A second look at memory: Different Approaches to Understanding Diversity in Memory and Cognition

Sofia Alejandra Jativa Vega

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Gatsby Computational Neuroscience Unit

University College London

May 1, 2020

I, Sofia Alejandra Jativa Vega, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Memory lies at the heart of human cognitive abilities. Therefore, understanding it from neural, psychological and computational viewpoints is of key importance for computational neuroscience, psychology and beyond. In this thesis, I explore two prominent, but different, memory systems: episodic memory and working memory.

First, I propose a modification to a recent reinforcement learning algorithm for decision making in which single memories of events, i.e., episodic memories, are integrated to compute the long run value of actions. I argue that these memories are recalled and that their contributions are weighted based on context. Further, I propose that predictions made by this algorithm are combined with those that come from a standard, model-free, reinforcement learning algorithm. I suggest that humans can flexibly choose between these two sources of information to make decisions and guide actions. I show that the resulting combined model best fits data on human choices, outperforming previously proposed models.

To complement these algorithmic and psychological suggestions, I present a generative model of the world according to which this sort of episodic recall is an appropriate method for making inferences and predictions of future rewards. Contrary to other suggestions for reward-based learning, this generative model can model events that not only drift continuously in time, but can also suddenly change to new or repeated events.

Turning to working memory, I use information theoretic analyses to show that dynamic synapses, whose strengths adjust with usage, can increase its capacity. I argue that these components should be included in the study of working memory. The thesis ends with an explanation of the connections between these memory systems.

Impact Statement

In this work, we explored some of the main concepts in the computational neuroscience literature: working memory and episodic memory. The first part of the thesis explores the role of episodic memory in decision making. We show that this memory system is very important for learning and guiding actions; however, its role in reward-based decision making has not been widely explored. Here, we propose a new reinforcement learning algorithm based on episodic memories that are recalled and weighted by context. We show that, in combination with conventional reinforcement learning algorithms, it predicts human choices better. Furthermore, we propose a generative model for which this learning algorithm is an adequate inference model. In this way, we are able to provide a solid statistical framework within which to study and analyze the capabilities of the proposed learning algorithm.

The impact of this new algorithm extends from theoretical Reinforcement Learning, to the modeling of human decision making, to improving the capabilities of artificial agents. This algorithm proposes the integration of episodic memory in reinforcement-based decision making frameworks, and it proposes a formal statistical framework in which to study theoretical questions and explore different environmental statistics and tasks where episodic memory is relevant. This algorithm can also be used to train artificial agents. In addition to other learning algorithms, artificial agents can use our new proposal to improve their learning skills and decision making. Integrating episodic memory learning among other learning algorithms for artificial agents would allow them to have greater flexibility and possibly improve their generalization capabilities. This model can also be used to study human cognition. In this thesis we showed that episodic memory is a key component of human decision making and consequently of human intelligence. Finally, this model can be used to study biases in recall and decisions in patients with mental disorders. For example, it could be used to study how depressed people recall episodic memories (often biased towards negative memories), and how this affects their decisions. This study could then be used to develop treatments targeted towards improving this recall process. In conclusion, this model proposes a new reinforcement learning framework that can be used to study episodic memory, training of artificial agents, and could be a starting point for further studies with clinical applications.

The second part of the thesis explores the neural substrates of working memory. Understanding working memory is key to understanding human cognition. One of the big open questions regarding working memory is how it is implemented by neural circuits, i.e., its neural substrate. The most widely accepted theory claims that working memory depends on the recurrent connections between neurons. Based on previous experimental observations, in this thesis, we suggest that the synapses that realize these connections have short-term dynamic states and should also be considered part of the neural substrate of working memory. We showed that by including dynamic synapses in a working memory model, the network's capacity increased. This work is significant since understanding working memory capacity is key to understanding the limits of human intelligence, and indeed brain-inspired algorithms. One of the big differences between human brains and machines is our memory capacity constraints. In order to understand the computational model of the human brain, we need to bound our algorithms according to our capacity limitations. For this reason, understanding the neural substrates of working memory is an important component for this endeavour. Consequently, the results of this thesis are an important contribution to solving the puzzle of how the human brain works.

Acknowledgements

Doing a PhD at the Gatsby Unit has been an invaluable experience for me, and I am very thankful to all the people who have supported me on this journey. First of all, I would like to thank my supervisor, Peter Dayan, for his support and guidance throughout these years. It has been a great opportunity to be able to work with one of the leading minds in the field. I would also like to thank the faculty and members of the Unit for always been there to discuss and challenge any scientific project or curiosity. It is thanks to the environment they created that I have been able to grow as a scientist. In particular, I would like to thank my friends Joana Soldado Magraner and Eszter Vertes for their personal and professional support. Finally, I would like to thank my family for always believing in me.

Contents

List of Figures

List of Tables

Chapter 1

Introductory Material

In order to improve their chances of survival, living beings have had to adapt to the continuous wealth of temporal and spatial time scales of their environments. To do so, organisms have developed a diverse set of mechanisms for storing and retrieving information. These mechanisms reflect the statistics of the environment necessary to guide actions – for example, the frequency of occurrence of a stimuli or the coincidence of stimuli and reward [Sherry and Schacter, 1987] [Miyake and Shah, 1999], [Tetzlaff et al., 2012]. These mechanisms form the basis of learning and memory, and give rise to a complex and diverse set of memory systems. These systems may be characterized based on the different time scales at which they operate and the interactions between information storage and retrieval.

Understanding memory is key to understanding cognition and intelligence. However, the study of memory has never been straightforward. From psychologists to neuroscientists to physicists, scientists from different fields have taken diverse approaches to address this subject [Miyake and Shah, 1999]. On one hand, memory has been studied as a theoretical construct that explains key cognitive phenomena; on the other hand, the complexities of the physical and a biological processes underlying these phenomena have fallen short of explaining the richness of subjective experience. To complicate matters further, there is no one agreed definition or set of characteristics outlining what memory should be [Baddeley, 2012].

Evidence in cognitive psychology and neuroscience shows that there is more than one type of memory systems [Squire et al., 1984]. The term memory systems refers to different systems, often associated with different neural substrates, that have different mechanisms for storage, retention and retrieval of memories and often times process different types of information [Tulving, 1985]. This specialization of memory systems developed to solve tasks with properties and characteristics for which alternate systems were incompatible [Sherry and Schacter, 1987]. An example of this can be seen in the distinctions between incremental habit formation and episodic memory. Both types of memories are important for guiding appropriate behaviour; however, they operate at different time scales and storage mechanisms that seem mutually incompatible. Habit formation requires gradual or incremental learning, while episodic memory requires a rapid one-trial learning [Sherry and Schacter, 1987].

The modern concept of multiple memory systems started with the case of the amnesic patient H.M. He suffered from epilepsy, and in an attempt to cure it, he had bilateral medial temporal lobotomy. With it, he had removed two thirds of his hippocampi, parahippocampal cortices, entorhinal cortices, piriform cortices, and amygdalae [White and McDonald, 2002] [Corkin, 2002]. As a consequence, he developed anterograde amnesia, which is the loss of the ability to create new memories, and moderate retrograde amnesia, which is the inability to remember events in a given period right before the onset of amnesia. However, he was still able to complete other tasks like solving puzzles, and was able to recall events from long term memory. These abilities suggested that other types of memory were intact and mediated by different brain regions. This case jump-started the study of memory systems, and many have been identified ever since.

At present, we understand memory as comprised of many distinct but interacting systems. Here, we present a brief simplified overview of the current taxonomy of memory systems, which is sufficient for the purposes of our thesis. This taxonomy separates memory into long term memory (duration of hours to years) and short term memory (duration of a few seconds). When information is been attended to in short term memory, it is referred to as working memory. Long term memory has been further subdivided into explicit or declarative and implicit or non-declarative.

Declarative memory is the type that can be consciously recalled. It encompasses semantic and episodic memory, which each refers to the memory of facts and events respectively. Non-declarative memory encompasses priming, procedural, associative learning, and non-associative learning. Priming is the memory type that is expressed unconsciously through the performance of certain tasks. It refers to the unconscious effect of the exposure to a word or object on a subsequent word or object, even when that first word or object are not consciously remembered. Procedural memory refers to skills and habits. Associative learning refers to classical and operant conditioning. Finally, non-associative learning is associated with reflex pathways [Squire, 2004] [Tulving and Schacter, 1990] [Thompson and Kim, 1996] [Squire and Zola, 1996]

The study of memory is very complex, and not straightforward. There are different levels of complexity and open questions that span from the neural substrates to the cognitive and psychological mechanisms of each memory system. For that reason, we follow David's Marr approach and separate the study of memory into three levels of analysis: the computational level, the algorithmic or representational level and the implementation or physical level [Marr and Poggio, 1976] [Marr et al., 1991]. The computational level refers to the study of the problem a system has to solve, the algorithmic level focuses on how the system solves that problem, and finally the implementation level studies how the system implements this solution. In terms of memory research, the computational level refers to the information processing goal of the memory system: i.e. what is the optimal way to store and recall information. The algorithmic level refers to the actual operations carried on during these information processing task: i.e what algorithms are used to store and retrieve information. Finally, the implementation level refers to the way networks of neurons and synapses perform these operations.

In this thesis, we focus on two memory systems: episodic memory and working memory. We use different levels of analysis to address open questions in each of them. By using different levels of analysis, we hope to separate different issues relevant to the two memory systems into separate specific questions. It might seem that the thesis comprises of two completely disparate parts. However, both memory systems studied are primarily concerned with the storage and recall of information about particular events rather than compressed sufficient statistics [Wenger and Shing, 2016]. The first part concerns the computational and algorithmic issues associated with longer term storage. Specifically we ask how individual events are stored for later retrieval when a similar situation occurs and an action needs to be taken. In this part, we look at usage of this type of memory during reward learning. In particular, we study how information retrieval from episodic memory guides reward based learning and decision making. To achieve this goal, we incorporate an episodic memory model into a reinforcement learning framework. We test this theory with human data, and we propose a generative model for which our algorithm is a suitable approximate inference method. The second part of the thesis explores the neural substrates of short term storage. To answer this question, we look at the implementation mechanisms of neural systems to retain information actively for short periods of time while an animal makes a decision or performs an action. The importance of this analysis lies on the fact that the neural mechanisms of this memory system constrain the capacity or number of items that can be stored in short term memory. Properly answering this question can reveal the physical limits of working memory, and by doing so give answers to many open question on the field. For example, we could understand some critical limits to human capacities and intelligence.

In short, this thesis explores two memory systems concerning the storage, maintenance and retrieval of one type of information (specific events) at two different time scales (short-term and long-term). Marr's approach becomes useful in conceptualizing the different open questions regarding these two systems. Taken together, the two parts of the thesis constitute a significant contribution to the understanding of one of the pillars of human cognition: human memory.

Chapter 2

Episodic Memory in Reward Based Learning

2.1 Motivation

Reinforcement learning (RL) is a learning framework for modeling adaptive decision-making based on reward history [Sutton and Barto, 2018], [Sutton and Barto, 1998], [Dayan and Niv, 2008]. Its goal is to optimize performance, measured in terms of cumulative reward, large by means of learning to predict the consequences of actions. Reinforcement learning has been able to both explain natural systems and build artificial ones [Sutton and Barto, 2018].

The two main categories in RL are called model-free and model-based reinforcement learning. These two categories were originally defined and are often treated as separate; however, it has been shown that human behaviour often lies somewhere in between both of them [Sutton and Barto, 1998],[Sutton and Barto, 2018]. A model-free learner chooses actions based on previously computed and stored estimates or cached values of the future utility of actions. The action with the highest expected future utility is chosen – up to stochasticity. These estimates are computed based on trial and error experiences, and are typically equal to running averages of past rewards, where each reward is weighted by recency. These values are updated at each time step using the difference between the expected reward and the reward received (known as prediction error in the RL literature) as a guidance[Sutton, 1988, Watkins, 1989]. Model free learning is computationally inexpensive in learning and use, but also statistically inefficient. This means that most information about the environment and task structure is lost, since the only information kept is a scalar value of the long run value of actions. Among other areas, the dorsolateral striatum and parts of the amygdala have been associated with this type of learning [Balleine, 2005], [Killcross and Coutureau, 2003].

On the other hand, a model-based learner learns and stores a model of the environment (transition probabilities between states and rewards) based on previous observations [Daw et al., 2005]. This model is then used to plan the consequences of actions before making a decision. A model-based learner computes long run values of actions by operations equivalent to running forward this learned mental model of actions and rewards whilst summing the latter. This type of learning uses experience in a statistically efficient manner [Dayan and Niv, 2008]. Contrary to model free learning, it is flexible, and it can quickly adapt after sudden changes. This flexibility is due to the fact that a model-based learner stores all information from the environment, so it can plan forward and if presented with new information, it can instantly change its behaviour at future states. In this way, it allows for goal-directed decision making [Dickinson and Balleine, 2002]. Model-free cached values are updated slowly and carry the long run average information from the past; therefore, a model-free learner requires more time to adapt to changes. Model-based RL is associated with the dorsomedial striatum, prelimbic pre-frontal cortex, orbitofrontal cortex and parts of the amygdala [Rushworth and Behrens, 2008],[Balleine, 2005],[Dolan, 2007], [Matsumoto and Tanaka, 2004].

