W. (2018). A dedicated population for reward coding in the hippocampus. *Neuron*, 99(1):179–193.
- Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive science*, 7(2):155–170.
- Gentner, D. (1988). Metaphor as structure mapping: The relational shift. *Child development*, pages 47–59.
- Gentner, D. and Hoyos, C. (2017). Analogy and abstraction. *Topics in cognitive science*, 9(3):672–693.
- Gentner, D., Loewenstein, J., and Hung, B. (2007). Comparison facilitates children’s learning of names for parts. *Journal of Cognition and Development*, 8(3):285–307.
- Gentner, D. and Markman, A. B. (1994). Structural alignment in comparison: No difference without similarity. *Psychological science*, 5(3):152–158.

- Gentner, D. and Rattermann, M. J. (1991). Language and the career of similarity. *Perspectives on language and thought: Interrelations in development*, 225.
- Gentner, D., Rattermann, M. J., and Forbus, K. D. (1993). The roles of similarity in transfer: Separating retrievability from inferential soundness. *Cognitive psychology*, 25(4):524–575.
- Gershman, S. J. and Daw, N. D. (2017). Reinforcement learning and episodic memory in humans and animals: an integrative framework. *Annual review of psychology*, 68:101–128.
- Gershman, S. J. and Ölveczky, B. P. (2020). The neurobiology of deep reinforcement learning. *Current Biology*, 30(11):R629–R632.
- Gibson, E. J. (1969). Principles of perceptual learning and development.
- Gick, M. L. and Holyoak, K. J. (1980). Analogical problem solving.
- Goldstone, R. L. (1996). Isolated and interrelated concepts. *Memory & cognition*, 24(5):608–628.
- Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT press Cambridge.
- Graves, A., Wayne, G., and Danihelka, I. (2014). Neural turing machines. *arXiv preprint arXiv:1410.5401*.
- Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., Colmenarejo, S. G., Grefenstette, E., Ramalho, T., Agapiou, J., et al. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476.
- Grill-Spector, K., Weiner, K. S., Gomez, J., Stigliani, A., and Natu, V. S. (2018). The functional neuroanatomy of face perception: from brain measurements to deep neural networks. *Interface Focus*, 8(4):20180013.
- Groenewegen, H., Vermeulen-Van der Zee, E., te Kortschot, A., and Witter, M. (1987). Organization of the projections from the subiculum to the ventral stri-

- tum in the rat. A study using anterograde transport of Phaseolus vulgaris leucoagglutinin. *Neuroscience*, 23(1):103–120.
- Güçlü, U. and van Gerven, M. A. (2015). Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *Journal of Neuroscience*, 35(27):10005–10014.
- Guerguiev, J., Lillicrap, T. P., and Richards, B. A. (2017). Towards deep learning with segregated dendrites. *ELife*, 6:e22901.
- Gupta, A. S., van der Meer, M. A., Touretzky, D. S., and Redish, A. D. (2010). Hippocampal replay is not a simple function of experience. *Neuron*, 65(5):695–705.
- Haber, S. N. (2016). Corticostriatal circuitry. *Dialogues in clinical neuroscience*, 18(1):7.
- Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., and Moser, E. I. (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–806.
- Harlow, H. F. (1949). The formation of learning sets. *Psychological review*, 56(1):51.
- Hassabis, D., Kumaran, D., Vann, S. D., and Maguire, E. A. (2007). Patients with hippocampal amnesia cannot imagine new experiences. *Proceedings of the National Academy of Sciences*, 104(5):1726–1731.
- Higgins, I., Amos, D., Pfau, D., Racaniere, S., Matthey, L., Rezende, D., and Lerchner, A. (2018). Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*.
- Higgins, I., Chang, L., Langston, V., Hassabis, D., Summerfield, C., Tsao, D., and Botvinick, M. (2020). Unsupervised deep learning identifies semantic disentanglement in single inferotemporal neurons. *arXiv preprint arXiv:2006.14304*.
- Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2016). beta-vae: Learning basic visual concepts with a constrained variational framework.

- Higgins, I., Pal, A., Rusu, A. A., Matthey, L., Burgess, C. P., Pritzel, A., Botvinick, M., Blundell, C., and Lerchner, A. (2017). Darla: Improving zero-shot transfer in reinforcement learning. *arXiv preprint arXiv:1707.08475*.
- Hinton, G. E., Dayan, P., Frey, B. J., and Neal, R. M. (1995). The “wake-sleep” algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161.
- Hollerman, J. R. and Schultz, W. (1998). Dopamine neurons report an error in the temporal prediction of reward during learning. *Nature neuroscience*, 1(4):304–309.
- Hollup, S. A., Molden, S., Donnett, J. G., Moser, M.-B., and Moser, E. I. (2001). Accumulation of hippocampal place fields at the goal location in an annular water-maze task. *Journal of Neuroscience*, 21(5):1635–1644.
- Holyoak, K. J. (2012). Analogy and relational reasoning.
- Hoskin, A. N., Bornstein, A. M., Norman, K. A., and Cohen, J. D. (2019). Refresh my memory: Episodic memory reinstatements intrude on working memory maintenance. *Cognitive, Affective, & Behavioral Neuroscience*, 19(2):338–354.
- Houk, J. C., Adams, J. L., and Barto, A. G. (1995). A Model of How the Basal Ganglia Generate and Use Neural Signals that Predict Reinforcement. *Computational Neuroscience. Models of Information Processing in the Basal Ganglia*, pages 249–270.
- Høydal, Ø. A., Skytøen, E. R., Andersson, S. O., Moser, M.-B., and Moser, E. I. (2019). Object-vector coding in the medial entorhinal cortex. *Nature*, 568(7752):400–404.
- Igata, H., Ikegaya, Y., and Sasaki, T. (2020). Prioritized experience replays on a hippocampal predictive map for learning. *bioRxiv*.
- Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., and Kavukcuoglu, K. (2016). Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*.
- Jaeger, H. (2016). Deep neural reasoning. *Nature*, 538(7626):467–468.

- Jang, A. I., Nassar, M. R., Dillon, D. G., and Frank, M. J. (2018). Positive reward prediction errors strengthen incidental memory encoding. *bioRxiv*.
- Ji, D. and Wilson, M. A. (2007). Coordinated memory replay in the visual cortex and hippocampus during sleep. *Nature neuroscience*, 10(1):100–107.
- Joel, D., Niv, Y., and Ruppin, E. (2002). Actor–critic models of the basal ganglia: New anatomical and computational perspectives. *Neural networks*, 15(4-6):535–547.
- Johnson, A. and Redish, D. A. (2007). Neural Ensembles in CA3 Transiently Encode Paths Forward of the Animal at a Decision Point. *Journal of Neuroscience*, 27(45):12176–12189.
- Jones, M. and Canas, F. (2010). Integrating reinforcement learning with models of representation learning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 32.
- Kamin, L. J. (1967). Predictability, surprise, attention, and conditioning.
- Karlsson, M. P. and Frank, L. M. (2009). Awake replay of remote experiences in the hippocampus. *Nature neuroscience*, 12(7):913–918.
- Kayser, C., Petkov, C. I., Augath, M., and Logothetis, N. K. (2007). Functional imaging reveals visual modulation of specific fields in auditory cortex. *Journal of Neuroscience*, 27(8):1824–1835.
- Kell, A. J., Yamins, D. L., Shook, E. N., Norman-Haignere, S. V., and McDermott, J. H. (2018). A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98(3):630–644.
- Kempka, M., Wydmuch, M., Runc, G., Toczek, J., and Ja, W. (2016). ViZDoom : A Doom-based AI Research Platform for Visual Reinforcement Learning. *arXiv*.
- Khaligh-Razavi, S.-M. and Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain it cortical representation. *PLoS computational biology*, 10(11):e1003915.

- Kingma, D. P. (2013). Fast gradient-based inference with continuous latent variable models in auxiliary form. *arXiv preprint arXiv:1306.0733*.
- Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.
- Kornell, N. and Bjork, R. A. (2008). Learning concepts and categories: Is spacing the “enemy of induction”? *Psychological science*, 19(6):585–592.
- Kotovsky, L. and Gentner, D. (1996). Comparison and categorization in the development of relational similarity. *Child Development*, 67(6):2797–2822.
- Kouider, S., Barbot, A., Madsen, K. H., Lehericy, S., and Summerfield, C. (2016). Task relevance differentially shapes ventral visual stream sensitivity to visible and invisible faces. *Neuroscience of consciousness*, 2016(1):niw021.
- Kriegeskorte, N., Mur, M., and Bandettini, P. A. (2008). Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:4.
- Krueger, K. A. and Dayan, P. (2009). Flexible shaping: How learning in small steps helps. *Cognition*, 110(3):380–394.
- Kruschke, J. K. (1992). Alcové: an exemplar-based connectionist model of category learning. *Psychological review*, 99(1):22.
- Kumaran, D., Hassabis, D., and McClelland, J. L. (2016). What learning systems do intelligent agents need? complementary learning systems theory updated. *Trends in cognitive sciences*, 20(7):512–534.
- Kumaran, D. and McClelland, J. L. (2012). Generalization through the recurrent interaction of episodic memories: a model of the hippocampal system. *Psychological review*, 119(3):573.
- Kurtz, K. H. and Hovland, C. I. (1956). Concept learning with differing sequences of instances. *Journal of experimental psychology*, 51(4):239.

- Kurtz, K. J. and Honke, G. (2020). Sorting out the problem of inert knowledge: Category construction to promote spontaneous transfer. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 46(5):803.
- Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338.
- Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and brain sciences*, 40.
- Lake, B. M., Vallabha, G. K., and McClelland, J. L. (2009). Modeling unsupervised perceptual category learning. *IEEE Transactions on Autonomous Mental Development*, 1(1):35–43.
- Lawrence, S., Giles, C. L., Tsoi, A. C., and Back, A. D. (1997). Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113.
- Lee, J. L. C., Everitt, B. J., and Thomas, K. L. (2004). Independent cellular processes for hippocampal memory consolidation and reconsolidation. *Science*, 304(5672):839–43.
- Lemon, N. and Manahan-vaughan, D. (2006). Dopamine D 1 / D 5 Receptors Gate the Acquisition of Novel Information through Hippocampal Long-Term Potentiation and Long-Term Depression. *The Journal of Neuroscience*, 26(29):7723–7729.
- Leong, Y. C., Radulescu, A., Daniel, R., DeWoskin, V., and Niv, Y. (2017). Dynamic interaction between reinforcement learning and attention in multidimensional environments. *Neuron*, 93(2):451–463.
- LePelley, M. and McLaren, I. (2004). Associative history affects the associative change undergone by both presented and absent cues in human causal learning. *Journal of Experimental Psychology: Animal Behavior Processes*, 30(1):67.
- Lever, C., Burton, S., Jeewajee, A., O’Keefe, J., and Burgess, N. (2009). Boundary vector cells in the subiculum of the hippocampal formation. *Journal of Neuroscience*, 29(31):9771–9777.

- Lillicrap, T. P., Cownden, D., Tweed, D. B., and Akerman, C. J. (2016). Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications*, 7(1):1–10.
- Lindsay, G. W. (2020). Attention in psychology, neuroscience, and machine learning. *Frontiers in Computational Neuroscience*, 14:29.
- Lindsay, G. W. and Miller, K. D. (2018). How biological attention mechanisms improve task performance in a large-scale visual system model. *ELife*, 7:e38105.
- Lisman, J. E. and Grace, A. A. (2005). The Hippocampal-VTA Loop : Controlling the Entry of Information into Long-Term Memory. *Neuron*, 46(5):703–713.
- Livingstone, M. and Hubel, D. (1988). Segregation of form, color, movement, and depth: anatomy, physiology, and perception. *Science*, 240(4853):740–749.
- Llera-Montero, M., Sacramento, J., and Costa, R. P. (2019). Computational roles of plastic probabilistic synapses. *Current opinion in neurobiology*, 54:90–97.
- Lodato, S. and Arlotta, P. (2015). Generating neuronal diversity in the mammalian cerebral cortex. *Annual review of cell and developmental biology*, 31:699–720.
- Lomber, S. G., Meredith, M. A., and Kral, A. (2010). Cross-modal plasticity in specific auditory cortices underlies visual compensations in the deaf. *Nature neuroscience*, 13(11):1421–1427.
- Longman, C. S., Lavric, A., Munteanu, C., and Monsell, S. (2014). Attentional inertia and delayed orienting of spatial attention in task-switching. *Journal of Experimental Psychology: Human Perception and Performance*, 40(4):1580.
- Love, B. C., Medin, D. L., and Gureckis, T. M. (2004). Sustain: a network model of category learning. *Psychological review*, 111(2):309.
- Lucas, C. G., Bridgers, S., Griffiths, T. L., and Gopnik, A. (2014). When children are better (or at least more open-minded) learners than adults: Developmental differences in learning the forms of causal relationships. *Cognition*, 131(2):284–299.