We consider a third type of reinforcement learning methods by introducing a new algorithm that uses episodic memories, cued by context, to choose actions. We call this new algorithm: Contextual Episodic. Episodic memory is a distinct and unique memory system that stores and retrieves information from single past events [Tulving and Schacter, 1990]. Contrary to short-term or working memory, it can store information for than a few minutes, and contrary to other forms of long term memory, such as semantic memory, it doesn't store summary statistics, but rather, it stores and recalls specific events. As all other memory systems, episodic memory presumably evolved [Sherry and Schacter, 1987], because the type of information it provides is useful for organisms to survive and maximize payoffs. We hypothesize that this memory system exists, because rare events can happen again, and in those situations, knowledge of that specific event is more useful than accumulated statistics from many non-relevant events. For this reason, and for the fact that humans draw on memory to compute decisions flexibly [Bornstein et al., 2017] [Bornstein and Norman, 2017], we propose a decision making framework that utilizes episodic memory. This memory system is used daily by humans; however, much of mainstream reinforcement learning has ignored it, with some important exceptions: [Bornstein et al., 2017] [Bornstein and Norman, 2017] [Gershman and Daw, 2017]. We show that by taking advantage of this memory system, it is possible to capture human behavioral data more accurately.

In fact, episodic memory can be used in the context of both model-based and model-free learning. Either the model of the world from a particular situation, or a summary statistic from it can be stored and retrieved as a single episode. In our work, we use episodic memory to replace (or to be combined with) model-free cached values. In this chapter, we show that such an episodic based learning algorithm fits human data, suggesting that humans might use this memory system for decision making. The particulars of the situations in which such an algorithm might be preferentially powerful is the topic of our next chapter.

Previous models that have studied the role of episodic memory in decision making include the work by Lengyel and Dayan [Lengyel and Dayan, 2008] and Bornstein et, al [Bornstein et al., 2017]. Dayan and Lengyel described situations where episodic memory can be more useful than model free or model based systems [Lengyel and Dayan, 2008]. Their work focused on a cost/benefit analysis that paralleled previous work on the trade-offs between model free and model based learning [Daw et al., 2005]. They found that when there isn't enough gathered information to have reliable model-free cached values, as well as when the environment is characterized by high complexity and inferential noise, retrieving single episodes is the most efficient way to use previous experiences [Lengyel and Dayan, 2008].

Bornstein et al. [Bornstein et al., 2017] proposed a model that describes how episodic memories can be used in a decision making framework. They suggested a sampling model, where individual events are sampled and used to make decisions. We expand on this model in the literature review section.

Gershman and Daw proposed that episodic memories can be used to construct non-parametric approximations to state, actions or state-action value functions [Gershman and Daw, 2017]. They proposed that individual episodes could be retrieved and averaged, based on a similarity function between current and retrieved states, at the moment of making a decision or choosing to perform an action. In this way, they suggested that episodic memory could be used to alleviate some of the open questions in RL such as computing values when environments are non-Markovian and have long time dependencies. An RL algorithm endowed with episodic memory can use raw data stored in Episodic memory, and retrieve the necessary episodes, including those with long temporal dependencies, to compute the value of state or actions. They proposed a theoretical framework, called Episodic RL that uses a kernel function to measure the similarity between states. Therefore, Episodic RL is non-parametric, which frees it from relying on a parameterized version of the value function, and in turns allows it to grow with observed data [Gershman and Daw, 2017].

Our Contextual Episodic learning model builds on Bornstein's model, and extends Gerschman and Daw's proposal so that episodic memories are recalled and combined weighted by contextual similarity. Our algorithm is also a non-parametric model based on a kernel function, but it defines similarity as the similarity between temporal contexts. Similarly, Bornstein uses episodic memories to make decisions, but its bases recall on recency rather than any similarity between states or context. It has been shown that context influences what events are recalled [Duncan and Shohamy, 2016], and that these events - cued by contextual recall - influence current decisions [Bornstein and Norman, 2017]. This finding demonstrates the importance of episodic memories in decision making, but furthermore, it highlights the relevance of context in decision making - through its influence in episodic recall. For this reason, we considered it crucial that our model is based around context. Furthermore, it has been shown that a similarity-based approach to recalling specific events from memory leads to higher performance in learning agents [Plonsky et al., 2015], and behavioural economists have developed a theory, where decision makers are endowed with a similarity function that compares previous states. This theory is called case-based decision theory [Gilboa and Schmeidler, 2001]. For these reasons, our algorithm uses context to recall events, and furthermore, it uses a contextual similarity function to weight the events it recalls. In this way, our algorithm allows for flexible recall of relevant information, measured by contextual similarity between events, that can be be used for reward learning and decision-making. Finally, our algorithm uses a Temporal Context Model (TCM) [Howard and Kahana, 2002] to model the temporal evolution of context. TCM was developed to understand free recall in humans, and we use it as a key component of our algorithm.

In this chapter, we further propose a Hybrid model between our Contextual Episodic model and standard model free learning. Learning frameworks do not happen in isolation. They often cooperate or compete [Gershman and Daw, 2017] [Daw et al., 2005] [O'Doherty et al., 2003] [Keramati et al., 2011], which imply that humans use a combination of techniques to make decisions. For this reason, we hypothesize that when humans make decisions they use a weighted combination of at least two decision making methods. For the purpose of this chapter, we focus on Contextual Episodic and model-free frameworks. Our model allows for flexibility in the utilization of information. Contextual cues are always triggering the recall of episodic memories [Duncan and Shohamy, 2016], and it is only at the time of decision that we select which information is relevant for our actions.

In the following sections, we review the details of Bornstein's sampling model and the TCM model, on which we base our proposal. Then, we formally introduce the Contextual Episodic Framework and the Hybrid model. We finish with an analysis of the Hybrid model, and how it compares to other models using both artificial data and data from previous human experiments.

2.2 Literature Review

2.2.1 Previous Models

Prior to reviewing previous models from the literature, we introduce the main concepts of Reinforcement Learning and set up the decision making problems we study. Reinforcement learning models study how decision making agents learn to interact with their environment in order to achieve a goal (i.e. maximize long term rewards). To achieve this goal, agents learn to choose actions, often based on, or at least influenced by, judgments of their long-run value. Different Reinforcement Learning models have different methods to estimate these values [Sutton and Barto, 1998].

The most common and simplest decision making problem studied in the Reinforcement Learning literature is the n-bandits task. A bandit is a slot machine, which is operated by pulling a handle. The decision task consists on choosing which bandit to operate. The decision to operate a specific bandit among n different bandits is called an action. After choosing a bandit (taking an action), a reward is received. The learning agent has to perform a sequence of actions to maximize reward received over time. As explained, a standard reinforcement learning agent chooses actions based on the estimated value of each action a at each time step t – denoted as $Q_t(a)$ [Watkins, 1989]. There are different methods for estimating the value of actions, and for selecting an action based on these estimates. In this thesis, we focus only on the methods for estimating action values. For action selection, we use a stochastic policy, or strategy for action selection, where an action is selected following a soft-max probability distribution. The equation below describes this probability distribution. In this equation, a and b are two actions corresponding to two bandits in a two-bandits task. The two probabilities add up to one, and their values vary from 0 to 1. The inverse temperature parameter β determines how deterministic the probability functions are with respect to action values. For large β, the probability is almost deterministic, while for small β, actions are more random.

[Sutton and Barto, 1998] [Dayan et al., 2003]:

$$P(a) = \frac{e^{\beta Q_t(a)}}{e^{\beta Q_t(a)} + e^{\beta Q_t(b)}} \tag{2.1}$$

$$P(b) = \frac{e^{\beta Q_t(b)}}{e^{\beta Q_t(b)} + e^{\beta Q_t(a)}} \tag{2.2}$$

The true value of each action is equal to the mean reward that would be received if that action is taken at that time step by the agent - this value is unknown for the agent. The agent can only form estimated values for each action. Informally, the estimated value of an action depends on the history of rewards received after taking the action. Different reinforcement learning frameworks compute this value differently based on how the information from previous rewards is utilized, and the assumptions of how the environment changes [Sutton and Barto, 1998] In the case of stationary bandits, where rewards values are not changing over time, the simplest estimation method is the sample average method. This method says that if at time t, action a has been chosen $N_t(a)$ times with rewards $r_1, ..., r_{N_t(a)}$, the estimated value for action a is [Sutton and Barto, 1998] :

$$Q_t(a) = \frac{r_1 + \ldots + r_{N_t(a)}}{N_t(a)} \tag{2.3}$$

In order to avoid exponential memory and computational cost, an incremental approach was derived, where only an estimate of the value of action a for the t_{th} reward received $Q_t(a)$ is maintained and updated when the next reward is received. The derivation of this incremental equation is the following. In the following equation, $Q_{t+1}(a)$ is the average of all previously received rewards for that action, and i is the index of previous rewards [Sutton and Barto, 1998] :

$$Q_{t+1}(a) = \frac{1}{N_t(a)} \sum_{i=1}^{t} r_i \tag{2.4}$$

$$Q_{t+1}(a) = \frac{1}{N_t(a)} \left(r_t + \sum_{i=1}^{t-1} r_i \right) \tag{2.5}$$

$$Q_{t+1}(a) = \frac{1}{N_t(a)}\left(r_t + (t-1)Q_t(a) + Q_t(a) - Q_t(a)\right) \tag{2.6}$$

$$Q_{t+1}(a) = \frac{1}{N_t(a)}\left(r_t + tQ_t(a) - Q_t(a)\right) \tag{2.7}$$

$$Q_{t+1}(a) = Q_t(a) + \frac{1}{N_t(a)}\left[r_t - Q_t(a)\right] \tag{2.8}$$

In these equations $\frac{1}{N_t(a)}$ is the step-size of each update. For non-stationary problems, where rewards are changing over time, a constant step-size α is used instead. For non-stationary problems, it makes sense to weight the most recent rewards more than the long past ones. However, as we will explain later on, the proposal of this thesis is that this is not always the case. Using this constant α, the update equations are equal to [Sutton and Barto, 1998]:

$$Q_{t+1}(a) = Q_t(a) + \alpha\left[r_t - Q_t(a)\right] \tag{2.9}$$

$$Q_{t+1}(a) = \alpha r_t + (1-\alpha)Q_t(a) \tag{2.10}$$

$$Q_{t+1}(a) = \alpha r_t + (1-\alpha)\left[\alpha r_{t-1} + (1-\alpha)Q_{t-1}(a)\right] \tag{2.11}$$

$$Q_{t+1}(a) = \alpha r_t + (1-\alpha)\alpha r_{t-1} + (1-\alpha)^2 Q_{t-1}(a) \tag{2.12}$$

$$Q_{t+1}(a) = \alpha r_t + (1-\alpha)\alpha r_{t-1} + (1-\alpha)^2\alpha r_{t-2} + \cdots + (1-\alpha)^{t-1}\alpha r_1 + (1-\alpha)^t Q_{t=1}(a) \tag{2.13}$$

$$Q_{t+1}(a) = (1-\alpha)^t Q_{t=1}(a) + \sum_{i=1}^{t} \alpha(1-\alpha)^{t-i} r_i \tag{2.14}$$

This new update rule defines a weighted running average. It is a weighted average, because $(1-\alpha)^t + \sum_{i=1}^{t} \alpha(1-\alpha)^{t-i} = 1$. As can be seen, this is a different method for estimating the value of actions. The information from previous rewards is not weighted equally. Each reward received at a different time step has a different weight. For example, in the equations above, each reward R_i is weighted by $\alpha(1-\alpha)^{t-i}$. This weight decays exponentially the further in the past reward R_i was received. The reason for this is that the term $(1-\alpha)$ is less than one, and consequently it decays exponentially with the value of the exponent $t-1$. For this reason, this weighted average is called recency-based weighted average[Sutton and Barto, 1998].

For the purposes of this chapter, we analyze the learning frameworks for a two-armed bandit task with non-stationary rewards. The learning agent has to choose between the two bandits to maximize rewards. We use the recency-based weighted average described above as our baseline model for comparison. This method was previously derived by Widrow and Hoff as the delta rule [Widrow and Hoff, 1960] and in classical conditioning to account for conditioned responses rather than actions. In classical conditioning this rule is known as the Rescorla-Wagner learning rule [Rescorla et al., 1972] (Here, we refer to this rule with the initials RW). In this chapter, when we use this method to compute action values, we name the estimates $Q_t^{RW}(a)$, and we use equation 2.9 to compute these estimates.