- Luo, X., Roads, B. D., and Love, B. C. (2020). The costs and benefits of goal-directed attention in deep convolutional neural networks. *arXiv preprint arXiv:2002.02342*.
- Mack, M. L., Love, B. C., and Preston, A. R. (2016). Dynamic updating of hippocampal object representations reflects new conceptual knowledge. *Proceedings of the National Academy of Sciences*, 113(46):13203–13208.
- Mack, M. L., Preston, A. R., and Love, B. C. (2020). Ventromedial prefrontal cortex compression during concept learning. *Nature communications*, 11(1):1–11.
- Mackintosh, N. J. (1975). A theory of attention: Variations in the associability of stimuli with reinforcement. *Psychological review*, 82(4):276.
- Maia, T. V. (2009). Reinforcement learning, conditioning, and the brain: Successes and challenges. *Cognitive, Affective, & Behavioral Neuroscience*, 9(4):343–364.
- Manchin, A., Abbasnejad, E., and van den Hengel, A. (2019). Reinforcement learning with attention that works: A self-supervised approach. In *International Conference on Neural Information Processing*, pages 223–230. Springer.
- Mante, V., Sussillo, D., Shenoy, K. V., and Newsome, W. T. (2013). Context-dependent computation by recurrent dynamics in prefrontal cortex. *nature*, 503(7474):78–84.
- Mareschal, D. (2010). Computational perspectives on cognitive development. *Wiley Interdisciplinary Reviews: Cognitive Science*, 1(5):696–708.
- Mareschal, D., Quinn, P. C., Lea, S. E., and Lea, S. (2010). *The making of human concepts*. Oxford Series in Developmental.
- Mason, A., Farrell, S., Howard-jones, P., and Ludwig, C. J. H. (2017). The role of reward and reward uncertainty in episodic memory. *Journal of Memory and Language*, 96:62–77.
- Mattar, M. G. and Daw, N. D. (2018). Prioritized memory access explains planning and hippocampal replay. *Nature Neuroscience*, 21(11):1609–1617.

- Mazzoni, P., Andersen, R. A., and Jordan, M. I. (1991). A more biologically plausible learning rule for neural networks. *Proceedings of the National Academy of Sciences*, 88(10):4433–4437.
- McClelland, J. L., McNaughton, B. L., and O’Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419.
- McClelland, J. L., Rumelhart, D. E., Group, P. R., et al. (1986). Parallel distributed processing. *Explorations in the Microstructure of Cognition*, 2:216–271.
- Mcclure, S. M., Berns, G. S., and Montague, P. R. (2003). Temporal Prediction Errors in a Passive Learning Task Activate Human Striatum. *Neuron*, 38(2):339–346.
- McNaughton, B. L., Battaglia, F. P., Jensen, O., Moser, E. I., and Moser, M.-B. (2006). Path integration and the neural basis of the ‘cognitive map’. *Nature Reviews Neuroscience*, 7(8):663–678.
- Miller, E. K. and Cohen, J. D. (2001). An integrative theory of prefrontal cortex function. *Annual review of neuroscience*, 24(1):167–202.
- Miller, K. D. (2016). Canonical computations of cerebral cortex. *Current opinion in neurobiology*, 37:75–84.
- Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.
- Mocanu, D. C., Vega, M. T., Eaton, E., Stone, P., and Liotta, A. (2016). Online contrastive divergence with generative replay: Experience replay without storing data. *arXiv preprint arXiv:1610.05555*.

- Momennejad, I., Russek, E. M., Cheong, J. H., Botvinick, M. M., Daw, N. D., and Gershman, S. J. (2017). The successor representation in human reinforcement learning. *Nature Human Behaviour*, 1(9):680–692.
- Morel, D., Singh, C., and Levy, W. B. (2018). Linearization of excitatory synaptic integration at no extra cost. *Journal of Computational Neuroscience*, 44(2):173–188.
- Morris, R. (2006). Elements of a neurobiological theory of hippocampal function: the role of synaptic plasticity, synaptic tagging and schemas. *European Journal of Neuroscience*, 23(11):2829–2846.
- Morris, R. W., Dezfouli, A., Griffiths, K. R., and Balleine, B. W. (2014). Action-value comparisons in the dorsolateral prefrontal cortex control choice between goal-directed actions. *Nature communications*, 5(1):1–10.
- Mott, A., Zoran, D., Chrzanowski, M., Wierstra, D., and Rezende, D. J. (2019). Towards interpretable reinforcement learning using attention augmented agents. In *Advances in Neural Information Processing Systems*, pages 12329–12338.
- Munakata, Y. and McClelland, J. L. (2003). Connectionist models of development. *Developmental Science*, 6(4):413–429.
- Nádasy, Z., Hirase, H., Czurkó, A., Csicsvari, J., and Buzsáki, G. (1999). Replay and time compression of recurring spike sequences in the hippocampus. *Journal of Neuroscience*, 19(21):9497–9507.
- Narvekar, S. (2017). Curriculum learning in reinforcement learning. In *IJCAI*, pages 5195–5196.
- Narvekar, S., Sinapov, J., and Stone, P. (2017). Autonomous task sequencing for customized curriculum design in reinforcement learning. In *IJCAI*, pages 2536–2542.
- Narvekar, S. and Stone, P. (2018). Learning curriculum policies for reinforcement learning. *arXiv preprint arXiv:1812.00285*.

- Nasser, H. M., Calu, D. J., Schoenbaum, G., and Sharpe, M. J. (2017). The dopamine prediction error: contributions to associative models of reward learning. *Frontiers in psychology*, 8:244.
- Navawongse, R. and Eichenbaum, H. (2013). Distinct pathways for rule-based retrieval and spatial mapping of memory representations in hippocampal neurons. *Journal of Neuroscience*, 33(3):1002–1013.
- Niv, Y. (2009). Reinforcement learning in the brain. *Journal of Mathematical Psychology*, 53(3):139–154.
- Niv, Y. (2019). Learning task-state representations. *Nature neuroscience*, 22(10):1544–1553.
- Niv, Y., Daniel, R., Geana, A., Gershman, S. J., Leong, Y. C., Radulescu, A., and Wilson, R. C. (2015). Reinforcement learning in multidimensional environments relies on attention mechanisms. *Journal of Neuroscience*, 35(21):8145–8157.
- O’keefe, J. and Nadel, L. (1978). *The hippocampus as a cognitive map*. Oxford: Clarendon Press.
- Olafsdottir, F. H., Bush, D., and Barry, C. (2018). Review The Role of Hippocampal Replay in Memory and Planning. *Current Biology*, 28(1):37–50.
- Ólafsdóttir, H. F., Barry, C., Saleem, A. B., Hassabis, D., and Spiers, H. J. (2015). Hippocampal place cells construct reward related sequences through unexplored space. *Elife*, 4:e06063.
- O’Reilly, R. C. (1996). Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm. *Neural computation*, 8(5):895–938.
- O’Reilly, R. C. and Norman, K. A. (2002). Hippocampal and neocortical contributions to memory: Advances in the complementary learning systems framework. *Trends in cognitive sciences*, 6(12):505–510.
- O’Reilly, R. C. and Rudy, J. W. (2000). Computational principles of learning in the neocortex and hippocampus. *Hippocampus*, 10(4):389–397.

- O’Doherty, J. P., Lee, S. W., and McNamee, D. (2015). The structure of reinforcement-learning mechanisms in the human brain. *Current Opinion in Behavioral Sciences*, 1:94–100.
- O’Neill, J., Boccarda, C., Stella, F., Schönenberger, P., and Csicsvari, J. (2017). Superficial layers of the medial entorhinal cortex replay independently of the hippocampus. *Science*, 355(6321):184–188.
- Padoa-Schioppa, C. and Assad, J. A. (2006). Neurons in the orbitofrontal cortex encode economic value. *Nature*, 441(7090):223–226.
- Paneri, S. and Gregoriou, G. G. (2017). Top-down control of visual attention by the prefrontal cortex. functional specialization and long-range interactions. *Frontiers in neuroscience*, 11:545.
- Pearce, J. M. and Hall, G. (1980). A model for pavlovian learning: variations in the effectiveness of conditioned but not of unconditioned stimuli. *Psychological review*, 87(6):532.
- Pearce, J. M. and Mackintosh, N. J. (2010). Two theories of attention: A review and a possible integration. *Attention and associative learning: From brain to behaviour*, pages 11–39.
- Pennartz, C., Ito, R., Verschure, P., Battaglia, F., and Robbins, T. (2011). The hippocampal–striatal axis in learning, prediction and goal-directed behavior. *Trends in neurosciences*, 34(10):548–559.
- Peterson, G. B. (2004). A day of great illumination: Bf skinner’s discovery of shaping. *Journal of the experimental analysis of behavior*, 82(3):317–328.
- Philiastides, M. G., Biele, G., and Heekeren, H. R. (2010). A mechanistic account of value computation in the human brain. *Proceedings of the National Academy of Sciences*, 107(20):9430–9435.
- Piche, M., Chabot, N., Bronchti, G., Miceli, D., Lepore, F., and Guillemot, J.-P. (2007). Auditory responses in the visual cortex of neonatally enucleated rats. *Neuroscience*, 145(3):1144–1156.

- Place, R., Farovik, A., Brockmann, M., and Eichenbaum, H. (2016). Bidirectional prefrontal-hippocampal interactions support context-guided memory. *Nature neuroscience*, 19(8):992–994.
- Plappert, M., Houthoofd, R., Dhariwal, P., Sidor, S., Chen, R. Y., Chen, X., Asfour, T., Abbeel, P., and Andrychowicz, M. (2017). Parameter space noise for exploration. *arXiv preprint arXiv:1706.01905*.
- Plunkett, K., Karmiloff-Smith, A., Bates, E., Elman, J. L., and Johnson, M. H. (1997). Connectionism and developmental psychology. *Journal of Child Psychology and Psychiatry*, 38(1):53–80.
- Poggio, T., Fahle, M., and Edelman, S. (1992). Fast perceptual learning in visual hyperacuity. *Science*, 256(5059):1018–1021.
- Preston, A. R. and Eichenbaum, H. (2013). Interplay of hippocampus and prefrontal cortex in memory. *Current Biology*, 23(17):R764–R773.
- Pritzel, A., Uria, B., Srinivasan, S., Puigdomenech, A., Vinyals, O., Hassabis, D., Wierstra, D., and Blundell, C. (2017). Neural episodic control. *arXiv preprint arXiv:1703.01988*.
- Quartz, S. R. and Sejnowski, T. J. (1997). The neural basis of cognitive development: A constructivist manifesto. *Behavioral and brain sciences*, 20(4):537–556.
- Quillan, M. R. (1966). Semantic memory. Technical report, BOLT BERANEK AND NEWMAN INC CAMBRIDGE MA.
- Radulescu, A., Niv, Y., and Ballard, I. (2019). Holistic reinforcement learning: the role of structure and attention. *Trends in cognitive sciences*.
- Radulescu, A., Niv, Y., and Daw, N. D. A particle filtering account of selective attention during learning.
- Ragozzino, M. E., Kim, J., Hassert, D., Minniti, N., and Kiang, C. (2003). The contribution of the rat prelimbic-infralimbic areas to different forms of task switching. *Behavioral neuroscience*, 117(5):1054.

- Rescorla, R. A. (1972). A theory of pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. *Current research and theory*, pages 64–99.
- Rescorla, R. A. and Lolordo, V. M. (1965). Inhibition of avoidance behavior. *Journal of comparative and physiological psychology*, 59(3):406.
- Reynolds, G. S. (1961). Attention in the pigeon. *Journal of the Experimental Analysis of Behavior*, 4(3):203.
- Reynolds, J. H. and Heeger, D. J. (2009). The normalization model of attention. *Neuron*, 61(2):168–185.
- Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*.
- Rich, E. L. and Shapiro, M. L. (2007). Prelimbic/infralimbic inactivation impairs memory for multiple task switches, but not flexible selection of familiar tasks. *Journal of Neuroscience*, 27(17):4747–4755.
- Richards, J. E. and Anderson, D. R. (2004). Attentional inertia in children’s extended looking at television. In *Advances in child development and behavior*, volume 32, pages 163–212. Elsevier.
- Richland, L. E., Morrison, R. G., and Holyoak, K. J. (2006). Children’s development of analogical reasoning: Insights from scene analogy problems. *Journal of experimental child psychology*, 94(3):249–273.
- Ritter, S., Wang, J. X., Kurth-Nelson, Z., and Botvinick, M. (2018). Episodic control as meta-reinforcement learning. *bioRxiv*, page 360537.
- Robinson, T. M. (1987). *Heraclitus: fragments*. University of Toronto Press.
- Roesch, M. R., Singh, T., Brown, P. L., Mullins, S. E., and Schoenbaum, G. (2009). rats deciding between differently delayed or sized rewards. *Journal of Neuroscience*, 29(42):13365–13376.

- Rohrer, D., Dedrick, R. F., and Burgess, K. (2014). The benefit of interleaved mathematics practice is not limited to superficially similar kinds of problems. *Psychonomic bulletin & review*, 21(5):1323–1330.
- Rolls, E. T. (2007). An attractor network in the hippocampus: theory and neurophysiology. *Learning & memory*, 14(11):714–731.
- Rosen, Z. B., Cheung, S., and Siegelbaum, S. A. (2015). Midbrain dopamine neurons bidirectionally regulate CA3-CA1 synaptic drive. *Nature Neuroscience*, 18(12):1763–1771.
- Rosenthal, O., Fusi, S., and Hochstein, S. (2001). Forming classes by stimulus frequency: Behavior and theory. *Proceedings of the National Academy of Sciences*, 98(7):4265–4270.
- Rossi, A. F. and Paradiso, M. A. (1995). Feature-specific effects of selective visual attention. *Vision research*, 35(5):621–634.
- Rouhani, N., Norman, K. A., and Niv, Y. (2018). Dissociable effects of surprising rewards on learning and memory. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 44(9):1430–1443.
- Rushworth, M. F. and Behrens, T. E. (2008). Choice, uncertainty and value in prefrontal and cingulate cortex. *Nature neuroscience*, 11(4):389.
- Sacramento, J., Costa, R. P., Bengio, Y., and Senn, W. (2018). Dendritic cortical microcircuits approximate the backpropagation algorithm. In *Advances in neural information processing systems*, pages 8721–8732.
- Saenz, M., Buracas, G. T., and Boynton, G. M. (2002). Global effects of feature-based attention in human visual cortex. *Nature neuroscience*, 5(7):631–632.
- Sara, S. J. (2000). Retrieval and Reconsolidation : Toward a Neurobiology of Remembering. *Learning & Memory*, 7(2):73–84.
- Sarel, A., Finkelstein, A., Las, L., and Ulanovsky, N. (2017). Vectorial representation of spatial goals in the hippocampus of bats. *Science*, 355(6321):176–180.

- Scellier, B. and Bengio, Y. (2017). Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in computational neuroscience*, 11:24.
- Schlichting, M. L. and Preston, A. R. (2016). Hippocampal–medial prefrontal circuit supports memory updating during learning and post-encoding rest. *Neurobiology of learning and memory*, 134:91–106.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.
- Schrimpf, M., Kumbhani, J., Hong, H., Majaj, N. J., Rajalingham, R., Issa, E. B., Kar, K., Bashivan, P., Prescott-Roy, J., Schmidt, K., et al. (2018). Brain-score: Which artificial neural network for object recognition is most brain-like? *BioRxiv*, page 407007.
- Schroeder, C. E., Lindsley, R. W., Specht, C., Marcovici, A., Smiley, J. F., and Javitt, D. C. (2001). Somatosensory input to auditory association cortex in the macaque monkey. *Journal of neurophysiology*, 85(3):1322–1327.
- Schultz, W. (1998). Predictive reward signal of dopamine neurons. *Journal of neurophysiology*, 80(1):1–27.
- Schultz, W. (2016). Dopamine Reward Prediction Error Coding. *Dialogues in Clinical Neuroscience*, 18(1):23–32.
- Schultz, W., Apicella, P., Scarnati, E., and Ljungberg, T. (1992). Neuronal Activity in Monkey Ventral Striatum Related to the Expectation of Reward. *Journal of Neuroscience*, 12(12):4595–4610.
- Schultz, W., Dayan, P., and Montague, P. R. (1997). A neural substrate of prediction and reward. *Science*, 275(5306):1593–1599.
- Schyns, P. G., Goldstone, R. L., and Thibaut, J.-P. (1998). The development of features in object concepts. *Behavioral and brain Sciences*, 21(1):1–17.
- Scoville, W. B. and Milner, B. (1957). Loss of recent memory after bilateral hippocampal lesions. *Journal of neurology, neurosurgery, and psychiatry*, 20(1):11.

- Seo, H. and Lee, D. (2008). Cortical mechanisms for reinforcement learning in competitive games. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 363(1511):3845–3857.
- Seo, M., Lee, E., and Averbeck, B. B. (2012). Action selection and action value in frontal-striatal circuits. *Neuron*, 74(5):947–960.
- Setlow, B., Schoenbaum, G., and Gallagher, M. (2003). Neural encoding in ventral striatum during olfactory discrimination learning. *Neuron*, 38(4):625–636.
- Sharpe, M. J., Chang, C. Y., Liu, M. A., Batchelor, H. M., Mueller, L. E., Jones, J. L., Niv, Y., and Schoenbaum, G. (2017). Dopamine transients are sufficient and necessary for acquisition of model-based associations. *Nature Neuroscience*, 20(5):735–742.
- Shepard, R. N., Hovland, C. I., and Jenkins, H. M. (1961). Learning and memorization of classifications. *Psychological monographs: General and applied*, 75(13):1.
- Shin, H., Lee, J. K., Kim, J., and Kim, J. (2017). Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pages 2990–2999.
- Shohamy, D. and Adcock, R. A. (2010). Dopamine and adaptive memory. *Trends in Cognitive Sciences*, 14(10):464–472.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Skaggs, W. E. and McNaughton, B. L. (1996). Replay of neuronal firing sequences in rat hippocampus during sleep following spatial experience. *Science*, 271(5257):1870–1873.
- Skinner, B. F. (1935). Two types of conditioned reflex and a pseudo type. *The Journal of General Psychology*, 12(1):66–77.
- Sorokin, I., Seleznev, A., Pavlov, M., Fedorov, A., and Ignateva, A. (2015). Deep attention recurrent q-network. *arXiv preprint arXiv:1512.01693*.
- Spelke, E. S. (1990). Principles of object perception. *Cognitive science*, 14(1):29–56.

- Stachenfeld, K. L., Botvinick, M., and Gershman, S. J. (2014). Design principles of the hippocampal cognitive map. In *Advances in neural information processing systems*, pages 2528–2536.
- Stachenfeld, K. L., Botvinick, M. M., and Gershman, S. J. (2017). The hippocampus as a predictive map. *Nature neuroscience*, 20(11):1643.
- Sutton, R. S. and Barto, A. G. (1998). Reinforcement learning: An introduction.
- Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.
- Takahashi, Y. K., Batchelor, H. M., Liu, B., Khanna, A., Morales, M., and Schoenbaum, G. (2017). Dopamine neurons respond to errors in the prediction of sensory features of expected rewards. *Neuron*, 95(6):1395–1405.
- Taube, J. S., Muller, R. U., and Ranck, J. B. (1990). Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis. *Journal of Neuroscience*, 10(2):420–435.
- Testolin, A., Stoianov, I., and Zorzi, M. (2017). Letter perception emerges from unsupervised deep learning and recycling of natural image features. *Nature human behaviour*, 1(9):657–664.
- Testolin, A. and Zorzi, M. (2016). Probabilistic models and generative neural networks: Towards an unified framework for modeling normal and impaired neurocognitive functions. *Frontiers in Computational Neuroscience*, 10:73.
- Thierry, A.-m., Gioanni, Y., Degenetais, E., and Glowinski, J. (2000). Hippocampo-Prefrontal Cortex Pathway : Anatomical and Electrophysiological Characteristics. *Hippocampus*, 10(4):411–419.
- Thorndike, E. L. (1911). Animal intelligence: Experimental studies.
- Tobler, P. N., Dickinson, A., and Schultz, W. (2003). Coding of predicted reward omission by dopamine neurons in a conditioned inhibition paradigm. *Journal of Neuroscience*, 23(32):10402–10410.

- Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological review*, 55(4):189.
- Torrey, L. (2009). Relational transfer in reinforcement learning. Technical report, University of Wisconsin-Madison Department of Computer Sciences.
- Treue, S. (2003). Visual attention: the where, what, how and why of saliency. *Current opinion in neurobiology*, 13(4):428–432.
- Treue, S. and Trujillo, J. C. M. (1999). Feature-based attention influences motion processing gain in macaque visual cortex. *Nature*, 399(6736):575–579.
- Triesch, J., Teuscher, C., Deák, G. O., and Carlson, E. (2006). Gaze following: Why (not) learn it? *Developmental science*, 9(2):125–147.
- Tse, D., Takeuchi, T., Kakeyama, M., Kajii, Y., Okuno, H., Tohyama, C., Bito, H., and Morris, R. G. (2011). Schema-dependent gene activation and memory encoding in neocortex. *Science*, 333(6044):891–895.
- van de Ven, G. M., Siegelmann, H. T., and Tolias, A. S. (2020). Brain-inspired replay for continual learning with artificial neural networks. *Nature communications*, 11(1):1–14.
- van de Ven, G. M. and Tolias, A. S. (2018). Generative replay with feedback connections as a general strategy for continual learning. *arXiv preprint arXiv:1809.10635*.
- Van Otterlo, M. (2005). A survey of reinforcement learning in relational domains. *Centre for Telematics and Information Technology (CTIT) University of Twente, Tech. Rep.*
- Vila-Ballo, A., Mas-Herrero, E., Ripolles, P., Simo, M., Miro, J., Cucurell, D., Lopez-Barroso, D., Juncadella, M., Marco-Pallares, J., Falip, M., and Rodriguez-Fornells, A. (2017). Unraveling the Role of the Hippocampus in Reversal Learning. *Journal of Neuroscience*, 37(28):6686–6697.
- Wang, J. X., Kurth-Nelson, Z., Kumaran, D., Tirumala, D., Soyer, H., Leibo, J. Z., Hassabis, D., and Botvinick, M. (2018). Prefrontal cortex as a meta-reinforcement learning system. *Nature neuroscience*, 21(6):860–868.

- Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., Blundell, C., Kumaran, D., and Botvinick, M. (2016). Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*.
- Waxman, S. R. (1998). Early expectations and the shaping role of language. *Psychology of Learning and Motivation: Advances in Research and Theory*, page 249.
- Waxman, S. R. and Booth, A. E. (2000). Principles that are invoked in the acquisition of words, but not facts. *Cognition*, 77(2):B33–B43.
- Westermann, G., Mareschal, D., Johnson, M. H., Sirois, S., Spratling, M. W., and Thomas, M. S. (2007). Neuroconstructivism. *Developmental science*, 10(1):75–83.
- Wharton, C. M., Holyoak, K. J., Downing, P. E., Lange, T. E., Wickens, T. D., and Melz, E. R. (1994). Below the surface: Analogical similarity and retrieval competition in reminding. *Cognitive Psychology*, 26(1):64–101.
- Whitman, J. R. and Garner, W. (1963). Concept learning as a function of form of internal structure. *Journal of Verbal Learning and Verbal Behavior*, 2(2):195–202.
- Whittington, J. C. and Bogacz, R. (2017). An approximation of the error back-propagation algorithm in a predictive coding network with local hebbian synaptic plasticity. *Neural computation*, 29(5):1229–1262.
- Whittington, J. C., Muller, T. H., Mark, S., Chen, G., Barry, C., Burgess, N., and Behrens, T. E. (2019). The tolmán-eichenbaum machine: Unifying space and relational memory through generalisation in the hippocampal formation. *bioRxiv*, page 770495.
- Wills, T. J., Lever, C., Cacucci, F., Burgess, N., and O’Keefe, J. (2005). Attractor dynamics in the hippocampal representation of the local environment. *Science*, 308(5723):873–876.
- Wilson, B., Mackintosh, N. J., and Boakes, R. A. (1985). Transfer of relational rules in matching and oddity learning by pigeons and corvids. *The Quarterly Journal of Experimental Psychology*, 37(4):313–332.

- Wilson, R. C. and Niv, Y. (2012). Inferring relevance in a changing world. *Frontiers in human neuroscience*, 5:189.
- Wimmer, G. E., Braun, E. K., Daw, N. D., and Shohamy, D. (2014). Episodic Memory Encoding Interferes with Reward Learning and Decreases Striatal Prediction Errors. *The Journal of Neuroscience*, 34(45):14901–14912.
- Wittmann, B. C., Schott, B. H., Guderian, S., Frey, J. U., Heinze, H.-j., and Düzel, E. (2005). Reward-Related fMRI Activation of Dopaminergic Midbrain Is Associated with Enhanced Hippocampus- Dependent Long-Term Memory Formation. *Neuron*, 45(3):459–467.
- Wolfe, J. M. and Horowitz, T. S. (2004). What attributes guide the deployment of visual attention and how do they do it? *Nature reviews neuroscience*, 5(6):495–501.
- Woolgar, A., Hampshire, A., Thompson, R., and Duncan, J. (2011). Adaptive coding of task-relevant information in human frontoparietal cortex. *Journal of Neuroscience*, 31(41):14592–14599.
- Xiao, X., Dong, Q., Gao, J., Men, W., Poldrack, R. A., and Xue, G. (2017). Transformed neural pattern reinstatement during episodic memory retrieval. *Journal of Neuroscience*, 37(11):2986–2998.
- Yamins, D. L. and DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3):356–365.
- Yang, H., Kwon, S. E., Severson, K. S., and O’connor, D. H. (2016). Origins of choice-related activity in mouse somatosensory cortex. *Nature neuroscience*, 19(1):127–134.
- Yasumatsu, N., Matsuzaki, M., Miyazaki, T., Noguchi, J., and Kasai, H. (2008). Principles of long-term dynamics of dendritic spines. *Journal of Neuroscience*, 28(50):13592–13608.
- Yerkes, R. M. and Morgulis, S. (1909). The method of pawlow in animal psychology. *Psychological Bulletin*, 6(8):257.

Zambaldi, V., Raposo, D., Santoro, A., Bapst, V., Li, Y., Babuschkin, I., Tuyls, K., Reichert, D., Lillicrap, T., Lockhart, E., et al. (2018). Relational deep reinforcement learning. *arXiv preprint arXiv:1806.01830*.

Zenke, F., Poole, B., and Ganguli, S. (2017). Continual learning through synaptic intelligence. *Proceedings of machine learning research*, 70:3987.

Appendix A

Supplementary Data for Experiment 1

This appendix contains supplementary data for Experiment 1 (Section 7.2). Figure A.1 shows a histogram of the test question results split by training regime. Figure A.2 shows the same results but for each block of training and test questions. Figure A.3 shows a histogram of the test question results based on the underlying relational rule for that block. Figure A.4 shows the same results but split by training regime.