In this chapter, we also investigate a variation to the Rescorla-Wagner equation, and use it for the comparison analysis we perform in this chapter. We call this variation the Time Constants model (abbreviated as TC), because it is similar to the Rescorla-Wagner learning rule, but it integrates information using two learning rates. The Q values estimated are a weighted combination between two Q value estimates, each following a RW rule with a different learning rate. In this way, it integrates information using two different time scales. Thus, not only the most recent rewards are the most relevant information for the computation of Q values, but also information from far in the past could be relevant. The learning rates are free parameters, so they can be adapted to fit the requirements of a data set. The equation for this variation is the following:

$$Q^1_{t+1}(a) = Q^1_t(a) + \alpha_1 \left[r_t - Q^1_t(a)\right] \tag{2.15}$$

$$Q^2_{t+1}(a) = Q^2_t(a) + \alpha_2 \left[r_t - Q^2_t(a)\right] \tag{2.16}$$

$$Q^{TC}_{t+1}(a) = (1-w)Q^1_{t+1}(a) + wQ^2_{t+1}(a) \tag{2.17}$$

The goal of this chapter is to propose a different way of integrating information from previous rewards by encoding and retrieving them as episodes. We review a previously proposed model that treats past rewards as episodes and suggests a different mechanism for estimating the value of actions. This model was advocated by Bornstein et. al [Bornstein et al., 2017]. In this model, the values of actions are estimated by sampling the rewards from individual past events instead of using incrementally averaged rewards. The proposal is that agents sample a previous event and use the value of the reward received at that point to estimate the current value of actions. This model samples previous events based on temporal recency and based on evoked memories from the past. It is important to note that the contextual cue serves as a reminder, but it is not part of the model. Its influence is only accounted by the influence of the value of the reward on the reminded trial. In terms of temporal recency, it maintains, similarly to RW, that the most recent events should have the highest influence in current decisions. For the two actions scenario we have have been considering, Bornstein's sampling probability functions for the values of each actions are the following [Bornstein et al., 2017]. In these equations, B_s specifies the sampling function based on recency and B_e the sampling function based on evoked or reminded trials. α_s and α_e are the parameters of the model that specify the likelihood of a recent (direct) or a reminded (evoked) trial been sampled. The index i refers to previous time steps.:

$$P(Q^{B_s}_t(a) = r_{i,s}) = \alpha_s(1-\alpha_s)^{t-i} \tag{2.18}$$

$$P(Q_t^{B_e}(a) = r_{i,e}) = \alpha_e(1-\alpha_e)^{t-i} \tag{2.19}$$

It is important to note that the first probability function has the same form as the weighting function used to compute the incremental running averages of rewards from equation 2.14 [Bornstein et al., 2017]. For action selection, the sampled values are used to replace the estimated values of actions used in equations 2.1 and 2.2. To compute the probability of taking a particular action, the model takes the expectation over every possible individual sample. It uses the sampling probability based on recency for all but the evoked trial's sample, where it uses the sampling probability for evoked trials. Equation 2.20 (below) describes this probability function $P(a)$ (when only one sample is evoked - as it is the case in Bornstein's experiment). In this equation, the parameter β_c models any perseverative effect of the previous trial's choice, and the term I_t^c is an indicator function that returns 1 if the previous choice was identical to the current choice and 0 otherwise.

These equations describe Bornstein's model, which he calls the Sampling model. The primary difference between the Sampling model and Rescorla-Wagner is where the expectation or average of past rewards is taken. Rescorla-Wagner computes the average first and then uses this average to compute choice probabilities, while the sampling model takes the average after computing the choice probability for each sample value. The motivation for this is to make action choices based on one sampled episode, but make this sample be the expected episode to be sampled. Conceptually the difference between these two models is that Rescorla-Wagner accumulates the information from all samples and averages them, while the Sampling model uses the information from one episode. Mathematically, however, these two quantities are related. When more than one sample is used in the Sampling model, these two models start to converge. In the limit of $k- > \infty$, these two models compute the sample value. When just one sample is used, they give different predictions [Bornstein et al., 2017]

In particular and in terms of performance, a learning agent that uses Bornstein's Sampling model would be able to accumulate higher rewards than a learning agent that uses the RW model when the environment (rewards) have high variance. Bornstein tested that indeed the latter form is a better fit for trial by trial human choices in two different experiments [Bornstein et al., 2017]. The first experiment consisted of data from a previously published learning study [Daw et al., 2005]. 26 participants completed a four-choice bandit task for 300 trials. On each trial, participants had to choose one of four different slot machines to maximize their overall payoff. Each machine had assigned a payoff between 0 and 100 points. The payoff for each machine was initialized at a number within this range, and then this payoff evolved over time following a random walk. Participants learned the values of each machine, and use this to make their choices. Figure 2.1 describes the task and the time-varying rewards for each bandit [Daw et al., 2005].

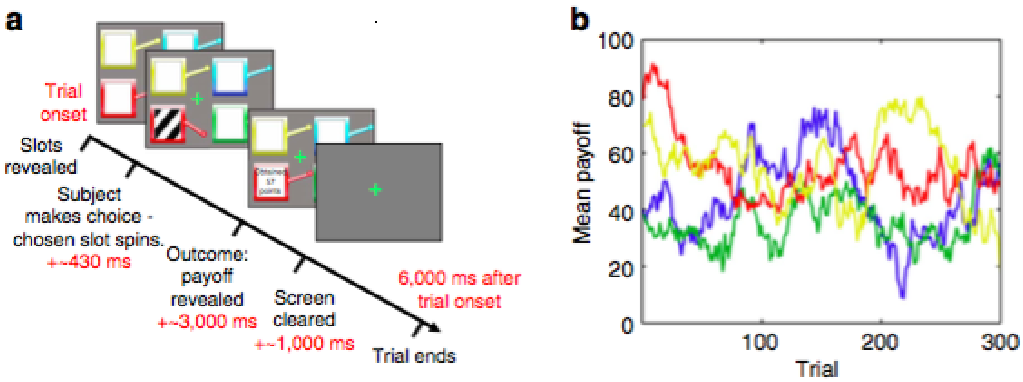

Figure 2.1: Four-arm bandit task from [Daw et al., 2005]. (a) Participants chose between four slot machines to receive points. (b) Payoffs. The mean amount of points paid out by each machine varied slowly over the course of the experiment

In the original paper by Daw, [Daw et al., 2005], a standard incremental learning rule was used to fit the data. Bornstein fitted the data using the proposed sampling model and a standard incremental learning rule. He computed the probability that each model would produced the observed choices (Likelihood of the data), and selected the parameters that maximized the likelihood of each model. Model selection was performed by comparing the likelihoods computed by each model. The Sampling model fit the data better with mean log Bayes factor (described in experiments section) of 8.867 and standard deviation of 1.081 against the incremental learning model [Bornstein et al., 2017]. In conclusion, the sampling model fits this data better even though the experiment was not designed with a structure that favors or encourages sampling.

In the previous experiment, each trial looked formally identical, and there was no clear indication that subjects might be sampling specific individual events from the past. For that reason, a second behavioural experiment was performed. The goal of this second experiment was to measure the impact of single and selected experiences on choices. 30 participants made a choice between two-armed bandits to maximize cumulative payoff during 162 trials. Each bandit had a probability of giving a winning (+5) ticket or a loosing (-5) ticket. The probability assigned to each bandit changed slowly over time. In order to differentiate across trials, each trial was tagged with a picture. In 32 of the 162 trials, a picture from a previous trial was presented. These trials were called memory probe trials. Figure 2.2 shows the structure of the task, and the varying probability of reward.

The memory probes were intended to remind the participants of previous specific trials and so to facilitate and mark the use of sampled trials from the past, if they were indeed used. It was hypothesized that the reminded picture would also remind the participant of the bandit chosen at that trial. Model selection results showed that indeed this hypothesis was correct, and the sampling model was still a better fit in this experiment than the incremental model with mean log Bayes factor of 6.918 and standard deviation of 1.32 favoring the Sampling model over the incremental model [Bornstein et al., 2017].

Bornstein's Sampling model proposes a new way to use past information to guide decisions. In particular, it uses episodic memory. However, the sampling model uses the information of just one episodic sample, sampled based on recency or the memory probe in the case of probed trials. It fails to integrate information from other trials that occur nearby in time. In this work, we argue that such the sampling method limits the flexibility and generalization of the usage of previous experience. Consequently, it limits the performance of learning agents. Recent events may or may not be optimal to guide current decisions. Often times, events far away in the past are key for making optimal choices in the present. We propose

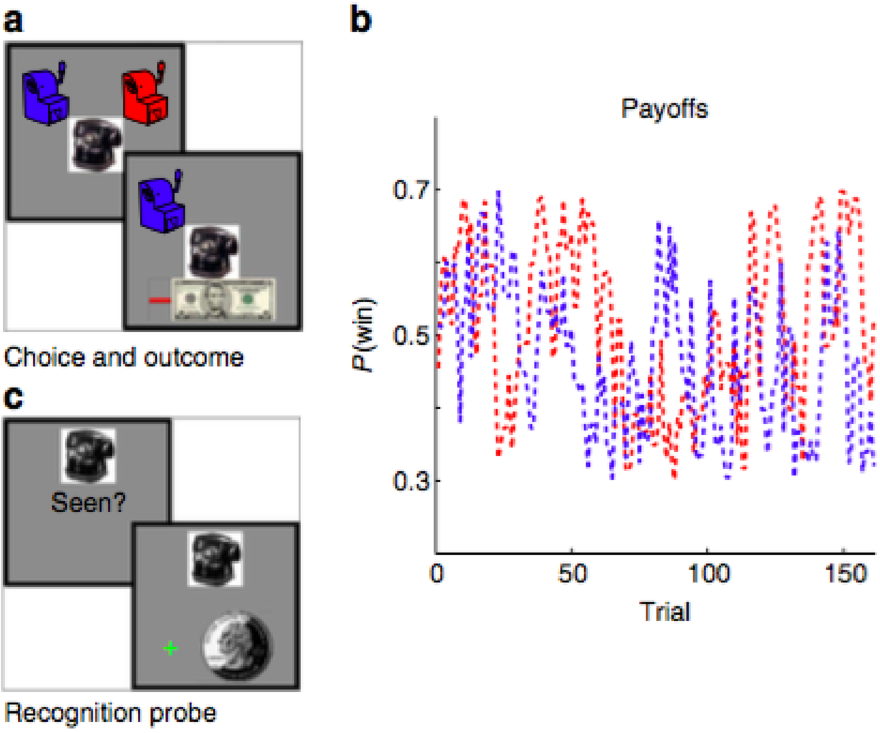

Figure 2.2: Two-arm bandit task from [Bornstein et al., 2017]. (a) Each bandit delivered tickets—trial-unique photographs—associated with a dollar value—either $5 or 5$. (b) Payoff probabilities. The probability of each bandit paying out a winning ticket varied slowly over the course of the experiment.(c) Memory probes. Participants encountered 32 recognition memory probes. On 26 of these probe trials, participants were shown objects that were either received on a previous choice trial ('valid'), whereas on others they were shown new objects that were not part of any previous trial ('invalid')

that the best way to access to this information is by recalling events that are similar in context - and therefore relevant to the current situation. For that, we base our model on a context based model of episodic memory. We describe this model in the following sub-section.

2.2.2 Temporal Context Model

Episodic memory refers to the memory of single events including features such as time, places, and associated context and emotions. It is the memory system that allows the recall of events that took place at a particular time and place. For this reason, episodic recall is often known as "traveling back in time" [Schacter and Addis, 2009]. During studies of free and serial recall, two basic principles of human episodic memory have been characterized: recency and contiguity [Mayo and Crockett, 1964]. Recency refers to the ubiquitous finding that memory performances decreases over time or with the presence of intervening items. Contiguity refers to the finding that presentation of an item facilitates the recall of another item that occurred close in time to this presented item. Both these principles relate to temporal factors of recall. For this reason, Howard and Kahanna proposed a computational model of episodic memory that captures these main two features [Howard and Kahana, 2002]. These two features, as well as the data that inspired this model, were observed in free recall experiments, where subjects are presented with a list of words and then asked to remember the words in any order. These principles imply that in free recall the last rehearsed items will be the first ones recalled, and that once an item is recalled, items that occurred right before or right after this item most likely to be recalled. Another important observed feature captured by this model is asymmetric recall. This feature refers to the observation that items that that occurred forward in time from the last item recalled are more likely to be recalled than items that occurred backwards in time [Howard and Kahana, 2002].

Howard and Kahanna's proposed model is called: Temporal Context Model (TCM) [Howard and Kahana, 2002]. This model adds a notion of context to episodic encoding and retrieval. In this model, context refers to everything else in an event that is not the item been attended. In free recall experiments, it refers to the environment, the person's mood, external sounds,and everything else that is happening during the experiments except for the studied words. Context evolves at slower time scales than items; in this way, allowing information to persist for longer. Context drifts through time, and it represents a running average of recent items. It binds to the transient representations of items, and consequently provides memory with a temporal structure. Figure 2.3 shows possible different contexts evolving at different time scales during a free recall experiment [Howard and Kahana, 2002].

TCM proposes that context can be used to store and later retrieve stored items. In free recall, this implies that context can cue the recall of specific words. For episodic memory, this means that events can be retrieved using contextual cues from

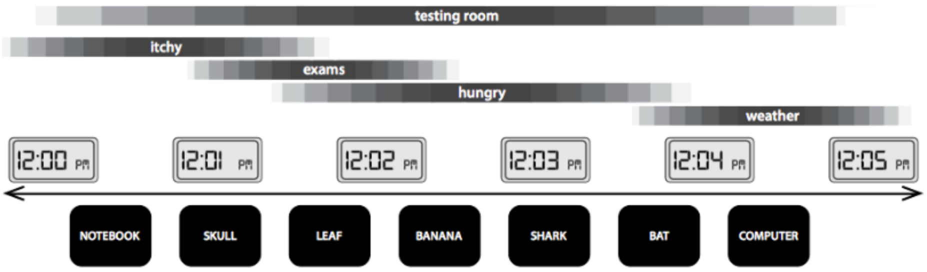

Figure 2.3: A schematic of TCM. An observed item remains activated in context for a period of time. Thus, context at each time step represents the current observation and decaying versions of previous observations. [Howard and Kahana, 2002]

the environment. In TCM, recall is stimulated by using the current context as a retrieval cue. To achieve this, each stored item is associated with the context that was active at the time of encoding. Stored items compete, and the item recalled is the one with the associated context that is the most similar to the current context. The item recalled and its associated mental context are incorporated in the current context. In this way, context evolves in time based on observed items and contexts recalled. It is this contextual retrieval that allows for the ability to recall events from far in the past and bring them to awareness in the present [Howard and Kahana, 2002].