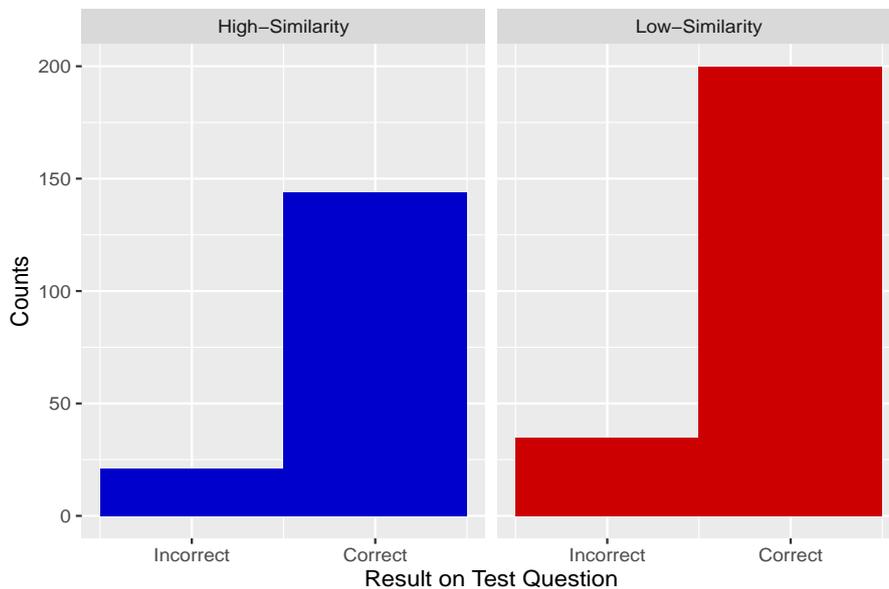


Figure A.1: *Histogram of responses to the test questions over all blocks. Each colour represents a different training regime.*

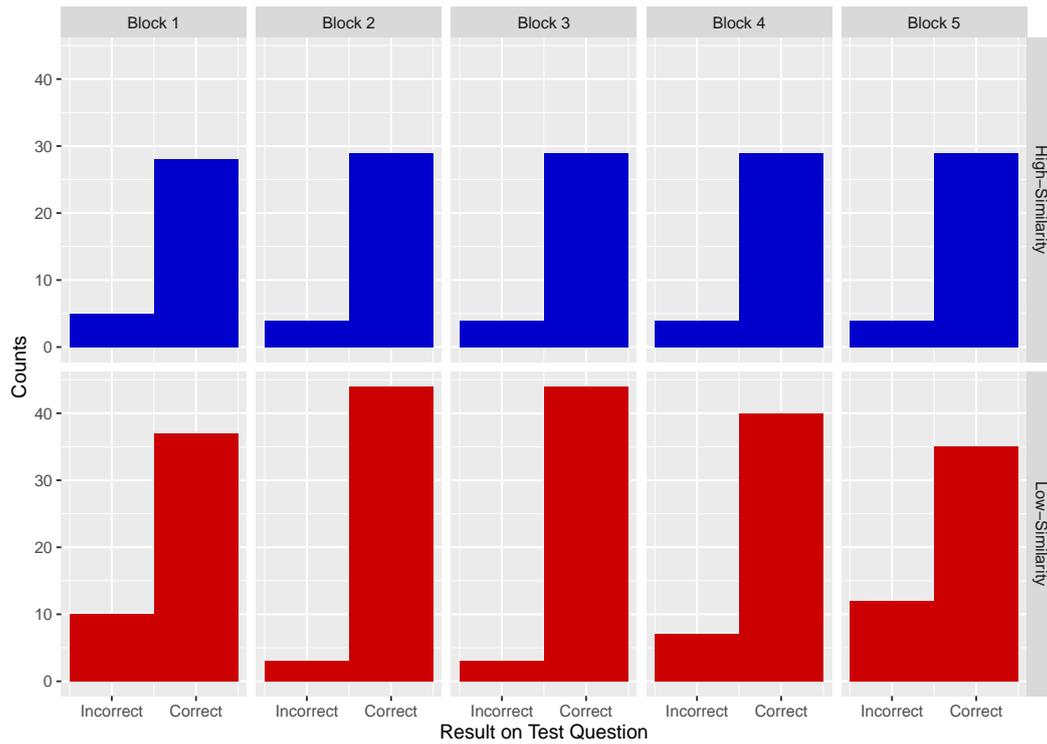


Figure A.2: Histogram of responses to the test question for each block. Each colour represents a different training regime.

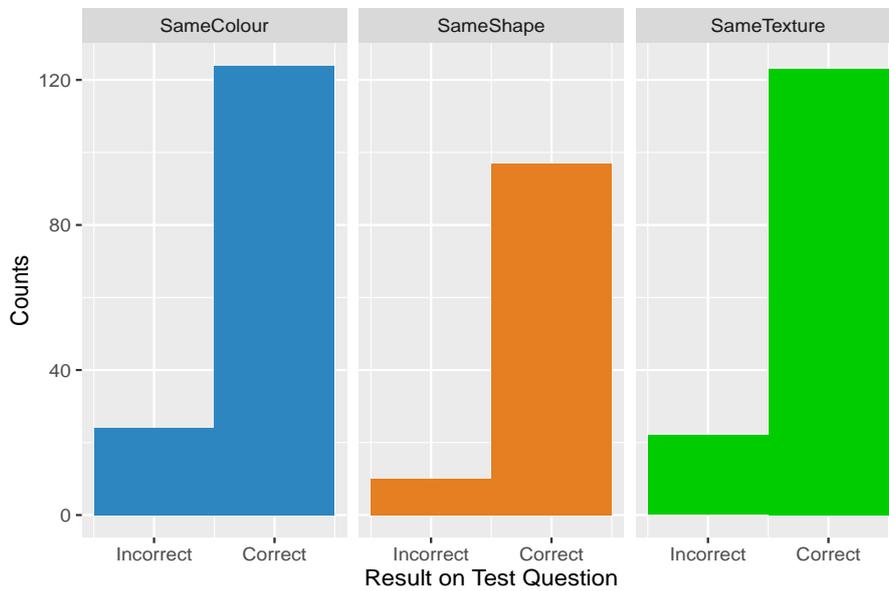


Figure A.3: Histogram of responses to the test questions for blocks with the same relational rule. Each colour represents a different relational rule.

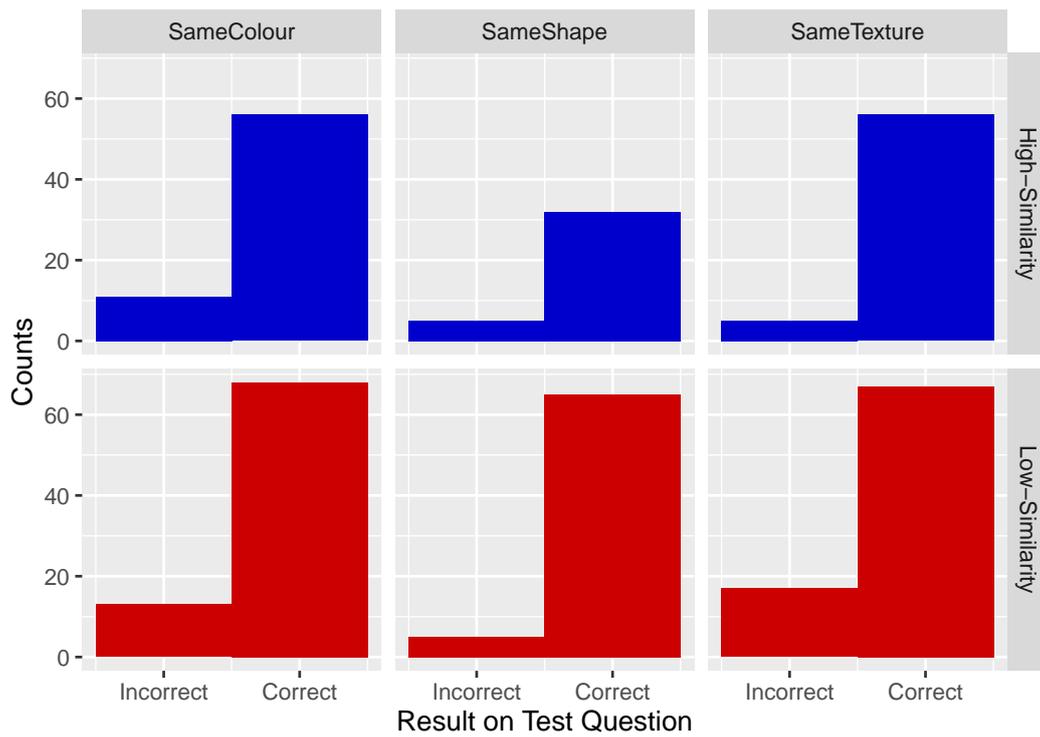


Figure A.4: Histogram of responses to the test questions for blocks with the same relational rule. Each colour represents a different training regime.

Appendix B

Supplementary Data for Experiment 2

This appendix contains supplementary data for Experiment 2 (Section 7.3). Figure B.1 shows a histogram of the score achieved on the first trial of a level summed over all training levels. Figure B.2 shows a histogram of the total score achieved during training. Figure B.3 shows a histogram of the score achieved on the first trial of the test level. Figure B.4 shows a histogram of the total score achieved on the test level.

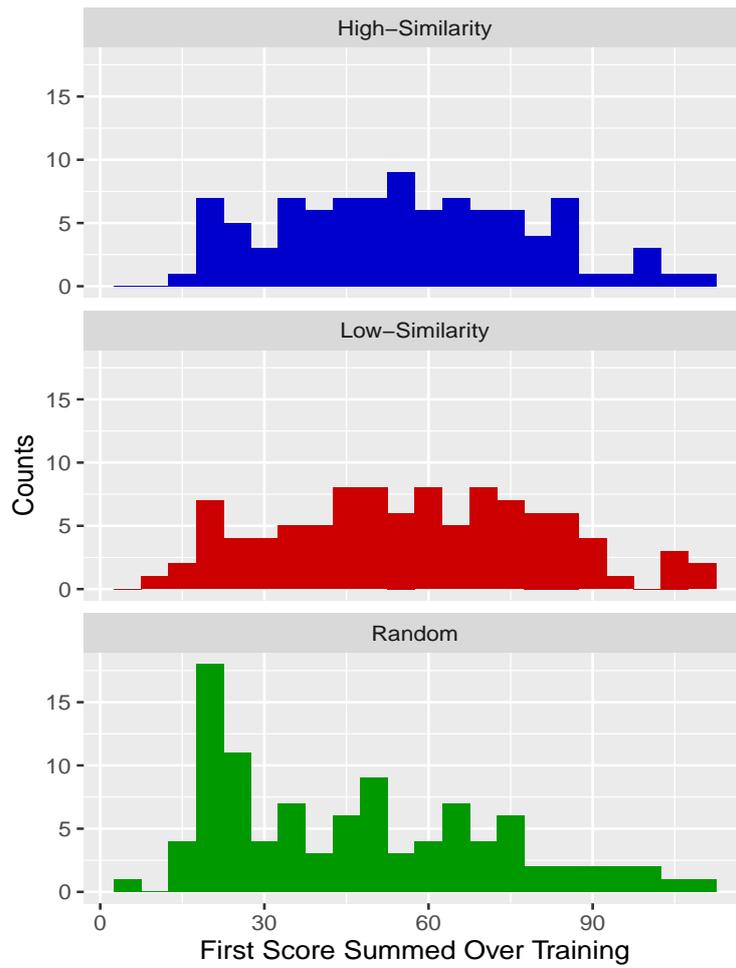


Figure B.1: *Histogram of the score achieved on the first trial of a level summed over all training levels. Each colour represents a different training regime. Participants were not told the rules of the games beforehand.*

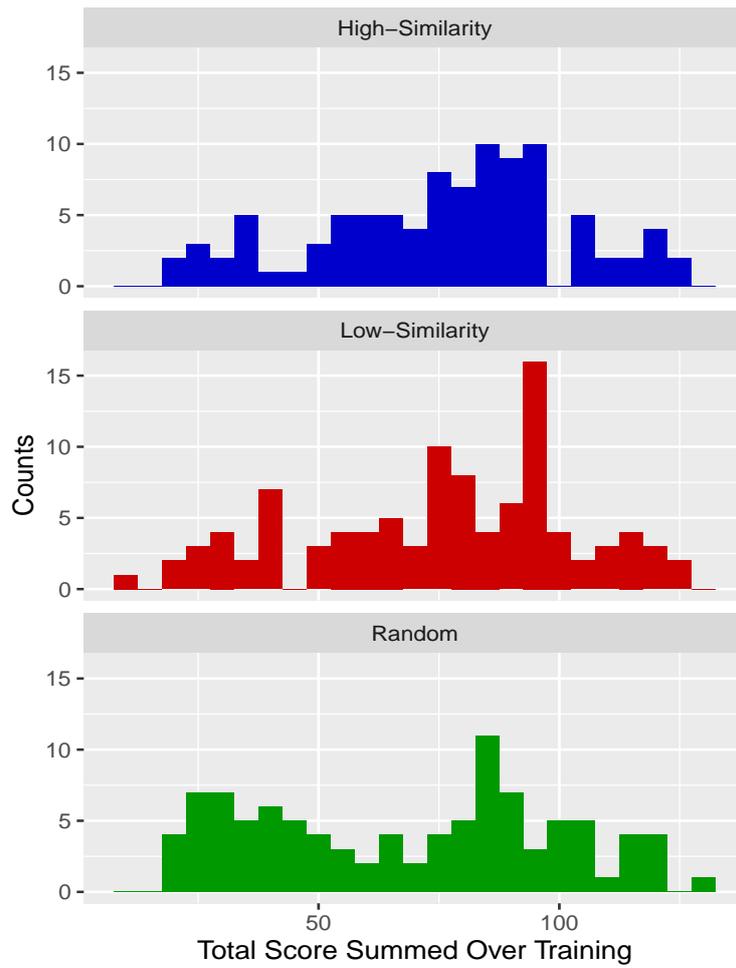


Figure B.2: *Histogram of the total score achieved during training. Each colour represents a different training regime. Participants were not told the rules of the games beforehand.*

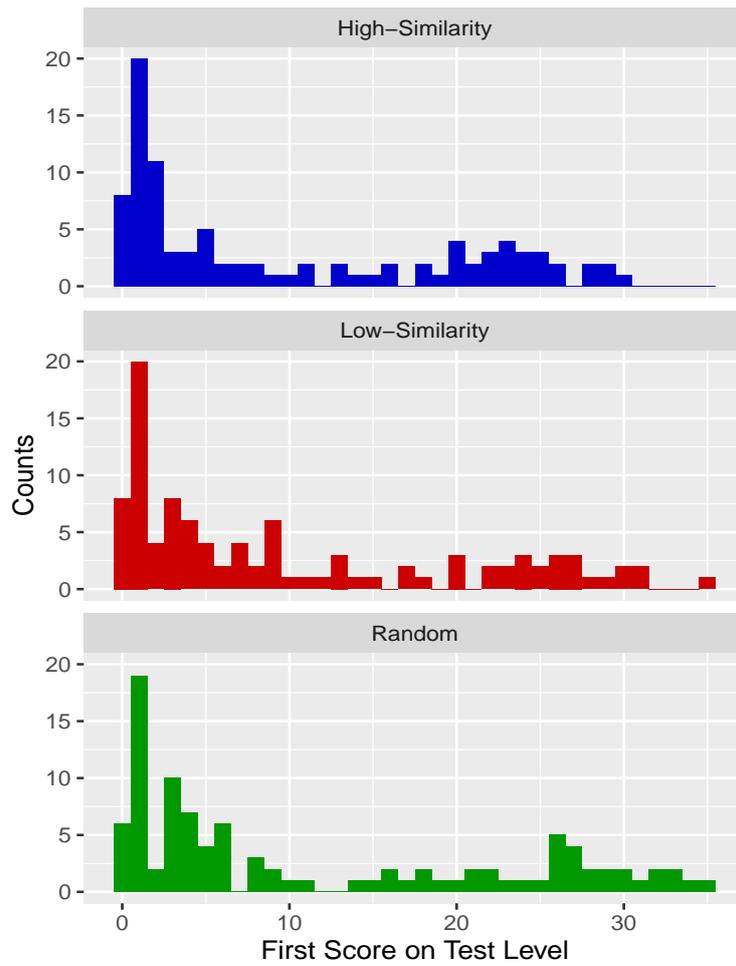


Figure B.3: *Histogram of the score achieved on the first trial of the test game. Each colour represents a different training regime. Participants were not told the rules of the games beforehand.*

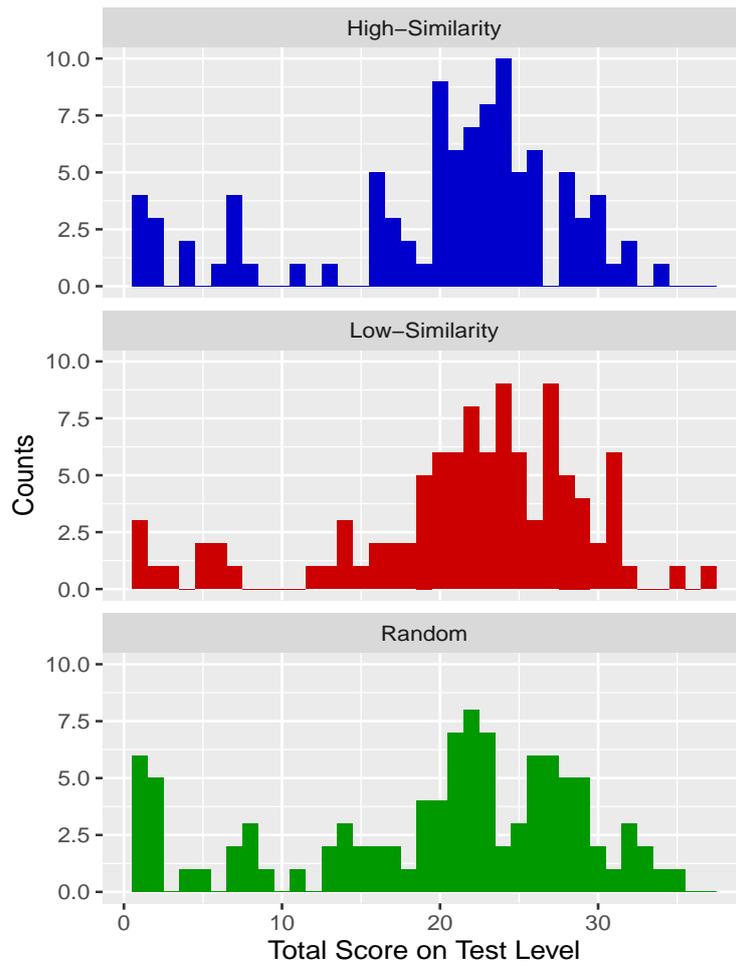


Figure B.4: *Histogram of the total score achieved on the test game. Each colour represents a different training regime. Participants were not told the rules of the games beforehand.*

Appendix C

Experiment 2.5

In this appendix we present the results of Experiment 2.5. The purpose of this experiment was to investigate whether the number of training games interacted with the effect of perceptual similarity on transfer performance. We hypothesised that the effect of perceptual similarity during training would be more pronounced if participants were tested earlier on during learning.

C.1 Methods

The experimental procedure was the same as Experiment 2 (Section 7.3) but we reduced the length of the experiment to 9 training games and 1 test game. We chose 9 training games because this allowed us to change every feature of each object once. This was important because it helped prevent participants from forming hypotheses based on a single perceptual feature that remained constant during training. We chose to keep the features of the goal object constant because they carry no information with regards to the underlying rules and we wanted to keep the number of training games as small as possible.

In Experiment 2 all training games lasted 30 seconds and the test game lasted 120 seconds. For Experiment 2.5 we set the duration of all training and test games to 30 seconds. In addition, we only tested the high-perceptual similarity and the low-perceptual similarity training regimes using 50 different random seed values. Figure C.1 shows example trajectories for each training regime. In total 200 participants were recruited (Male=121, Female=78, Undisclosed=1) using the online

platform ‘Prolific Academic’ and they were rewarded £2.50 for participation. As before, all participants had to be aged 18-30, speak fluent English and be using a desktop computer. Participants were removed from the analysis if the number of key presses was 0 for any of the games. Participants were randomly assigned to the two different training regimes resulting in 103 participants in the high-perceptual similarity condition and 94 participants in the low-perceptual similarity condition.

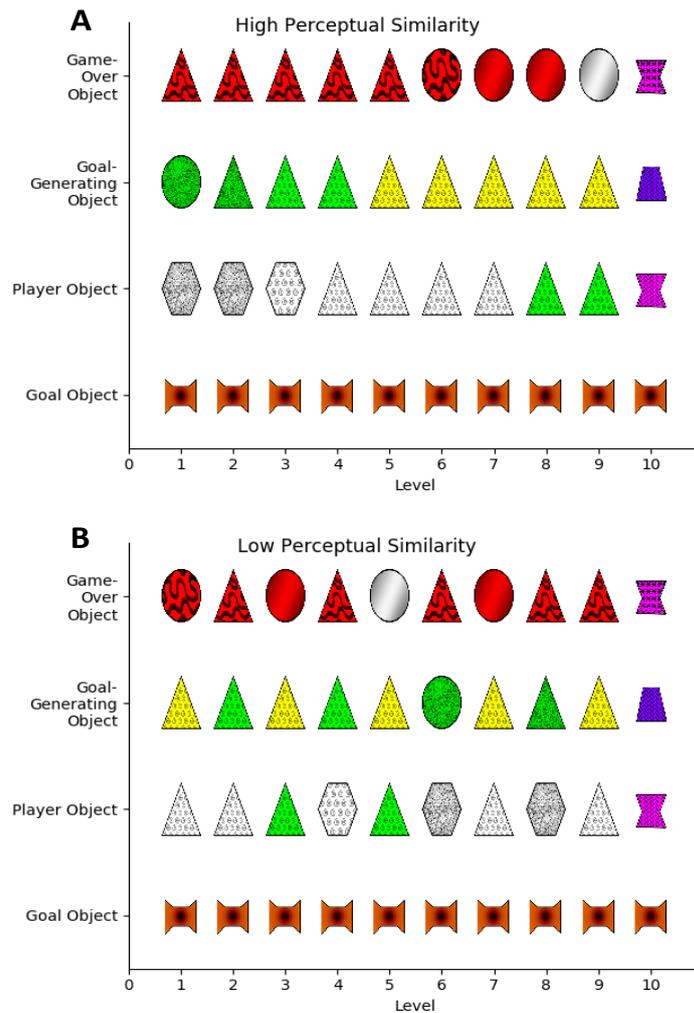


Figure C.1: *Depiction of the different training conditions. The y-axis represents the object and the x-axis represents the level number. The final level represents the perceptually novel test level. (A) High-perceptual similarity condition. On any two consecutive levels only the shape, texture or colour of one object was changed. If the player object’s or goal-generating object’s texture changed then both needed to be changed to ensure that there was always a texture match. (B) Low-perceptual similarity condition. The high-perceptual similarity condition was randomly shuffled to decrease perceptual similarity but control for the games experienced.*

C.2 Results

Training Performance

Figure C.2 shows the training scores over the course of the experiment. Both training regimes showed consistent increases in performance during training. Importantly, by the final training level, performance was still increasing and had not plateaued.

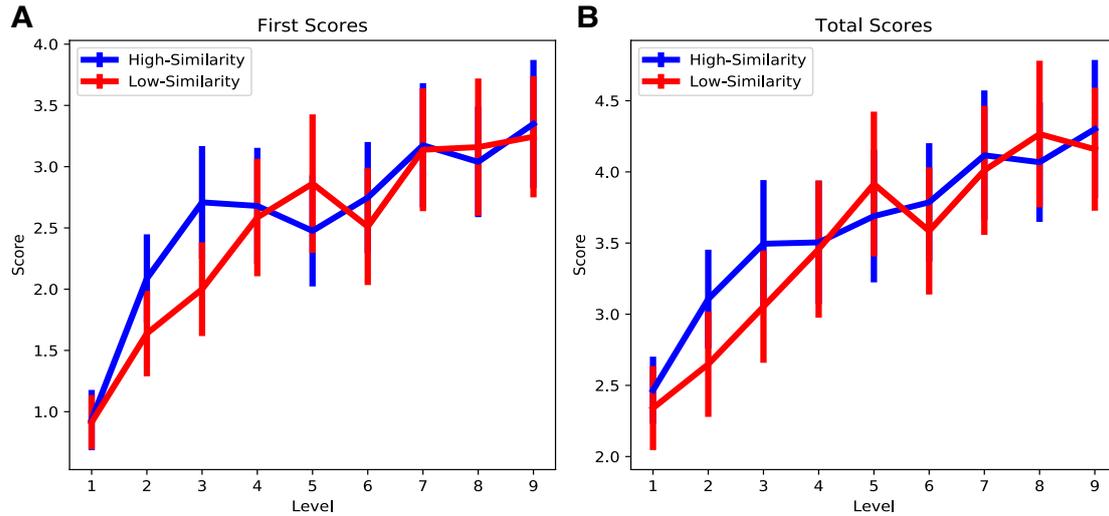


Figure C.2: Scores for each training level over the course of the experiment for the high- and low-perceptual similarity conditions. (A) First trial scores. (B) Total scores. Error bars represent 95% confidence intervals.

Figure C.3 shows bar charts of the first and total scores summed over training, while Figures C.4 and C.5 show the corresponding histograms. A Kruskal-Wallis rank sum test indicated there were no significant differences between the high- and low-perceptual similarity conditions in terms of summed first ($\chi^2(1, N=197)=0.1, p=0.700$) or total ($\chi^2(1, N=197)=0.1, p=0.719$) scores over training.

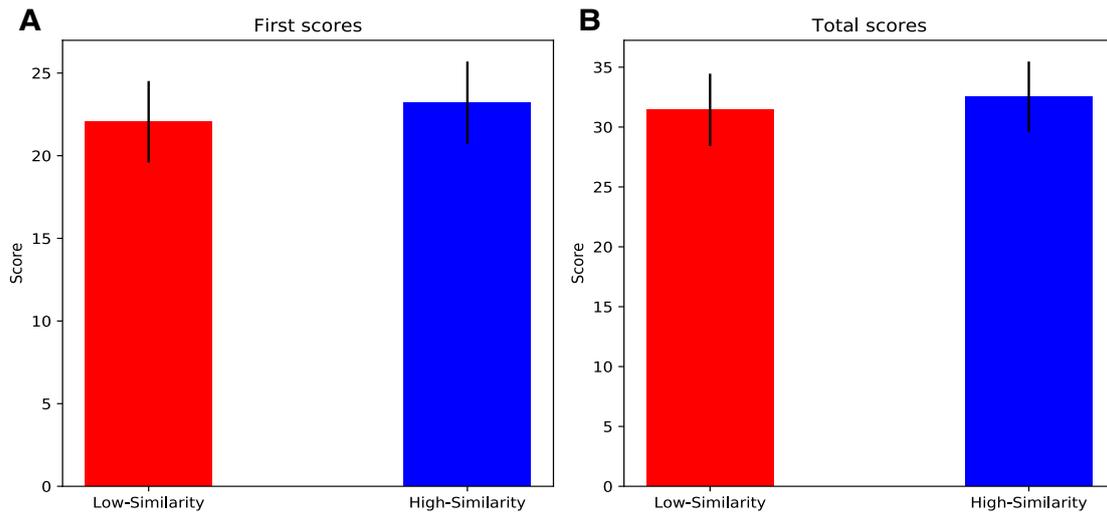


Figure C.3: Scores summed over all training levels for the high- and low-perceptual similarity conditions. (A) First trial scores. (B) Total scores. Error bars represent 95% confidence intervals.