Algorithmically, TCM has five basic components [Howard and Kahana, 2002]:

1. A space of items F, with elements $\mathbf{f}_t$ corresponding to item representations at time t. Each item is represented by a unique unit-length vector. All items are orthogonal to each other.

2. A context space C with entries $\mathbf{c_t}$ corresponding to the state of the context at time t.

3. A matrix M^{TC} that represents the strength of the connections between individual items in F and contextual states in C. This matrix enables context to act as a cue for the retrieval of items. It is constructed as a set of outer-product terms: $M^{TC} = \sum_{t=1} \mathbf{f}_t \mathbf{c}_t^{'}$, where $\mathbf{f}_t$ represents an item from F presented at time t. This matrix is used to represent items can be used as cues for context. In our work, we focus exclusively on how context can be used to retrieve items, so we do not use the matrix M^{TC}. We included it here for completion.

4. A matrix M_t^{CT} that represents the strength of the connections between con-

texts at each time step and individual items. This matrix enables an individual item to act as a cue for contextual retrieval. We describe the learning rule for this matrix below.

5. A contextual evolution equation that describes how context drifts over time, and how it is updated each time step using the context retrieved after the presentation of an item. This is the core of the temporal dynamics of TCM:

$$\mathbf{c}_t = \phi \mathbf{c}_{t-1} + \beta_b \mathbf{c}_t^{in} \tag{2.20}$$

Here, β_b is a free parameter, and it represents the contribution of $\mathbf{c}_t^{in}$ (retrieved context) to the evolution of context. ϕ represents how much information from the previous time step $\mathbf{c}_{t-1}$ is kept in the current time step. A large ϕ indicates that context is drifting slowly, and that the information from the recent past is more relevant than the new inputs. If ϕ is close to 0, then the context vector has no memory, and it changes each time step depending on the new inputs. ϕ can take values between 0 and 1, and it is chosen to satisfy the constraint $|\mathbf{c}_t| = 1$. An equation that satisfies this constraint is given by the equation below, where $\mathbf{c}_t^{in}$ represents the context retrieved by the presentation of item $\mathbf{f}_t$, and it is also unit norm. [Howard and Kahana, 2002]:

$$\phi_t = \sqrt{1 + \beta_b^2[(\mathbf{c_t}\mathbf{c}_t^{in})^2 - 1]} - \beta_b(\mathbf{c_t}\mathbf{c}_t^{in}) \tag{2.21}$$

When item $\mathbf{f}_t$ has not been presented or has not been presented recently in the sequence, we can write the equation for ϕ_t as:

$$\phi_t = \phi = \sqrt{1 - \beta_b^2} \tag{2.22}$$

Contextual retrieval is obtained by presenting the current item to the matrix M_t^{CT} [Howard and Kahana, 2002]:

$$\mathbf{c}_t^{in} = M_t^{CT} \mathbf{f}_t \tag{2.23}$$

The matrix M_t^{CT} is updated each time step using a rule that balances the contribution between the pre-experimental context (context associations before the presentations of items in a study) and newly learned associations. Thus, when an item is presented, the context that is retrieved is a combination of the pre-experimental context ($\mathbf{c}^{pre}$)) and the context when the item was presented during the study at time $t-r$ ($\mathbf{c}^{new}_{t-r}$)) . This enables TCM to account for the asymmetric retrieval characteristic observed in free and serial recall. The reason for this is that the pre-experimental context favors forward recall, which biases the probability of recall towards items in the forward time direction - as observed in human studies. Specifically, the rule for updating the matrix M^{CT}_t is the following [Howard and Kahana, 2002]:

$$M^{CT}_{t+1} = M^{CT}_t \tilde{\mathbf{P}}_{\mathbf{f}t} + A_t M^{CT}_t \mathbf{P}_{\mathbf{f}t} + B_t \mathbf{c}_t \mathbf{f}^{'}_t \tag{2.24}$$

The first two terms in this equation represent the strength of pre-experimental associations between items and context, and the last term represent a new Hebbian association between the context at time t and the item presented. A and B are chosen such that the norm of the input at time t is equal to one : $|\mathbf{c}^{in}_t| = 1$. For that, a variable γ is introduced such that: [Howard and Kahana, 2002]:

$$\gamma = \frac{A_t}{B_t} \tag{2.25}$$

and

$$A_t = \gamma B_t \tag{2.26}$$

Using these definitions and solving for the constraint on the norm of $\mathbf{c}^{in}_t$, we have that:

$$B_t = \frac{1}{\gamma^2 + 2\gamma(\mathbf{c}^{in}_t \mathbf{c}_t) + 1} \tag{2.27}$$

A_t and B_t balance the contribution of pre-experimental and newly learned context in the retrieved context vector such that:

$$\mathbf{c}^{in}_t = A_t \mathbf{c}^{new}_{t-r} + B_t \mathbf{c}^{pre} \tag{2.28}$$

$\mathbf{P_f}_t$ defines a projection operator with respect to $\mathbf{f}_t$:

$$\mathbf{P_f}_t \equiv \frac{\mathbf{f}_t \mathbf{f}_t^{'}}{||\mathbf{f}_t||^2} \tag{2.29}$$

and :

$$\tilde{\mathbf{P}}_\mathbf{f} = \mathbf{I} - \mathbf{P_f} \tag{2.30}$$

In conclusion, the step by step process of TCM starts when an item is presented at time t. First, $\mathbf{f}_t$ retrieves context $\mathbf{c}_t^{in}$. Then this retrieved context is used to evolve the context vector, resulting in a new context state $\mathbf{c}_t$. Then both association matrices are updated. This process describes contextual retrieval and contextual integration. TCM characterizes the process of contextual encoding, storage and retrieval of episodic memory and it is that we use in this chapter for our proposal [Howard and Kahana, 2002].

2.3 Proposal

In this section, we describe the Contextual Episodic learning framework (abbreviated later on as CE), and the Hybrid model. The Contextual Episodic learning framework has similarities and differences with previous models. For example, similar to Bornstein's sampling model, it recalls episodic events to compute the long run value of each action. However, it differs in the way this information is integrated. Bornstein proposes that learning agents sample one - or more episodes (although his work only shows results for one) - based on recency or evoked trials. Then, it uses the value of this sample to compute the probability of choosing each action. The final step, as was described in the previous section, is to take the expected value of this probability over all possible events that could have been sampled from the reward history [Bornstein et al., 2017]. Our work differs from this model in the fact that it does not use the expected value of one sample, but the value of multiple samples to compute a weighted average [Bornstein and Norman, 2017]. In this way, our model differs from Borsntein's in a similar way that Rescorla Wagner differs from Bornstein's sampling model. The reason for this is that we know from previous work [Bornstein and Norman, 2017] [Duncan and Shohamy, 2016] that context cues the retrieval of more than one event. We simply propose a model that explains how this retrieval happens based on context, and then uses this context to weight the relevance of each episode recalled for the current decision. In a way, our model is similar to model-free cached values. It is a weighted average of previous rewards. However, it is different from this framework, because model-free cached values are computed with a recency-based average (decaying exponential weighting function), while we use a contextual similarity- based average. In this section, we describe the equations of the new proposals, and discuss in detail these similarities and differences.

2.3.1 Equations Summary

Similarly to the other two reinforcement learning frameworks, in our Contextual Episodic framework, the goal of the agent is to maximize future expected rewards. For that, a learning agent computes a value of the utility of each action based on previous rewards or punishments. We call this utility value $Q_t^{CE}(a)$, because it is computed as a weighted sum of rewards received in previous episodes. To calculate $Q_t^{CE}(a)$, we use the TCM model to recall past episodes. However, contrary to the TCM model, we don't assume that only one item is retrieved given a contextual cue, but rather that different items are retrieved with a weighted contribution. We weight how much each episode influences the current decision, based on how similar its associated context is to the current context. Computationally, we implement this using the Nadayara Watson Kernel Regression, which estimates future expected value as a locally weighted average. This regression uses a kernel as a weighting (or similarity measure) function. $Q_t^{CE}(a)$ is computed using this regression function in the following way:

$$Q_t^{CE}(a) = \frac{\sum_j K(\mathbf{c}_{t-j}, \mathbf{c}_t) r_{t-j}(a)}{\sum_k K(\mathbf{c}_{t-k}, \mathbf{c}_t)} \tag{2.31}$$

where:

$$K(x,y) = e^{-||x-y||} \tag{2.32}$$

In this equation, $\mathbf{c}_t$ represents the current context, and $\mathbf{c}_{t-j}$ represents context at previous times j. r_{t-j} refers to the reward received at previous time step j. $\mathbf{c}_t$ is computed by the context evolution equation from the TCM model [Howard and Kahana, 2002] described in the section above. We describe episodes as the rewards received, and these episodes are associated with a particular context at each time step.

As we have stated, our hypothesis is that organisms can use a combination of learning frameworks. For this reason, we propose the following Hybrid model:

$$Q_t^{hybrid}(a) = (1-w)Q_t^{CE}(a) + wQ_t^{RW}(a) \tag{2.33}$$

In this model, $Q_t^{CE}(a)$ is the Q value computed using the Contextual Episodic model, $Q_t^{RW}(a)$ is the current model free running average computed using the Rescorla-Wagner incremental learning rule, and w measures the trade-off or weighted contribution between these values. Once the Q values have been estimated by either the Contextual Episodic or the Hybrid model, we can compute the probability of each action using a soft-max distribution in terms of the Q values of all actions. In the following equation, we use Q values computed by the Hybrid model as an example:

$$P(a) = \frac{e^{\beta_c I_t^c + \beta Q_t^{hybrid}(a)}}{e^{\beta_c I_t^c + \beta Q_t^{hybrid}(a)} + e^{\beta_c I_t^c + \beta Q_t^{hybrid}(b)}} \tag{2.34}$$

2.3.2 Discussion

As can be seen, this new proposal has elements of similarities and differences with other models. Here we discuss the implications of each of them.

2.3.2.1 Contextual Episodic and Model-free RL

Both these models compute a weighted average of past rewards. The difference between these two models is the form of the weighting function. While model-free RL algorithms such as the Rescorla-Wagner rule, compute the average of past rewards using a decaying exponential, Contextual Episodic model uses a contextual similarity based function. A decaying exponential weighting function implies that more recent events are more relevant to current decisions; however, this is often not the case, when events far in the past repeat themselves in the present. In those cases, a recency-based approach fails to capture the relevant information. A contextual similarity weighting function makes sure that relevant information has more weight in current decisions regardless of when they occurred in the past.

What is important for this model is the similarity between contexts. Intuitively, it makes sense that we would want to use information collected from similar situations to the present one. For example, if we want to evaluate whether to go to a pizzeria in London, we would not want to use the information from our recent pizzeria experiences in Italy, but rather, remember our experiences in a similar context i.e. last time we visited a pizzeria in London.

Furthermore, the Contextual Episodic model uses a temporal context model (TCM) to compute the contextual similarity weighting function. The TCM model allows us to not only compare the contexts between specific events, but also the events that occurred nearby in time to these events. The reason for this is that context that evolves according to TCM has a slow time constant of evolution. Thus, it preserves information from nearby contexts. Consequently, all the events that occurred nearby in time to the cued context have a higher weight of recall. In an environment in which the state of the world is moving slowly, recalling events nearby in time to the cued episode would be useful. This characteristic resembles episodic memory recall in humans, and allows us to ”travel back in time”. The motivation for our proposal is that we hypothesize that humans assume the world has these characteristics, and use this property of episodic memory when observing a cue. In this chapter, we focus on showing how these characteristics of episodic memory fit data on human choices better. In this way, showing that the retrieval mechanism just described is necessary to understand human choices in certain situations. The precise characteristics of the world and situations for which an episodic based RL algorithm would be useful is the topic of next chapter

Computationally, at time t, these two models compute averages of rewards using the following weighting functions for past rewards: r_{t-j}. Equation 2.36 describes the weighting function of the Rescorla Wagner model, and equation 2.37 describes the weighting function of the Contextual Episodic model.

$$w_{RW} = \alpha \cdot (1-\alpha)^{t-j} \tag{2.35}$$

$$w_{CE} = \frac{K(\mathbf{c}_j, \mathbf{c}_{t-j})}{\sum_k K(\mathbf{c}_{t-k}, \mathbf{c}_t)} \tag{2.36}$$

2.3.2.2 Contextual Episodic model and Sampling model

Both Contextual Episodic and the Sampling models use episodes in the form of previous rewards to compute the Q values of actions. The sampling model uses a recency-based sampling probability, and a probability based on evoked trials (or trials where a reminder is presented). At the moment of making a decision, a learning agent takes the expected value of choosing one sample over all reward history, including evoked trials, weighted by their associated sampling probabilities. For most trials, this sampling probability favors recent events, and makes it unlikely that samples from the past are selected. Evoked trials are also sampled by recency, but they have an independent parameter (that can be fitted to data) that might make them more likely to be recalled even if they happened far in the past. It has similar drawbacks as the Rescorla Wagner model, recency plays a big role in what events get sample. Evoked trials have a higher chance of been recalled than other trials. However, there is no principled way for recalling evoked trials, and recency is not a very good measure of utility of information for the decision making agent. In this way, our Contextual Episodic model is different. We propose a context based approach to recall past events, and have a measure of relevance of these events for the current decision also based on this context.