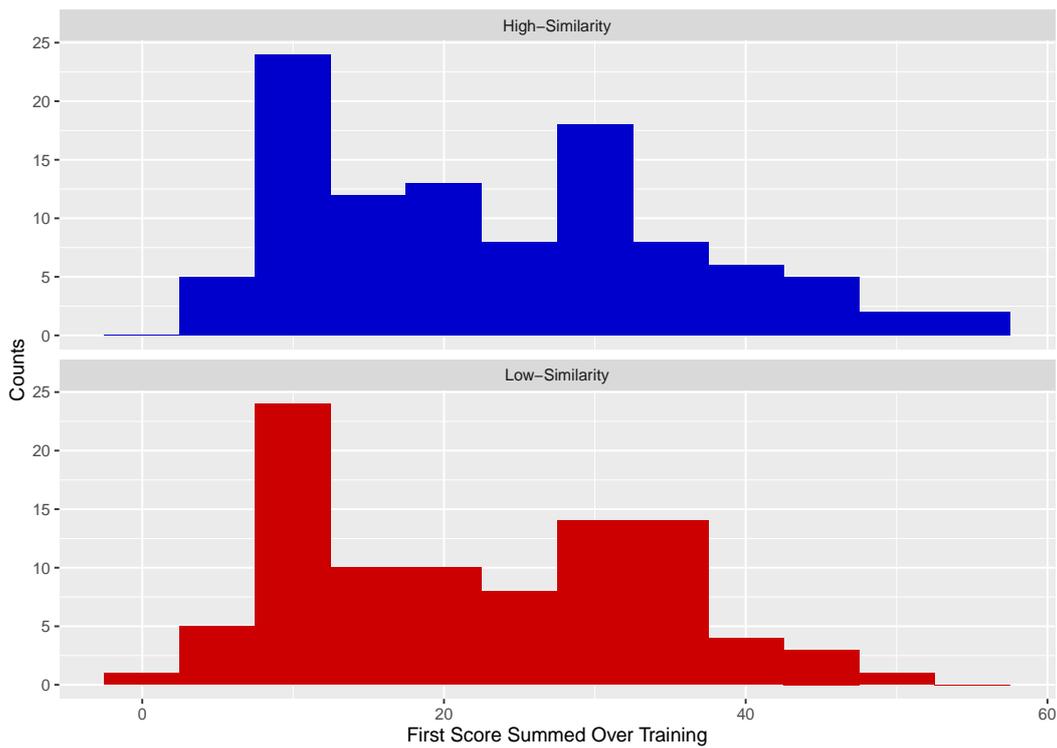


Figure C.4: Histogram of the score achieved on the first trial of a level summed over all training levels. Each colour represents a different training regime. Participants were not told the rules of the games beforehand.

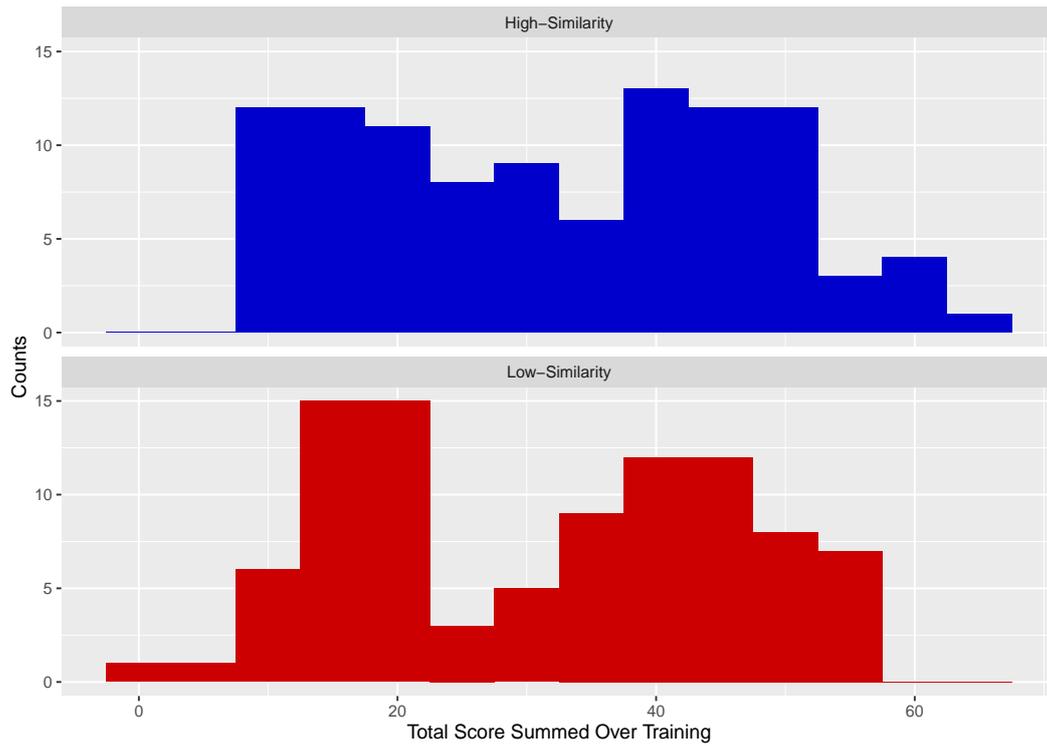


Figure C.5: *Histogram of the total score achieved during training. Each colour represents a different training regime. Participants were not told the rules of the games beforehand.*

Test Performance

Figure C.6 shows a histogram of the first scores on the final test level for each training regime. Similarly, figure C.7 shows a histogram of the total scores on the final test level for each training regime.

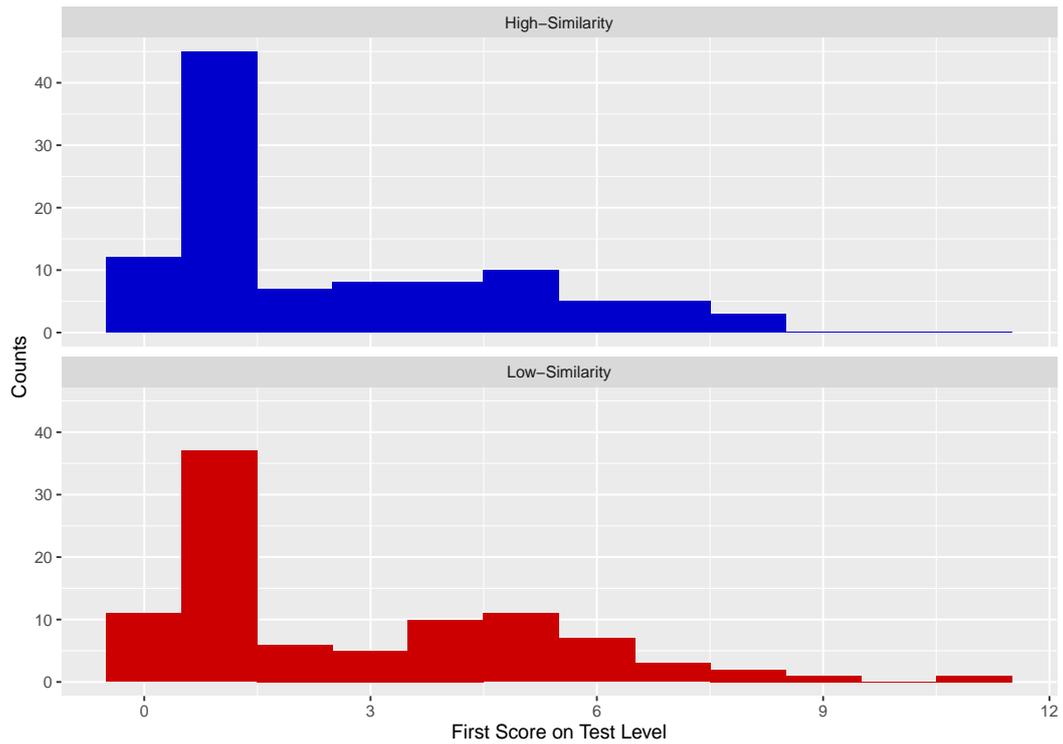


Figure C.6: Histogram of the score achieved on the first trial of the test game. Each colour represents a different training regime. Participants were not told the rules of the games beforehand.

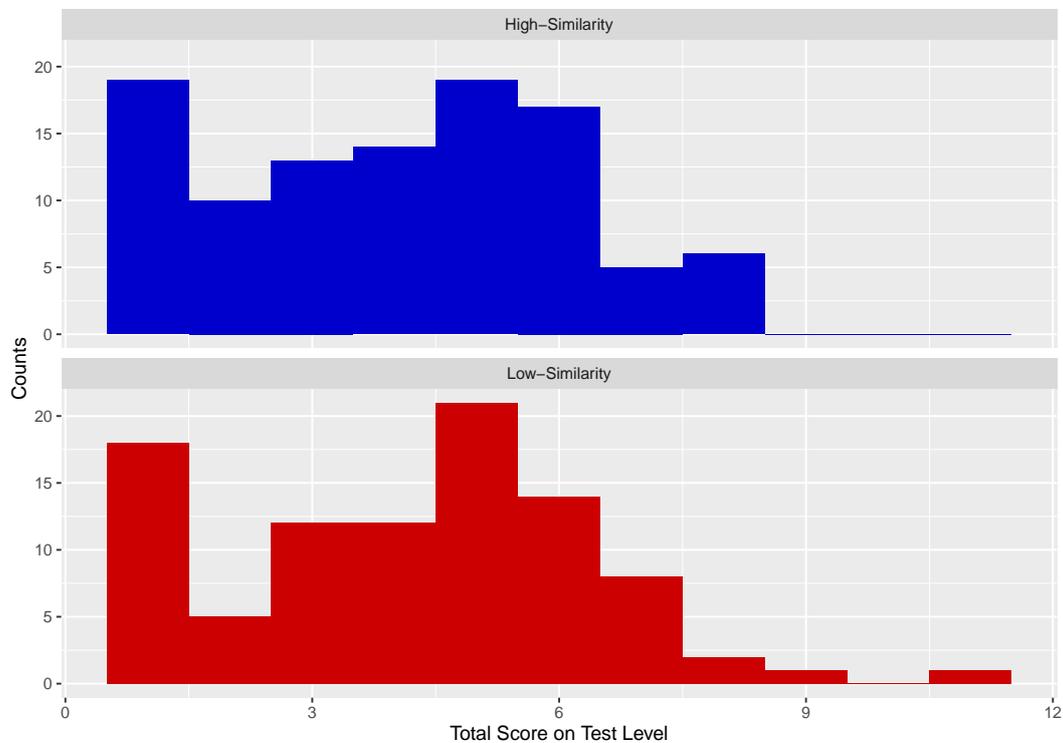


Figure C.7: Histogram of the total score achieved on the test game. Each colour represents a different training regime. Participants were not told the rules of the games beforehand.

Figure C.8 shows bar charts of the first and total scores on the final test level. A Kruskal-Wallis rank sum test between the test scores of each training regime revealed no significant differences between them in terms of first scores ($\chi^2(1, N=197)=0.4$, $p=0.518$) or total scores ($\chi^2(1, N=197)=0.2$, $p=0.666$). This suggests that the degree of perceptual similarity between consecutive levels in training does not have an effect on transfer, and that this effect is consistent even when the number of training levels is reduced.

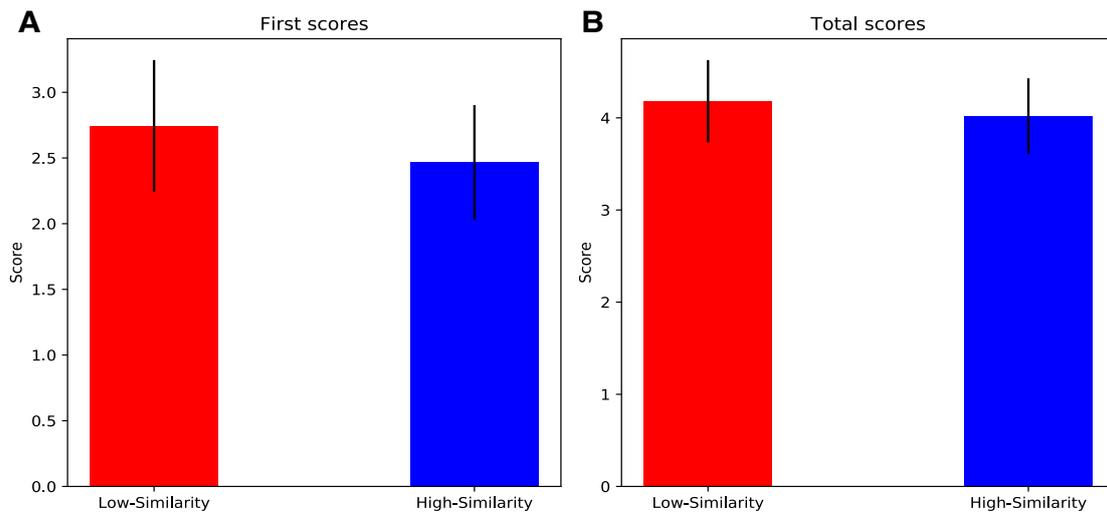


Figure C.8: *Scores on the final test game for the high- and low-perceptual similarity conditions. (A) Score achieved on the first trial of the final game (B) Total score achieved on the final game. Error bars represent 95% confidence intervals.*

As before, we performed two separate regression analyses with the first and total scores on the final test level as the dependent variables and the training regime, the self-reported video game experience, the number of texture matches during the free classification task, the performance on the first motor control task and the difference in performance between the first and second motor control tasks as independent variables. Table C.1 shows the results of these regression analyses. We saw the same pattern as in Experiment 2, with neither the training regime or number of texture matches being significant predictors of test performance. This further suggests that changing the number of training levels had no effect on the findings in Experiment 2.