2.3.2.3 Hybrid model and Sampling model

An interesting comparison arises when we compare the Sampling model to our Hybrid model (between RW and Contextual Episodic). Bornstein et al. also explored hybrid version between RW and his Sampling model [Bornstein et al., 2017], but concluded that the Sampling model was better than the Hybrid model, but the reasons for this were unclear [private correspondence, May, 2019]. One hypothesis for this is that the Sampling model already captures the relevant information from previous trials when it takes the expected value over all previous rewards weighted by their recency. For this reason, the Rescorla Wagner model did not offer any extra useful information to Bornstein's Hybrid model. However, the Sampling model performs better because it weights more the contribution of evoked trials, which is exactly what our model proposes. But, the Contextual Episodic takes this two steps further. First it allows the possibility of one or many episodes been recalled, and the number of episodes recalled and how much they are weighted is based on their contextual similarity. In this way, our Contextual Episodic model provides a way of measuring how relevant an episode is to the current decision. The Hybrid model combines these advantages of the Contextual Episodic model with information provided by the RW model. This explanation is obvious for experiment 2, where specific trials are reminded. For experiment 1, this can be explained by humans finding (or trying to find) a structure in a sequence of items. In order to learn and make inferences and predictions, humans must make certain assumptions about the structure of the environment. The fact that we show that the Contextual Episodic model fits human data better than other models even for experiment 1 shows that humans make certain assumptions about the temporal structure of events. In the next chapter, we propose a generative model for which the Contextual Episodic model would be a good inference approximation with the objective to unravel some of these assumptions. We don't know what the generative model that humans assume, but we aim to begin an exploration in this direction.

An interesting comparison arises when we compare the Sampling model to our Hybrid model (between RW and Contextual Episodic). Bornstein et al. also ex-

plored hybrid version between RW and his Sampling model [Bornstein et al., 2017], but concluded that the Sampling model was better than the Hybrid model, but the reasons for this were unclear [private correspondence, May, 2019]. One hypothesis for this is that the Sampling model already captures the relevant information from previous trials when it takes the expected value over all previous rewards weighted by their recency. For this reason, the Rescorla Wagner model did not offer any extra useful information to Bornstein's Hybrid model. However, the Sampling model performs better because it weights more the contribution of evoked trials, which is exactly what our model proposes. But, the Contextual Episodic takes this two steps further. First it allows the possibility of one or many episodes been recalled, and the number of episodes recalled and how much they are weighted is based on their contextual similarity. In this way, our Contextual Episodic model provides a way of measuring how relevant an episode is to the current decision. The Hybrid model combines these advantages of the Contextual Episodic model with information provided by the RW model.

2.3.2.4 TCM applications in free recall and in Contextual Episodic decision making

The Contextual Episodic model uses the Temporal Context Model to define context evolution, and storage. They differ in how retrieved episodes are used. In both models episodes are retrieved using contextual cues. However, in TCM, retrieved episodes compete and then only one episode is selected. The reason for this is that in free recall, subjects need to name one item at a time. In our Contextual Episodic model, a learning agent does not need to select just one episode, but rather use as much information as possible to make good decisions. For that reason, retrieved episodes are integrated and then the integrated information is used to select an action. In short, the main difference is that in free recall, the output is a specific recalled event, while in decision making, the output is an action, for which the information from many recalled events is used.

2.4 Model Implementation

In this section, we explain how we implemented the Contextual Episodic and Hybrid models to fit the data from Experiment 1 [Daw et al., 2005] and Experiment 2 [Bornstein et al., 2017] (description in literature review).

2.4.1 Contextual Episodic Model

First, we define the items that modify context as the time steps of each trial for Experiment 1 (since it does not have a contextual cue), and as the pictures shown at each trial in Experiment 2. The context vector is a unit length vector of size equal to the number of time steps. The context evolves each time step using the picture items as inputs. The pictures are random, so each of them is a unit length vector of size n with only one entry equal to one. Each picture is associated with the context at the time of item presentation,and stored in the matrix M^{TC}. This matrix is initialized as a diagonal matrix, and it evolves with the TCM equations described in equations 3.5,3.6 and 3.7 [Howard and Kahana, 2002].

In experiment 2, after a picture is presented, the current context is updated using the context evolution equation from TCM [Howard and Kahana, 2002]. Then this context is used to retrieve previous episodes. These episodes are retrieved by comparing the current context to the context when these episodes happened. The goal with this mechanism is to use the picture presented as a contextual cue that modifies current context, and then to mimic previously observed contextual retrieval [Duncan and Shohamy, 2016] by searching in memory for events that had similar context to the current one. If the picture presented is new, then current context will only be similar to the most recent context - thanks to the slow drift term of the context evolution. However, if the picture presented is the same picture or similar to a picture that was shown in the past, then the current context will be similar to other previous contexts. In this situation, one or more episodes are retrieved and averaged weighted by how similar their associated contexts are to the current context. Using this weighted average, Q values are updated and an action is chosen. With this action, a reward is received, and this episode with the associated context is stored in memory.

2.5 Results

In this section, we test the Hybrid model hypothesis presented in this chapter. For that, we generate artificial data from the Hybrid model, and compute the likelihood of this data been generated by the Hybrid model, the Rescorla-Wagner model, the Sampling model, and the Time Constants model [Rescorla et al., 1972]. We use the Time Constants model, because it has two different time scales, and can have recency-based weighted averaged combined with an average that weights events from further in the past - both these time constants can be learned to fit the data. We use this model to test whether learning two different time constants per data set could replace the contextual similarity weighting. We test whether the Hybrid model can be learned best by the Hybrid model, and therefore confirm that it is a model independent from previously proposed models. Afterwards, we generate data from these models, and do model fitting to compute a confusion matrix. We show the results of this confusion matrix, where the results of the performance of all the models using data generated by the Hybrid model can also be seen.

We perform data fitting for each model and data set using maximum likelihood estimation. We compare these models using BIC scores. BIC is a method for comparing likelihoods penalized by the number of parameters in each model [Schwarz et al., 1978]. The BIC score is:

$$\mathrm{BIC} = -2\log(\mathrm{Likelihood}) + Klog(N) \tag{2.37}$$

where K refers to the number of free parameters, and N to the number of data points. In the confusion matrix, we compare how well each model fits the data from the other models, by reporting BIC scores for each data set and each model. Our goal is to show that indeed these models are different, and can be distinguished with model comparison. In the next section, we perform a similar analysis using human data and claim that we indeed are able to distinguish the strategy used by humans among the different proposed models.

We chose the BIC over AIC, because we wanted to compare our results to that of [Bornstein et al., 2017]. In his work, Bornstein use BIC to compare between models. For this reason, in our work we use the BIC to first replicate his results, and second to compare the relevance of our results. The improvements in performance between our models and his measured through the BIC that we found and report in this thesis are of similar magnitude to the improvements reported in his paper between his model and previous ones.

2.5.1 Analysis using Artificial Data

The models used in this section are the Rescorla-Wagner model, the Time Constants model, the Contextual Episodic Model, the Hybrid model and the Bornstein's Sampling model.

2.5.1.1 Experiments Description

We generate artificial data from the models described above. The data that we generate is similar to Experiment 1 [Daw et al., 2005] and to Experiment 2 [Bornstein et al., 2017] described earlier. For experiment 1, we generate data for 300 trials using four bandits with payoffs between 0 and 100. These payoffs evolve following a Gaussian random walk. For experiment 2, we generate data for 162 trials using two bandits with payoffs equal to +5 or -5. Each payoff is assigned to a probability that slowly drifts over time. Experiment 1 generates choice data when no memory probes are given to the learning agent, while experiment 2 uses the same memory probes (items and times) as in Bornstein's experiments. In this way, these two experiments explore the same two cases that Bornstein did when he described and analyzed the sampling model. The parameters that we use to generate these data sets are the same, when possible, to the ones estimated by Bornstein in his paper (i.e. the learning rate α,the learning rate for the Sampling model α_s and the temperature parameter β) [Bornstein et al., 2017]. For the second learning rate of the Time Constants model, we picked a value that was about half of the first learning rate. For both the Time Constants model and the Hybrid model, we chose a value of $w = 0.5$, so the data has influence from both the learning rates and from both models in the case of the Time Constants and Hybrid models respectively. The parameter of the TCM update in the Hybrid model was chosen to be $\gamma = 0.7$. This parameter determines how fast the context updates its value given new inputs. The value we have chosen allows the context to be updated, but it also maintains memory from previous contexts. The Sampling model and the Hybrid model are the only models for which the contextual cues are incorporated in the model. Both the Time Constants and RW models are not influenced by contextual cues. For this reason, these two models do not benefit from the addition of memory cues from Experiment 2.

2.5.1.2 Maximum Likelihood Estimates and Confusion Matrices

We do model fitting using maximum likelihood estimation, and we compare each model using the BIC described earlier. Here, we present the results for the data sets generated based on the two experiments we described. First we show the estimated parameters for each model fitted to data generated from the Hybrid model and from itself.

We start with the Rescorla-Wagner model, and we fit data generated by the RW and the Hybrid models using the RW model. We do this for both experiment 1 and experiment 2. Table 2.1 and Table 2.2 show the actual parameters used to generate the data (using the Rescorla Wagner model and the Hybrid model respectively), and the estimated parameters by the RW model.

	Actual Parameters	**Estimated Parameters**	**Actual - Mean Estimates**
Exp. 1	$\alpha = 0.77$ $\beta = 8.7$ $\beta_c = 0.65$	$\alpha = 0.75$ (0.05) $\beta = 8.51$ (0.31) $\beta_c = 0.59$ (0.15)	0.02 0.19 0.06
Exp. 2	$\alpha = 0.55$ $\beta = 1.74$ $\beta_c = 0.12$	$\alpha = 0.66$ (0.07) $\beta = 1.55$ (0.32) $\beta_c = 0.15$ (0.05)	0.11 0.19 0.03

Table 2.1: Model Fitting Results. In this table, we show the RW model parameters used to generate the data (left), the parameters learned through model fitting using the RW model (middle), and the difference between the actual and the means of the estimated parameters (right). In parenthesis next to the estimated parameters (middle), we show the standard deviation in the parameters' estimates.

	Actual Parameters	**Estimated Parameters**
Exp. 1	$\alpha = 0.77$ $\beta = 8.7$ $\beta_c = 0.65$ $w = 0.4$ $\beta_b = 0.52$	$\alpha = 0.8(0.15)$ $\beta = 8.2(0.6)$ $\beta_c = 1.05(0.56)$
Exp. 2	$\alpha = 0.55$ $\beta = 1.75$ $\beta_c = 0.12$ $w = 0.4$ $\beta_b = 0.52$	$\alpha = 0.63(0.08)$ $\beta = 1.59(0.22)$ $\beta_c = 0.84(0.04)$

Table 2.2: Model Fitting Results. In this table, we show the RW model parameters used to generate the data (left), and the parameters learned through model fitting using the Hybrid model (right). In parenthesis next to the estimated parameters, we show the standard deviation in the parameters' estimates.

Tables 2.3 and 2.4 show the results of model fitting using the Time Constants model. These results also show that this model is better at learning its own data. Furthermore, it shows that for experiment 2 - where contextual cues are given, this model fails to capture the temporal dependencies found in the data. The two time constants it learns are very close to each other. This shows that the model is not using its capacity to integrate information with two temporal dependencies. The reason for this is that the Hybrid model does not integrate information with two rigid time constants, but it is rather flexible and it adapts to external temporal cues and reminders. The Time Constant model probably found that it is much better to fit the data with one learning rate than to use two learning rates that are rather a misleading source of information i.e., for one time step information from the far past might be useful, but for the next time step it might actually be detrimental.

	Actual Parameters	**Estimated Parameters**	**Actual - Mean Estimates**
Exp. 1	$\alpha_1 = 0.77$	$\alpha_1 = 0.75$ (0.03)	0.02
	$\alpha_2 = 0.30$	$\alpha_2 = 0.28(0.14)$	0.02
	$\beta = 8.7$	$\beta = 8.2$ (0.62)	0.5
	$\beta_c = 0.65$	$\beta_c = 0.71(0.21)$	0.6
	w = 0.5	w = 0.65 (0.26)	0.15
Exp. 2	$\alpha_1 = 0.55$	$\alpha_1 = 0.44$ (0.08)	0.11
	$\alpha_2 = 0.30$	$\alpha_2 = 0.39(0.23)$	0.09
	$\beta = 1.75$	$\beta = 1.55$ (0.32)	0.2
	$\beta_c = 0.12$	$\beta_c = 0.08(0.10)$	0.04
	w = 0.5	w = 0.52 (0.98)	0.02

Table 2.3: Model Fitting Results. In this table, we show the Time Constants model parameters used to generate the data (left), the parameters learned through model fitting using the Time Constants model (middle), and the difference between the actual and the means of the estimated parameters (right). In parenthesis next to the estimated parameters (middle), we show the standard deviation in the parameters' estimates.

	Actual Parameters	**Estimated Parameters**
Exp. 1	$\alpha = 0.77$	$\alpha_1 = 0.91(0.36)$
	$\beta = 8.7$	$\alpha_2 = 0.60(0.21)$
	$\beta_c = 0.65$	$\beta_c = 1.25(0.85)$
	$w = 0.4$	$\beta = 8.21(0.92))$
	$\beta_b = 0.52$	$w = 0.8(0.06))$
Exp. 2	$\alpha = 0.55$	$\alpha_1 = 0.62(0.35)$
	$\beta = 1.75$	$\alpha_2 = 0.59(0.21)$
	$\beta_c = 0.12$	$\beta_c = 0.15(0.17)$
	$w = 0.4$	$\beta = 1.75(0.14)$
	$\beta_b = 0.52$	$w = 0.56(0.05))$

Table 2.4: Model Fitting Results. In this table, we show the Time Constants model parameters used to generate the data (left), and the parameters learned through model fitting using the Hybrid model (right). In parenthesis next to the estimated parameters, we show the standard deviation in the parameters' estimates.

We repeat the same analysis using Bornstein's Sampling model, and show the results in Tables 2.5 and 2.6. Similar to the other models, it fits its own data better than it fits the Hybrid model data. The most interesting parameter of this model is the α_e, which defines the influence of evoked trials (memory probes) on choices. For experiment 1, this parameter is set to zero [Bornstein et al., 2017], and it is only used for experiment 2. When fitting the Hybrid data, this parameter is higher than the learning rate α, which defines the influence of not evoked previous trials. This demonstrates that indeed the sampling model learns that memory cues are relevant; however, it fails to fit Hybrid data well since Hybrid choices are driven by a gradually evolving temporal context and not just by memory cues at particular time steps. Later, we will see how this affects probability of choice at one and two time steps after the memory probe.

	Actual Parameters	**Estimated Parameters**	**Actual - Mean Estimates**
Exp. 1	$\alpha = 0.72$ $\beta = 9.01$ $\beta_c = 0.65$	$\alpha = 0.65$ (0.16) $\beta = 9.52$ (0.62) $\beta_c = 0.72$ (0.21)	0.07 0.51 0.07
Exp. 2	$\alpha = 0.53$ $\alpha_e = 0.438$ $\beta = 2.28$ $\beta_c = 0.56$	$\alpha = 0.61$ (0.18) $\alpha_e = 0.46$ (0.35) $\beta = 2.09$ (0.86) $\beta_c = 0.48$ (0.06)	0.11 0.09 0.2 0.04 0.02

Table 2.5: Model Fitting Results. In this table, we show the Sampling model parameters used to generate the data (left), the parameters learned through model fitting using the Sampling model (middle), and the difference between the actual and the means of the estimated parameters (right). In parenthesis next to the estimated parameters (middle), we show the standard deviation in the parameters' estimates.

	Actual Parameters	**Estimated Parameters**
Exp. 1	$\alpha = 0.77$ $\beta = 8.7$ $\beta_c = 0.65$ $w = 0.4$ $\beta_b = 0.52$	$\alpha = 0.62(0.25)$ $\beta = 9.2(0.08)$ $\beta_c = 0.52(0.92)$
Exp. 2	$\alpha = 0.55$ $\beta = 1.75$ $\beta_c = 0.12$ $w = 0.4$ $\beta_b = 0.52$	$\alpha = 0.52(0.65)$ $\alpha_e = 0.68(0.12)$ $\beta_c = 0.75(0.06)$ $\beta = 2.5(0.06)$

Table 2.6: Model Fitting Results. In this table, we show the Sampling model parameters used to generate the data (left), and the parameters learned through model fitting using the Hybrid model (right). In parenthesis next to the estimated parameters, we show the standard deviation in the parameters' estimates.

Finally, we show how the Hybrid model learns its own data in Table 2.7. This results demonstrates that this model is distinguishable. We show in the next section with our analysis of human data that humans choices more closely resemble the Hybrid model than the other models.

	Actual Parameters	**Estimated Parameters**	**Actual - Mean Estimates**
Exp. 1	$\alpha = 0.77$ $\beta = 8.7$ $\beta_c = 0.65$ w = 0.4 $\beta_b = 0.52$	$\alpha = 0.79$ $\beta = 8.5$ (0.65) $\beta_c = 0.13$ (0.38) w = 0.56 (0.20) $\beta_b = 0.52$ (0.06)	0.02 0.2 0.52 0.16 0
Exp. 2	$\alpha = 0.55$ $\beta = 1.75$ $\beta_c = 0.12$ w = 0.4 $\beta_b = 0.52$	$\alpha = 0.50$ (0.19) $\beta = 1.65$ (0.72) $\beta_c = 0.09$ (0.24) w = 0.39 (0.04) $\beta_b = 0.52$ (0.15)	0.05 0.1 0.3 0.01 0

Table 2.7: Model Fitting Results. In this table, we show the Hybrid model parameters used to generate the data (left), the parameters learned through model fitting using the Hybrid model (middle), and the difference between the actual and the means of the estimated parameters (right). In parenthesis next to the estimated parameters (middle), we show the standard deviation in the parameters' estimates.

We compute the BIC scores for each of these models using the parameters estimated and data sets generated by each of these models, and generate a Confusion Matrix for both experiment 1 and experiment 2. These matrices shows how well each model learns data from other models compared to how it learns its own data. We normalize these matrices by subtracting the diagonal elements, which correspond to the BICs for each model using its own data. In this way, we can see the difference in performance on each column. The lowest value corresponds to the model's performance using its own data (diagonal entries of the confusion matrix), and the other entries show how much this performance decreases when using other model's data. The plot below corresponds to this confusion matrix. As can be seen, in Figure 2.4 the Hybrid model fits its data better, and the other models are not as good as fitting data generated by this model. The Hybrid model is a more complex model, not only in terms of parameter number, but also its dynamics are difficult to capture with other simpler models.

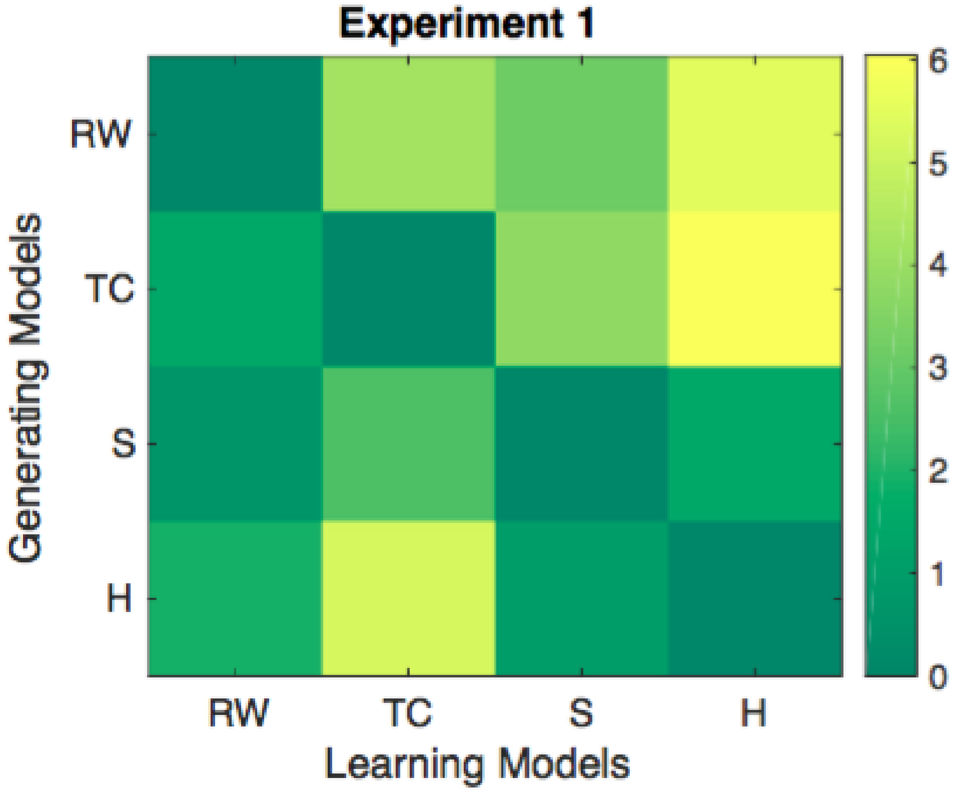

Figure 2.4: Confusion Matrix Experiment 1: The confusion matrix was normalized by subtracting the diagonal entries from each row. Diagonal entries correspond to each model's performance using its own data, and are equal to zero after normalization. The values of the Confusion Matrix are detailed in Appendix A.

The confusion matrices in Figure 2.4 and Figure 2.5 are based on data from the artificial data sets corresponding to experiment 1. Dark green represents lower values, while bright yellow represents higher values. We want to minimize the negative log-likelihood of the data; therefore, we say that a better data fit corresponds to lower BIC value (darker green).As can be seen, there is a difference in how much each model can learn with different data sets. All models struggle to learn data from the Hybrid model in both experiments.The Hybrid model learns data from the Hybrid model best: the Hybrid model has the lowest BIC when learning data generated from itself. This shows that if any data set would be generated by this model, we would be able to identify it correctly- this result is true for all other existing models as well. It is interesting to see that for all data sets, but particularly for the Hybrid

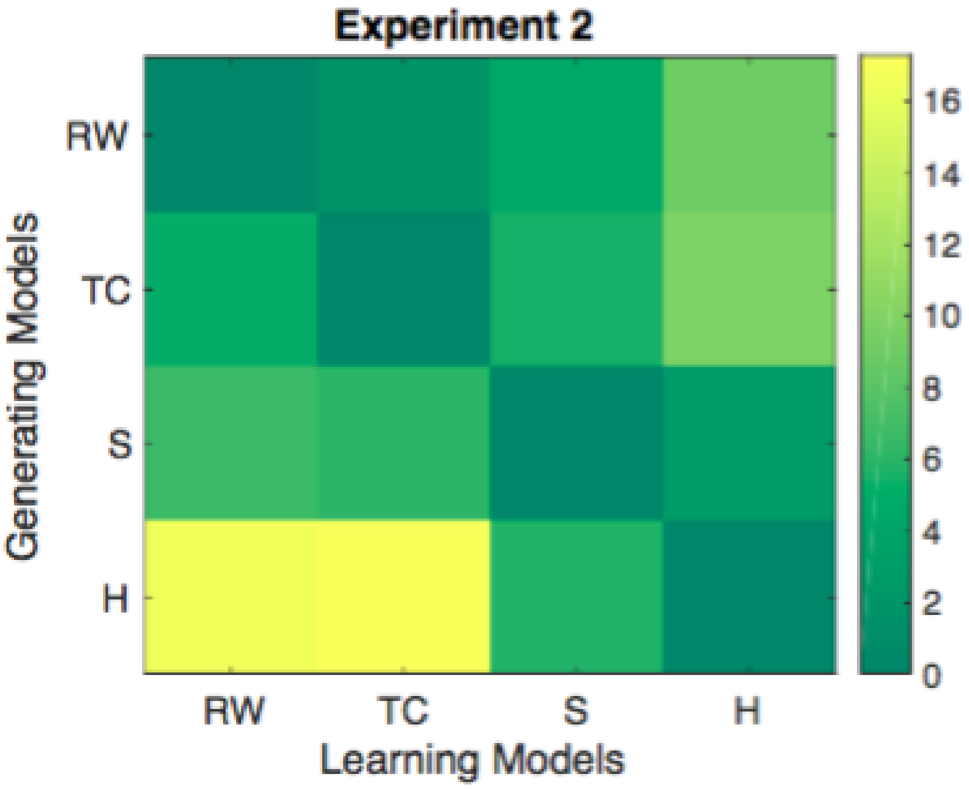

Figure 2.5: Confusion Matrix Experiment 2: The confusion matrix was normalized by subtracting the diagonal entries from each row. Diagonal entries correspond to each model's performance using its own data, and are equal to zero after normalization. The values of the Confusion Matrix are detailed in Appendix A.

data, all learning models perform better in experiment 1. There are differences in performance but they are small compared to the differences in performance in experiment 2. This result isn't surprising to us, since the memory probes are the main element that drives the difference between these models - the ability to be reminded of a particular trial. However, it is encouraging to also notice that despite the lack of memory probes, the Hybrid model still has unique characteristics that can not be captured by other models. The confusion matrix from experiment 2 shows that the Hybrid model is more adaptable at learning from data from other models. The differences in performance show that the worst model at capturing the data from the Hybrid model is the Time Constants model. Particularly, for experiment 2, it does not use the fact that it has two different learning rates, but it is penalized for having this extra parameter. Both the Time Constants model and the Rescorla Wagner model do not learn the Hybrid data well, because they fail to capture one of the main components that drive the data from the Hybrid model - memory probes. In experiment 1 where there are no memory probes, these models perform slightly better, but still the Sampling model out-performs them (a result that corroborates Bornstein's findings). The Sampling model, in both experiment 1 and 2, fits Hybrid data better; the reason for this is that it is the only model that has the capability to use information about memory probes.

The Time Constants model fits data worse than RW for all data sets; however, this difference is very small. So, we attribute it to the fact that the RW is penalized by fewer parameters, and the fact that the two learned learning rates for the Time Constants model are almost the same. The latter fact further shows that an extra time constant is not sufficient or necessary to capture data generated by a Hybrid model. Finally, we hypothesize that a better experiment - one designed to capture the temporal components of context evolution, would show the unique properties of the Hybrid model more.