Table C.1: *Regression analysis of test scores. Each column is a separate regression using either the first or total score on the test level. Values not in brackets represent beta coefficients. Values in brackets represent 95% confidence intervals.*

	<i>Dependent variable:</i>	
	First Score	Total Score
	(1)	(2)
Experience	0.10 (-0.02, 0.22)	0.16** (0.05, 0.26)
Low-Perceptual Similarity	0.23 (-0.41, 0.86)	0.13 (-0.44, 0.71)
Motor Task 1	-0.32*** (-0.48, -0.15)	-0.24** (-0.39, -0.09)
Motor Task 2 - Motor Task 1	-0.32*** (-0.48, -0.15)	-0.24** (-0.39, -0.09)
No. Texture Matches	0.03 (-0.10, 0.16)	0.02 (-0.10, 0.14)
Constant	7.53*** (3.93, 11.12)	7.29*** (4.03, 10.55)
Observations	197	197
R ²	0.11	0.12
Adjusted R ²	0.09	0.10
Residual Std. Error (df = 191)	2.24	2.03
F Statistic (df = 5; 191)	4.74***	5.39***
<i>Note:</i>	*p<0.05; **p<0.01; ***p<0.001	

Object Interactions

We again analysed the object interactions of participants over the course of the experiment to investigate what they had learnt. Figure C.9 shows the proportion of participants that interacted with a specific object for each level of the experiment. As before, we used chi-squared tests to analyse all of the object interactions.

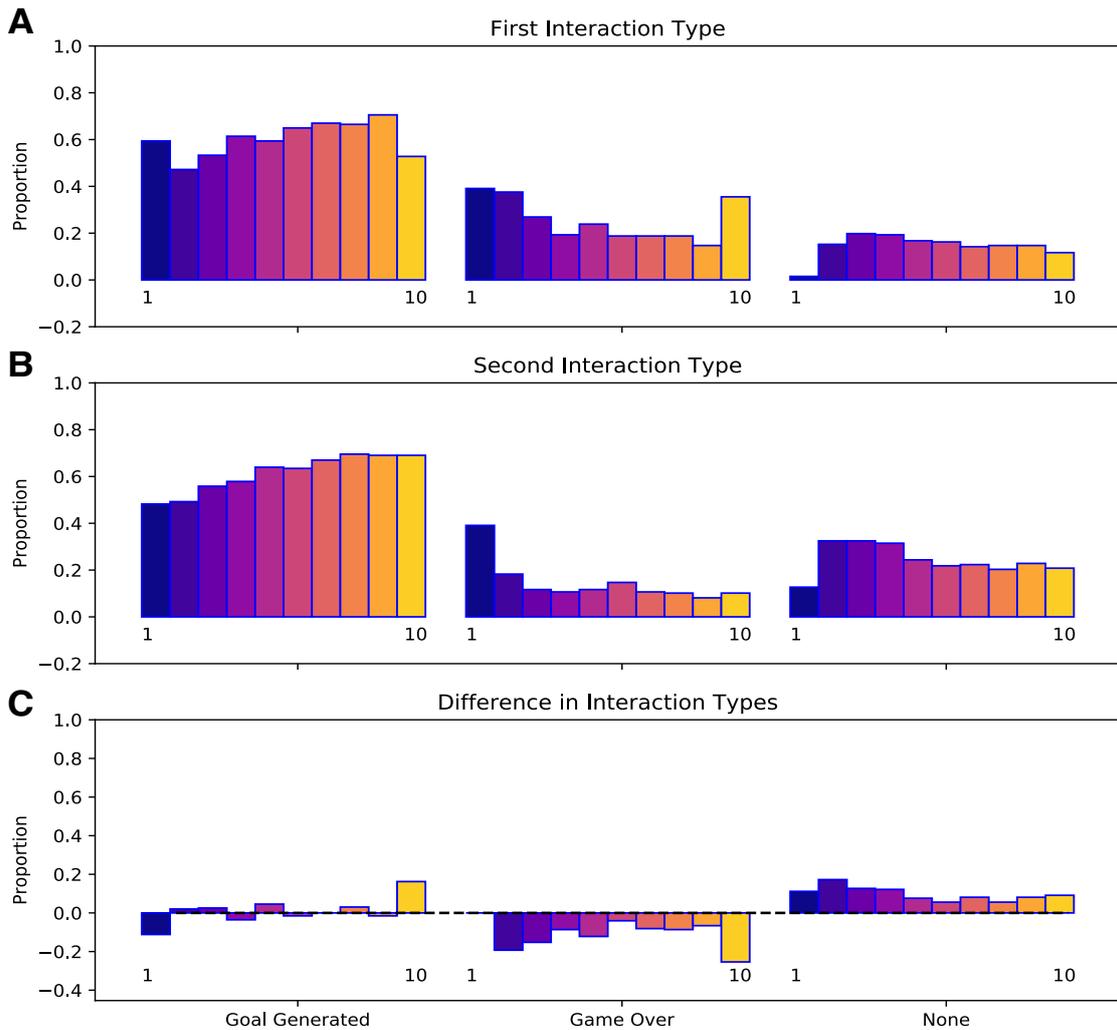


Figure C.9: *The proportion of interaction types during learning. (A) The proportion of first interaction types for all participants. An interaction is defined as a collision with another object. The x-axis indicates the object that was first interacted with (excluding the goal object) and the colour of the bars indicates the level number (blue is first training level and yellow is final test level). (B) Same as A but for second interaction types. (C) The difference between the first and second interaction types.*

We first tested whether the number of goal-generating first interactions was significantly different between the first training level and the test level. We found no significant differences ($\chi^2(1, N=394)=1.5, p=0.223$), which suggests that participants were no better on the test level at inferring which object they first needed to interact with than they were at the beginning of the experiment. Similarly, the number of game-ending first interactions was not significantly different between the first training level and the test level ($\chi^2(1, N=394)=0.4, p=0.532$). Together these results suggest that the participants were unable to learn and transfer knowledge of

the relational rules to the test level. That said, at test the number of goal-generating first interactions was significantly larger than the number of game-over first interactions ($\chi^2(1, N=394)=11.2, p=0.001$). This suggests that participants were better than chance at inferring which object to interact with first and so some degree of transfer must have occurred.

We next compared the first interactions on the test level to the second interactions on the test level. From first to second interaction we found that the number of goal-generating interactions increased ($\chi^2(1, N=394)=10.2, p=0.001$) and the number of game-over interactions decreased ($\chi^2(1, N=394)=34.6, p<0.001$). This suggests that a significant amount of ‘one-shot’ learning may have occurred, whereby participants used the first interaction to infer which object was goal-generating and which object was game-ending.

Figure C.10 shows the object interactions over the course of the experiment but split by training regime. Figure C.11 shows the object interactions split by training regime but for just the test level. Focusing on just the test level, we found no significant differences between the training regimes for both the first ($\chi^2(2, N=197)=0.3, p=0.860$) and second ($\chi^2(2, N=197)=1.8, p=0.407$) interaction types. This supports the finding that the degree of perceptual similarity during training had no effect on the participants’ ability to perform transfer and that this is robust to a decrease in the number of training levels.

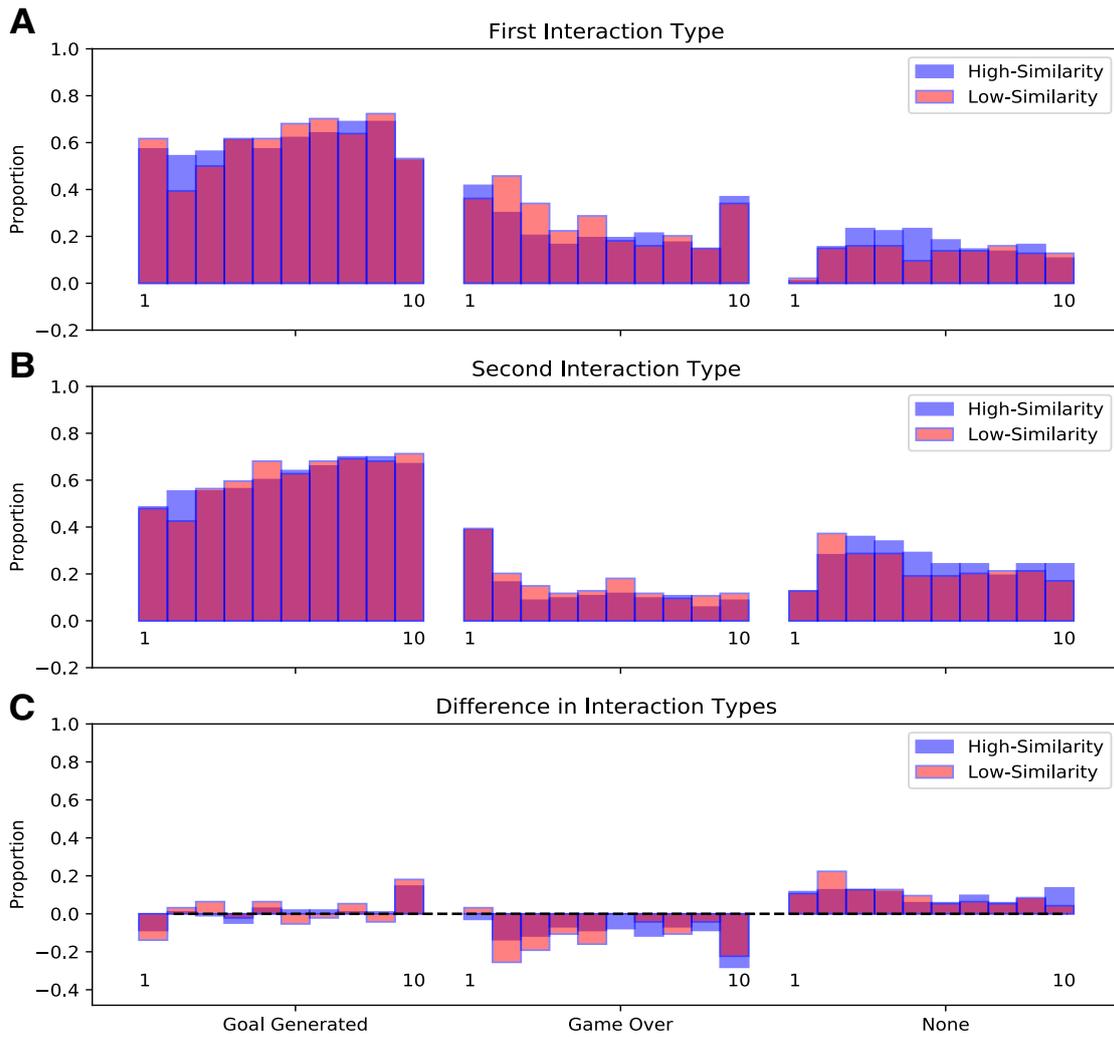


Figure C.10: *The proportion of interaction types during learning split by training regime. (A) The proportion of first interaction types for all participants. An interaction is defined as a collision with another object. The x-axis indicates the object that was first interacted with (excluding the goal object) for each level of the experiment (1-10). The colour represents the training regime. (B) Same as A but for second interaction types. (C) The difference between the first and second interaction types.*

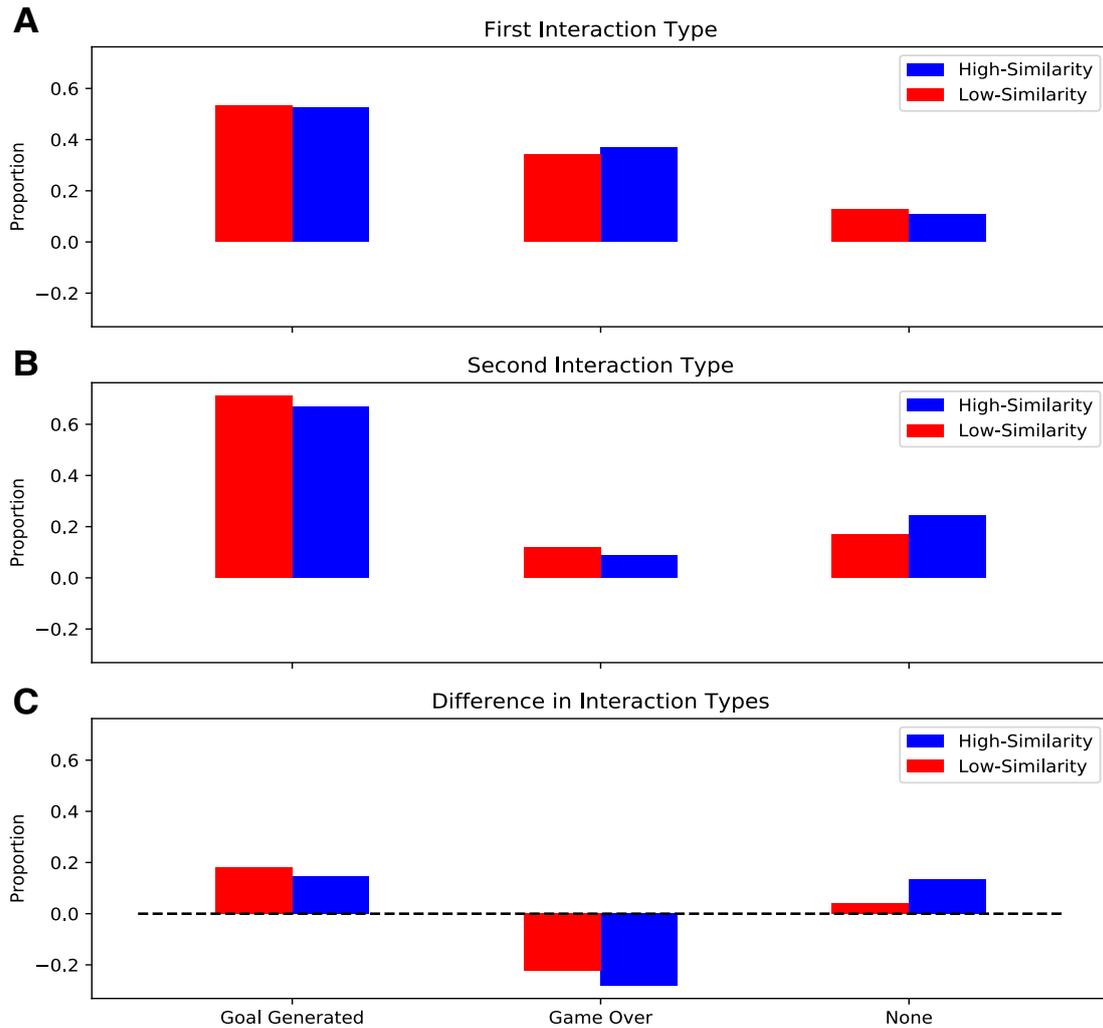


Figure C.11: *The number of interaction types for the final test level. (A) The number of first interaction types summed over all participants. An interaction is defined as a collision with another object. The x-axis indicates the object that was first interacted with (excluding the goal object) and the colour of the bars indicates the training regime. (B) Same as A but for second interaction types. (C) The difference between A and B.*

C.3 Discussion

In this Experiment we investigated whether the findings of Experiment 2 were robust to a decrease in the number of training games. We hypothesised that testing the participants' transfer ability early on in learning may reveal a critical period when the degree of perceptual similarity during training has an effect on transfer ability. We therefore repeated Experiment 2 with only 9 training levels. This meant that the test of transfer occurred while training performance was increasing at a greater

rate compared to Experiment 2. We found that the degree of perceptual similarity between consecutive training games had no effect on transfer ability, as indicated by the lack of differences between training regimes in terms of first and total test scores. A reduction in the number of training levels therefore had no impact upon our findings in Experiment 2.