2.5.1.3 Trial by trial analysis

In this subsection, we perform a detailed analysis of how contextual cues affect decisions; particularly, decisions at the time of the cue and two time steps later. To do this, we simulate what amounts to a third experiment which does not resemble either experiment described before. Its objective is to have a simpler data set to perform a more detailed analysis. In this experiment, rewards are either +10 or -10 for two bandits. The probabilities of receiving a reward drift over time. We generate 100 data points with 10 memory probes from the past. These memory probes are contextual cues from previous time steps. The memory probe times and the reminded trials at each probed time are shown below:

To analyze the effect of memory probes, we compute the average conditional probability of choosing the same action taken at the time of the probes given the sign of the reward received at that time. We label the current time as t and the time when the memory probed appeared the first time as t_m If the reminded trials affect decisions, we expect to see a bias towards the action taken at that time, when a

Probed Times	**Reminded Trials**
10	2
18	7
22	5
30	28
42	12
55	28
60	3
72	65
80	30

Table 2.8: In this table, we show the description of a third experiment, which does not resemble either experiment described before, consists of 100 data points with 10 memory probes from the past. This table shows the times of those 10 probes (right, Probed Times), and the trials that are cued at those times (right, Reminded Trials).

positive reward was received, and a bias towards taking the opposite action, when a negative reward was received. The mechanism for this is that the Q values computed will be updated using the value of the reward at the time of the probe, and this will increase or decrease the value of the action taken at that time depending on the sign for the reward received. We compare the average conditional probability described above for the Rescorla-Wagner, the Time Constants, the Sampling and the Hybrid models. We show that indeed the probes influence action selection in the Hybrid model and the Sampling model more than the other two models, which do not incorporate information from the memory probe.

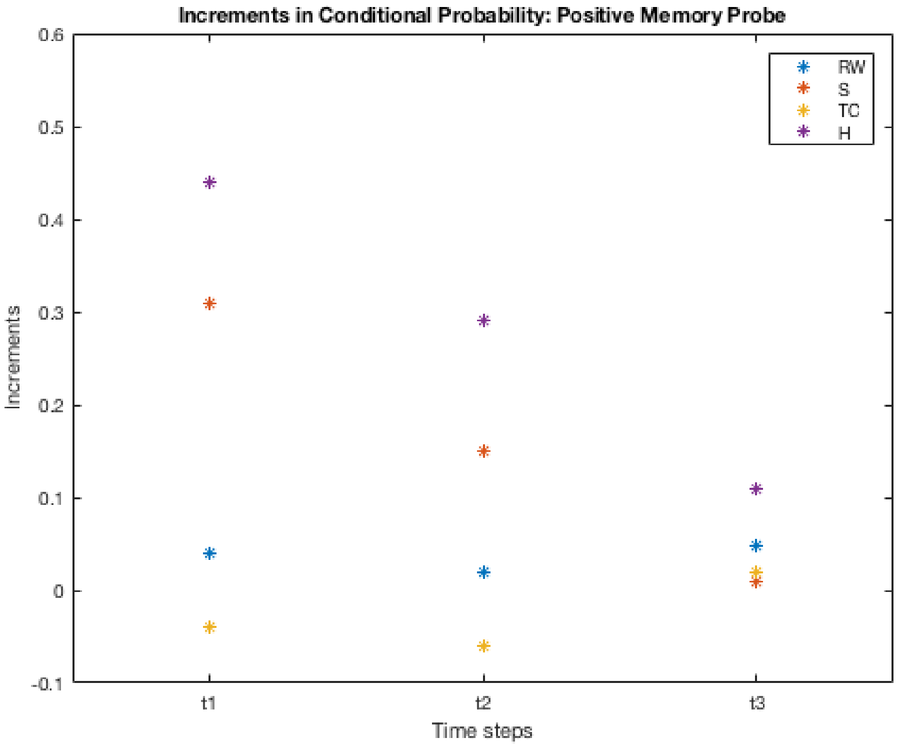

Figure 2.6: Increments (relative to chance) in Conditional Probability of repeating the same action at times t, t+1 and t + 2 after seeing a memory probe from time t_m at time t given that $r(t_m)$ was positive. (a). Comparison between all models at time t. There was a statistically significant difference between groups as determined by one-way ANOVA (F(3,396) = 16.30, p <0.001). A Tukey post hoc test revealed that the difference in conditional probabilities between the Hybrid model (0.94 +/- 0.021) and the RW model (0.54+/- 0.06, p <0.001), the Hybrid model and the TC model (0.46 +/- 0.056, p <0.001), and the Hybrid model and the Sampling model (0.81 +/- 0.035, p = 0.045) were statistically significant. There was no statistically significant difference between the RW and TC models (p = 0.66).(b). Comparison between all models at time t. There was a statistically significant difference between groups as determined by one-way ANOVA (F(3,396)) = 6.84, p <0.001)). A Tukey post hoc test revealed that the difference in conditional probabilities between the Hybrid model (0.79 +/- 0.045) and the RW model (0.48 +/- 0.078, p = 0.018), the Hybrid model and the TC model (0.44 +/- 0.067, p <0.01=27), and the Hybrid model and the Sampling model (0.65 +/- 0.056, p = 0.042) were statistically significant. There was no statistically significant difference between the RW and TC models (p = 0.65).(c).Comparison between all models at time t.There was not a statistically significant difference between groups as determined by one-way ANOVA (F(3,396)) = 1.79, p = 0.65).

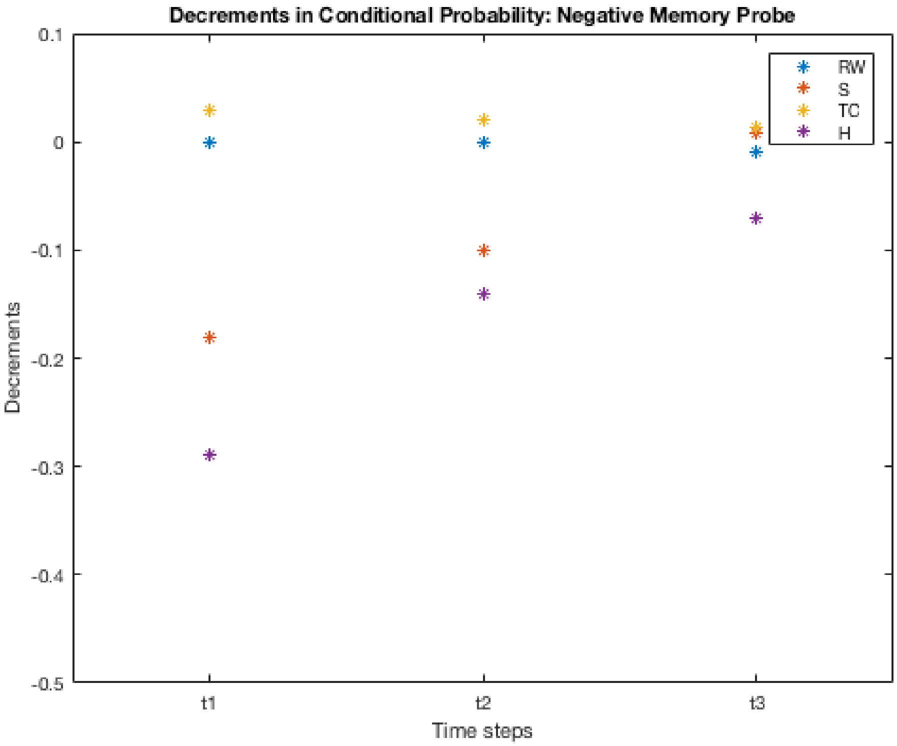

Figure 2.7: Decrements (relative to chance) in Conditional Probability of repeating the same action at times t, t+1 and t + 2 after seeing a memory probe from time t_m at time t given that $r(t_m)$ was negative. (a). Comparison between all models at time t. There was a statistically significant difference between groups as determined by one-way ANOVA (F(3,396) = 7.55 , p <0.001)). A Tukey post hoc test revealed that the difference in conditional probabilities between the Hybrid model (0.21 +/- 0.018) and the RW model (0.5 +/- 0.051, p = 0.002), the Hybrid model and the TC model (0.53 +/- 0.045, p <0.017), and the Hybrid model and the Sampling model (0.32 +/- 0.022, p = 0.038) were statistically significant. There was no statistically significant difference between the RW and TC models (p = 0.86). (b). Comparison between all models at time t. There was a statistically significant difference between groups as determined by one-way ANOVA (F(3,396)= 5.04, p = 0.0019 . A Tukey post hoc test revealed that the difference in conditional probabilities between the Hybrid model (0.36 +/- 0.039) and the RW model (0.5 +/- 0.072, p = 0.041), the Hybrid model and the TC model (0.52 +/- 0.061, p <0.037), and the Hybrid model and the Sampling model (0.4 +/- 0.048, p = 0.047) were statistically significant. There was no statistically significant difference between the RW and TC models (p = 0.92). (c).Comparison between all models at time t. There was not a statistically significant difference between groups as determined by one-way ANOVA (F(3,396)= 2.467, p = 0.636).

Our results show that the Hybrid and the Sampling model have a bias towards choosing an action conditional on the sign of the reward received at the time of the memory probe that was presented. The reason for this bias is that these two models incorporate the memory probe in their algorithms, and therefore, their choices reflect this additional information. We also investigate the effects of these memory probes one and two time steps after the presentation of the probe. The results indicate that there is still an influence of the memory probes in the Hybrid and Sampling models one step in the future. The reason for this is that the Hybrid model's context variable carries information about the previous time step (thanks to its evolution equation, which depends on previous time-steps), and the Sampling model (thanks to its preservation parameter β_c), which has a smaller influence than the one observed in the Hybrid model. It is important to note that the influence of this parameter is not very large, and for that reason, the influence of the reminded trial is less than it was for the previous time step. Similar conclusions are observed for two time steps after the presentation of the cue. The main difference for these probabilities is that the biases from the cue observed are smaller.

We can see that there is a bias in the probability of choice in the Hybrid model. The mechanism for this influence is again due to the context evolution equation. Episodes that happened following the first presentation of the memory probe (episodes occurring after time t_p) integrate the context of the memory probe in their own context. In this way, they have similar contexts to the context of the probed episode. When retrieving episodes based on contextual similarity, all nearby episodes (which are the ones that keep the memory of the context from the memory probe) will be recalled and have a weighted influence on choice. This fact is one of the main contributing factors for the difference in performance between these models. Specially, between the Sampling and the Hybrid models. The Rescorla Wagner and Time Constant models do not have a mechanism to include information from the probe or from nearby episodes, and the Sampling model can only retrieve information from the cued episode. The Sampling model can use some of the influence of the cued episode up to one more time step after it was presented, but then all influence is gone. Furthermore, the Sampling model does not have the property, where other temporally close episodes can influence choices. For all these reasons, data generated by the Hybrid model is difficult to capture by the other models. In the Hybrid model, there is more complexity regarding the influences of previous episodes on choices. This is due to the fact that it uses a temporal model of episodic recall as its base. In the next section, we test the performance of these models with real data.

2.5.2 Analysis of Human Data

We next tested the Hybrid model's performance with human data from the four armed bandit task described in [Daw et al., 2005] (Experiment 1) and the two armed restless bandit task described in [Bornstein et al., 2017] (Experiment 2). We report the parameter estimates and the BIC scores for each model.

Models	**Parameter Values**
RW	$\alpha = 0.67(0.14)$ $\beta = 8.09(0.73)$ $\beta_c = 0.61(0.34)$
	BIC = 330.01 (0.81)
TCs	$\alpha_1 = 0.68(0.23)$ $\alpha_2 = 0.48(0.52)$ $\beta = 6.63(0.12)$ $\beta_c = 0.31(0.13)$ w = 0.66 (0.05)
	BIC = 341.33 (0.87)
Sampling	$\alpha = 0.82(0.11)$ $\alpha_e = 0.33(0.53)$ $\beta = 10.26(0.41)$
	BIC = 319.96 (0.77)
Hybrid	$\alpha = 0.56(0.19)$ $\beta = 6.31(0.03)$ w = 0.65 (0.09) $\beta_b = 0.56(0.27)$
	BIC = 317.02 (0.72)

Table 2.9: Model Fitting Results. In this table, we show the estimates of parameters using data from Experiment 1. We show the parameters learned by the RW, Time Constants, Sampling and Hybrid models. In parenthesis next to the estimated parameters (middle), we show the standard deviation in the parameters' estimates.

Models	**Parameter Values**
RW	$\alpha = 0.45(0.5)$ $\beta = 1.85(0.47)$ $\beta_c = 0.08(0.21)$
	BIC = 115.9 (1.04)
TCs	$\alpha_1 = 0.73(0.14)$ $\alpha_2 = 0.35(0.21)$ $\beta = 2.86(0.58)$ $\beta_c = 0.59(0.33)$ w = 0.69 (0.02)
	BIC = 120.20 (0.64)
Sampling	$\alpha = 0.52(0.34)$ $\alpha_e = 0.37(0.78)$ $\beta = 3.1(0.96)$
	BIC = 107.25 (0.95)
Hybrid	$\alpha = 0.65(0.04)$ $\beta = 3.25(0.73)$ w = 0.46 (0.08) $\beta_b = 0.68(0.29)$
	BIC = 101.79(0.74)

Table 2.10: Model Fitting Results. In this table, we show the estimates of parameters using data from Experiment 2. We show the parameters learned by the RW, Time Constants, Sampling and Hybrid models. In parenthesis next to the estimated parameters (middle), we show the standard deviation in the parameters' estimates.

From this analysis, we can see that the Hybrid model is a better fit than the other models. This suggest that the features we have described about this model manage to capture the complexity of the dynamics of decision making in humans better than previously presented models. In order to compare further the performance between these models, we compute an approximation to the Log-Bayes: Difference of penalized likelihoods using BIC. It was already shown that the Sampling model is better than Rescorla-Wagner, and Rescorla-Wagner is better here than Time Constants model. For that reason, we show the log Bayes scores comparing the Hybrid and Sampling models. Since we are trying to minimize the negative log-likelihood of the data, a positive log Bayes ratio between Sampling and Hybrid model indicates that the Sampling model fits the data worse - its penalized negative log-likelihood is higher. The difference between these two models is similar to the improvements shown in [Bornstein et al., 2017] between the Sampling model and the Rescorla Wagner model. We used this paper as our metric of comparison, and show that our data shows a similar improvements as the ones reported in this previous work.