Analysis of the object interactions revealed very limited evidence of ‘zero-shot’ transfer. More specifically, the number of goal-generating and game-over first interactions were not significantly different between the first training level and the test level. The test level may therefore have occurred before participants had time to discover and utilise the underlying relational rules. Instead, it appears that participants relied on ‘one-shot’ strategies, as indicated by an increase in goal-generating interactions and a decrease in game-over interactions between the first and second interactions on the test level. This suggests that perhaps early on in learning participants rely more on a ‘one-shot’ strategy and only subsequently do participants discover and utilise the underlying relational rules. It also suggests that there may be a hierarchy of strategies, whereby learning the sub-optimal strategy that one object is always goal-generating and one is always game-ending is easier than learning the relational rules.

One important aspect of our experimental paradigm is the need for exploration. Participants need to explore and interact with the goal-generating and game-over objects in order to discover that they can generate more goal objects and score more points. By reducing the number of training games we reduce the amount of time available for exploration, which reduces the likelihood of participants discovering the underlying rules. Indeed, in Experiment 2, Figure 7.26, the number of ‘None’ interactions gradually decreases over the course of the experiment as participants become increasingly aware that they can score more points. In comparison, in Experiment 2.5, Figure C.9, the number of ‘None’ interactions remains relatively high as participants do not have time to explore the rules of the game and instead exploit a conservative strategy. Future iterations of the experiment should therefore take this into account and encourage exploration. This could be done by explicitly telling participants that more than one point can be scored by interacting with other objects, or by starting each trial with the goal absent so that participants have to

interact with the other objects straight away in order to score points.

Appendix D

Supplementary Data for Experiment 3

This appendix contains supplementary data for Experiment 3 (Section 7.4). Figure D.1 shows a histogram of the score achieved on the first trial of a level summed over all training levels. Figure D.2 shows a histogram of the total score achieved during training. Figure D.3 shows a histogram of the score achieved on the first trial of the test level. Figure D.4 shows a histogram of the total score achieved on the test level.

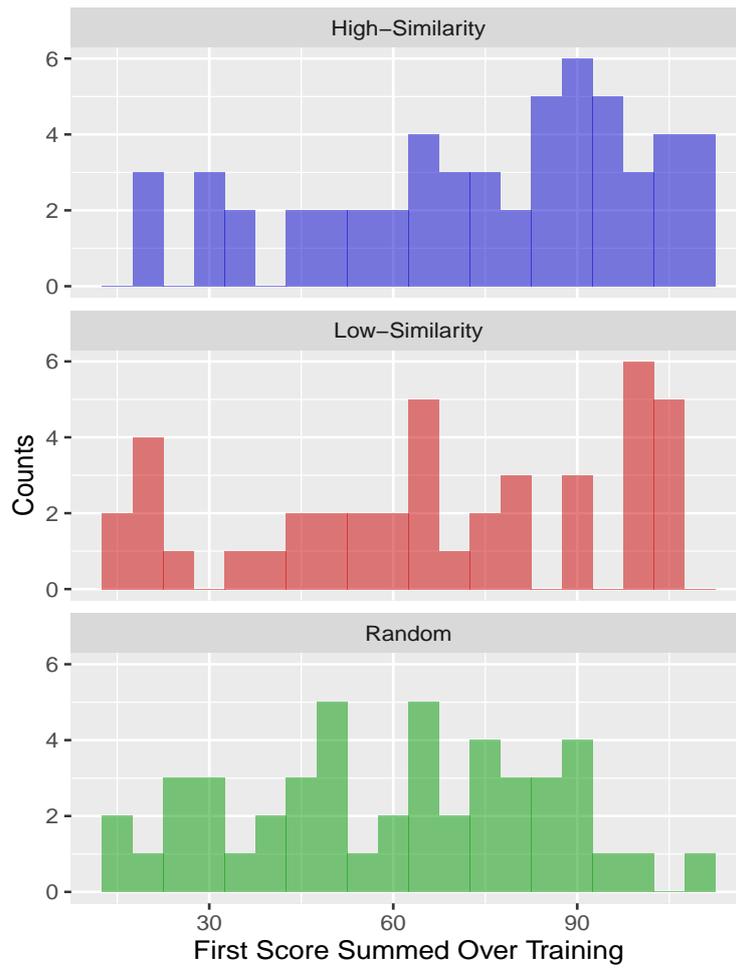


Figure D.1: *Histogram of the score achieved on the first trial of a level summed over all training levels. Each colour represents a different training regime. Participants were told the rules of the games beforehand.*

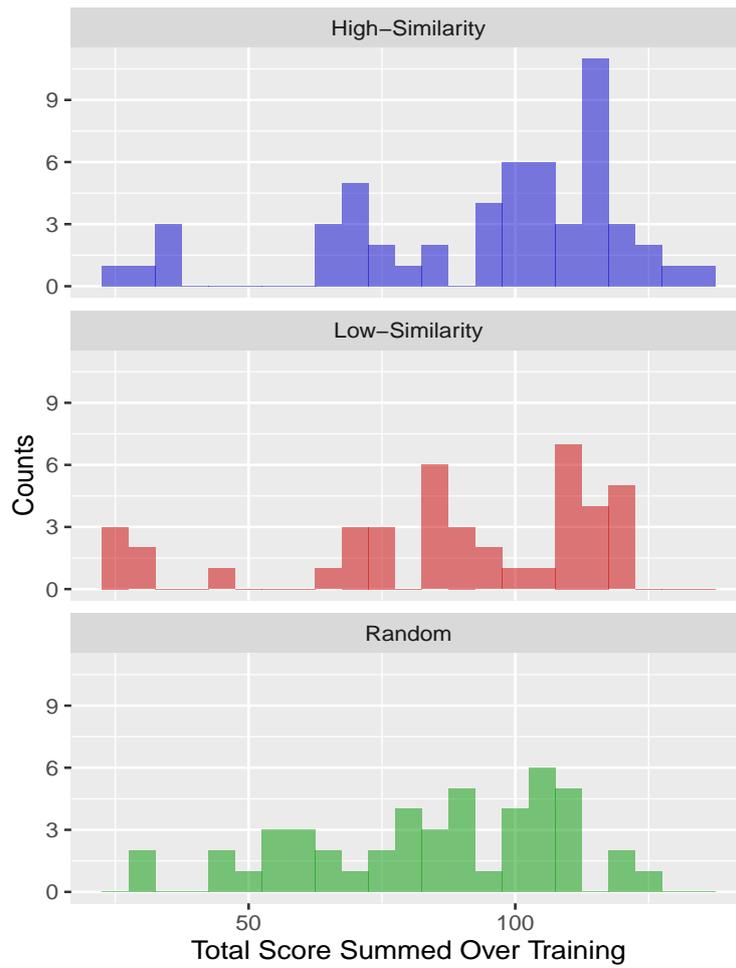


Figure D.2: *Histogram of the total score achieved during training. Each colour represents a different training regime. Participants were told the rules of the games beforehand.*

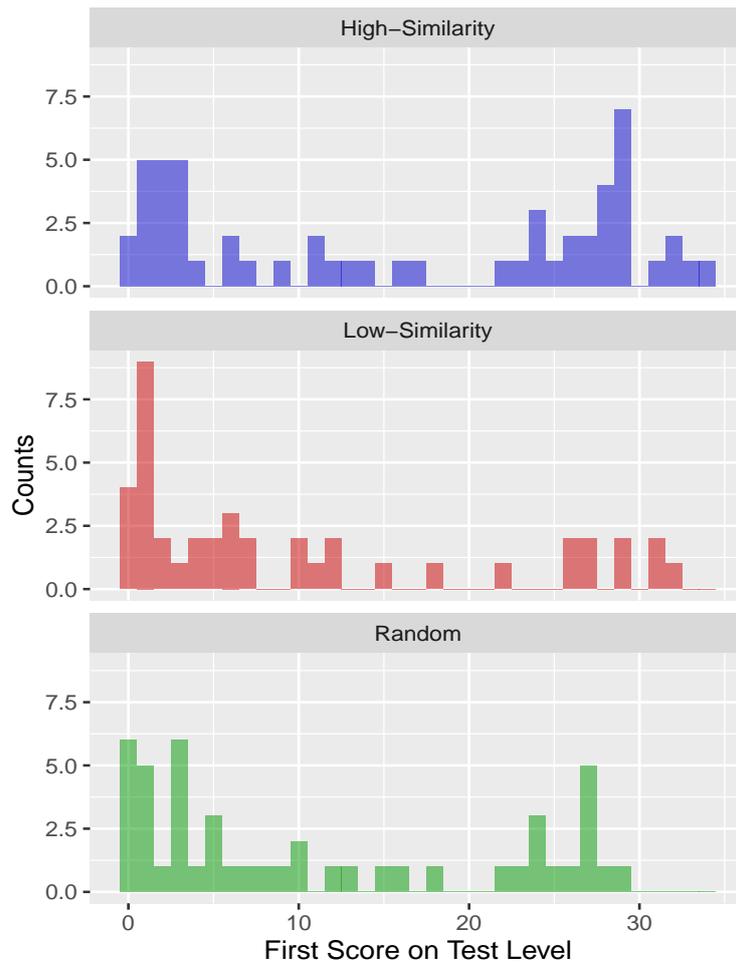


Figure D.3: *Histogram of the score achieved on the first trial of the test game. Each colour represents a different training regime. Participants were told the rules of the games beforehand.*

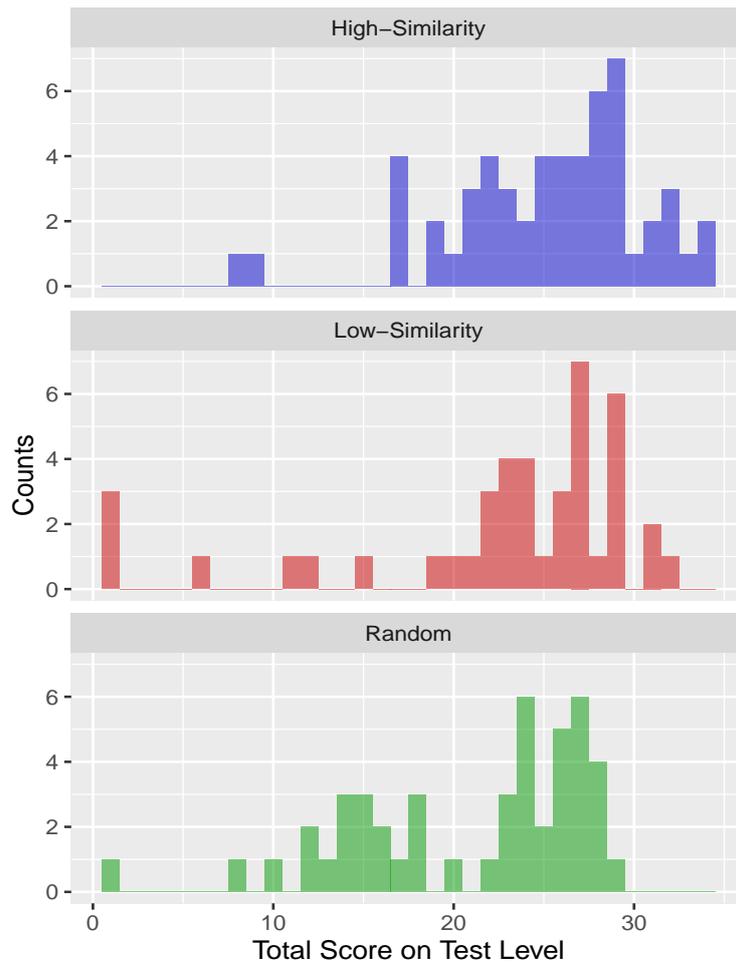


Figure D.4: *Histogram of the total score achieved on the test game. Each colour represents a different training regime. Participants were told the rules of the games beforehand.*