Table 2.11: Log Bayes scores between Sampling and Hybrid

Experiment	**Scores**
Experiment 1	2.94
Experiment 2	5.46

2.5.2.1 Trial by trial analysis

In order to better understand the reasons for the better performance of the Hybrid model, we perform the same analysis described in the artificial data section regarding the conditional probability of choosing an action given the value of the reward at the time of a probe. Here, we use the data set from experiment 2. The difference between this analysis and the one done with artificial data is that in the latter the data generated was indeed generated by a model that uses the information from the probes. With human data, we are uncertain as to what model humans subject use to make their choices. For that reason we compare the conditional probabilities computed empirically from the human choices of this experiment with the conditional probabilities obtained from using the Rescorla-Wagner, Time Constants, Sampling, and Hybrid models.

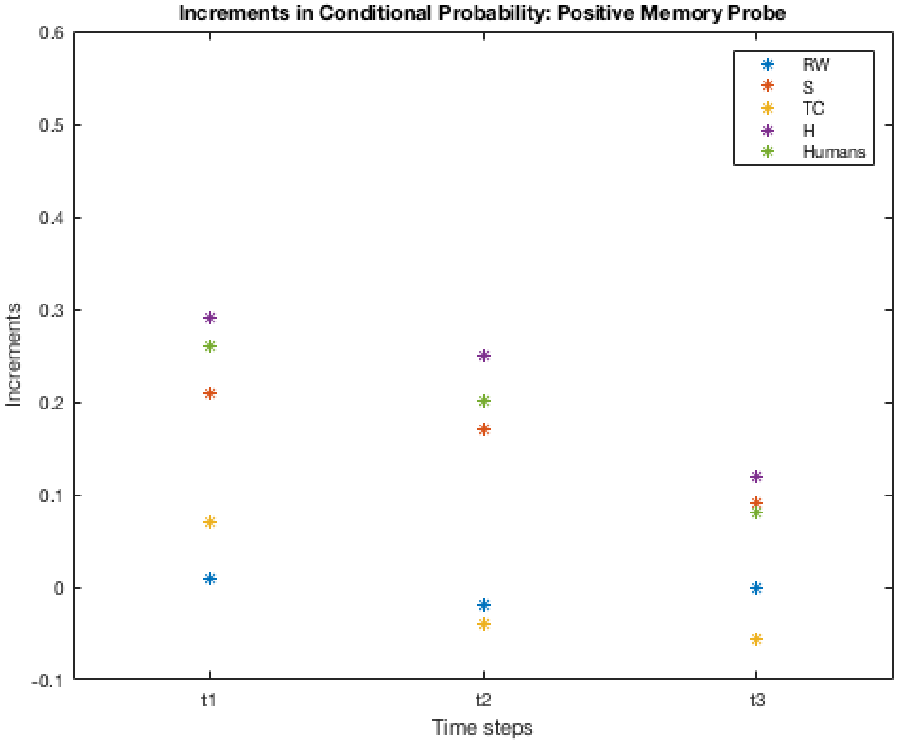

Figure 2.8: Increments (relative to chance) in Conditional Probability of repeating the same action at times t, t+1 and t + 2 after seeing a memory probe from time t_m at time t given that $r(t_m)$ was positive. (a). Comparison between all models at time t. There was a statistically significant difference between groups as determined by one-way ANOVA (F(3,396) = 18.30, p <0.001). A Tukey post hoc test revealed that the difference in conditional probabilities between data from Humans (0.76 +/- 0.45) and the RW model (0.51 +/- 0.6, p <0.001), data from Humans and the TC model (0.537+/- 0.05, p =0.008), and data from Humans and the Sampling model (0.71 +/- 0.03, p = 0.036) were statistically significant. There was no statistically significant difference between the RW and TC models (p = 0.74), and the Hybrid model (0.79 +/- 0.05, p = 0.57) and data from Humans. (b). Comparison between all models at time t. There was a statistically significant difference between groups as determined by one-way ANOVA (F(3,396)) = 6.7, p = 0.011)). A Tukey post hoc test revealed that the difference in conditional probabilities between data from Humans (0.46 +/- 0.002) and the RW model (0.49 +/- 0.03, p <0.001), data fromHumans and the TC model (0.52+/- 0.028, p <0.001), and data from Humans and the Sampling model (0.32 +/- 0.035, p = 0.021) were statistically significant. There was no statistically significant difference between the RW and TC models (p = 0.75), and the Hybrid model (0.41 +/- 0.032, p = 0.97) and data from Humans. (c). Comparison between all models at time t. There was not a statistically significant difference between groups as determined by one-way ANOVA (F(3,396)= 2.467, p = 0.0736). A Tukey post hoc test revealed that the difference in conditional probabilities between data from Humans (0.59 +/- 0.022) and the RW model (0.4 +/- 0.018, p =0.012), and data from Humans and the TC model (0.42+/- 0.021, p =0.015) were statistically significant. There was no statistically significant difference between the RW and TC models (p = 0.74), data from Humans and the Sampling model (0.671 +/- 0.033, p = 0.066) and the Hybrid model (0.63 +/- 0.025, p = 0.65) and data from Humans.

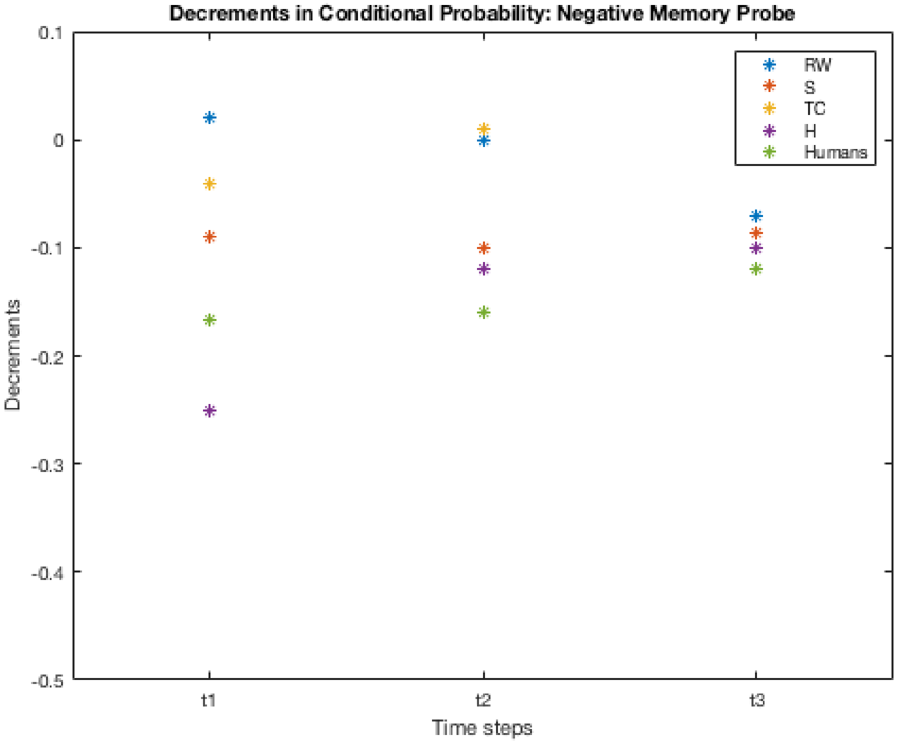

Figure 2.9: Decrements (relative to chance) in Conditional Probability of repeating the same action at times t, t+1 and t + 2 after seeing a memory probe from time t_m at time t given that $r(t_m)$ was negative. (a). Comparison between all models at time t. There was a statistically significant difference between groups as determined by one-way ANOVA (F(3,396) = 9.67, p <0.001)). A Tukey post hoc test revealed that the difference in conditional probabilities between Human data (0.38 +/- 0.035) and the RW model (0.52 +/- 0038, p <0.001), Humans and the TC model (0.46+/- 0.067, p <0.001), data from Humans and the Sampling model (0.41 +/- 0.016, p = 0.009), the RW and the TC model (p = 0.025), and the Hybrid model (0.22 +/- 0.001, p = 0.018) and data from Humans were statistically significant. (b). Comparison between all models at time t. There was a statistically significant difference between groups as determined by one-way ANOVA (F(3,396)= 12.467, p <0.001. A Tukey post hoc test revealed that the difference in conditional probabilities between data from Humans (0.33 +/- 0.044) and the RW model (0.5+/- 0.07, p =0.023), data from Humans and the TC model (0.52+/- 0.06, p=0.014), and data from Humans and the Sampling model (0.4 +/- 0.05, p = 0.033) were statistically significant. There was no statistically significant difference between the RW and TC models (p = 0.74), and the Hybrid model (0.38+/- 0.04) , p = 0.87) and data from Humans (c). Comparison between all models at time t. There was not a statistically significant difference between groups as determined by one-way ANOVA (F(3,396)) = 3.45, p = .045). A Tukey post hoc test revealed that the difference in conditional probabilities between Human data (0.35 +/- 0.03) and the RW model (0.5 +/- 0.0456, p <0.001), Humans and the TC model (0.51+/- 0.038, p <0.001), and Humans and the Sampling model (0.4 +/- 0.03, p = 0.026) were statistically significant. There was no statistically significant difference between the RW and TC models (p = 0.74), and the Hybrid model (0.38 +/- 0.025, p = 0.57) and data from Humans.

Our results show that it is indeed the case that the Hybrid model captures biases in decisions due to memory probes better than other models. The conditional probability of repeating the same action from the time of a probe is more biased in the direction of the reward received when using the Hybrid model than when using the other models. Furthermore, we can see that the Hybrid model's conditional probabilities are closer to the empirical conditional probabilities computed with human data. We continue with an analysis similar to the analysis performed using artificial data in the previous section. The conditional probabilities of humans exhibit biases similar to those of the Hybrid model. This result explains why the Hybrid model is a better fit to human data; furthermore, it shows that humans are biased not only by the probed event, but also by nearby events from that probed event. In this way, suggesting that humans use episodic information in a manner that resembles our proposal. Human subjects recall events based on cues, but also appear to make assumptions about the environment that favor the usage of a temporal context model of recall. In particular, it appears that humans assume that the world can suddenly change and go back to a previous situation, and once they are back in that situation, or a very similar one, the world continues to change slowly. This means that they react to a memory probe, and reinstate a similar mental state to that of the time where that probed event occurred. With the retrieval of this mental state, they also retrieve memories of that event and of nearby events. Consequently, the information from events following the memory probe, also appear to have an effect on the current time step, and subsequent time steps. The memory probe itself also continues to influence decisions in the future. At each time step, humans preserve information from previous time steps. In this way, the information from the probe can still influence decisions one and two time steps after the probe was presented. This feature confirms our hypothesis that humans subjects believe the world changes slowly.

In this chapter, we have shown that an episodic based RL algorithm; particularly, the Hybrid model, is a better fit for human choices than the other models discussed. But more importantly, we have shown that this result implies that humans make certain assumptions about the world, and store and recall information accordingly. In the next chapter, we focus on understanding what these assumptions are, and propose a generative model for our episodic based RL algorithms.

2.6 Conclusions

Memory systems lie at the heart of animal learning, and therefore they also play a central role in Reinforcement Learning. This invites us to understand how distinctions between different memory and different RL systems relate, both mechanistically, and in terms of the different aspects of the environment to which they are tuned. Specifically, model free RL resembles the rigid structure of striatal procedural memory, while model based RL resembles the flexibility of declarative semantic memory associated with the hippocampus and neocortex [Gershman and Daw, 2017] [Poldrack et al., 2001] [Eichenbaum and Cohen, 2004]. These memory systems have been the focus of reinforcement learning research, while other systems—such as working and episodic memory—have been mostly neglected, with few exceptions [Collins and Frank, 2012] [Lengyel and Dayan, 2008] [Gershman and Daw, 2017] [Bornstein et al., 2017]. Episodic memory differs from procedural or semantic memory in that it stores and recalls information about specific events that took place in particular times and places. Contrary to the memory systems already used in RL, it does not store accumulated statistics of the environment. For this reason, its associated RL framework is fundamentally different from existing models.

In this chapter, we extended previously proposed models [Bornstein and Norman, 2017] and introduced a new RL framework that uses episodic memory. We introduced two new RL frameworks: Contextual Episodic and Hybrid—a mixture between Contextual Episodic and the standard Rescorla-Wagner model. The Contextual Episodic model uses episodic memories to make decisions about actions. This model recalls episodic memories based on contextual cues to make decisions. We used the Temporal Context Model (TCM) [Howard and Kahana, 2002] to model contextual integration, drift and recall. Episodic memories are each associated with the context in which they occurred. Episodes are retrieved, weighted and integrated during decision-making, based on the similarity between their associated contexts and the current context. Episodes are then weighted based on this contextual similarity to the current context, and integrated to make a choice. This use of past events based on contextual similarity is the core pillar of the Contextual Episodic model. Despite the well-established influence of context and contextual cues on decision-making [Duncan and Shohamy, 2016] [Bornstein and Norman, 2017], no previous RL model has bound episodes to context. Furthermore, thanks to the temporal structure of TCM, our model organizes the storage and retrieval of episodes based on temporal contexts. This feature allows for the flying-back-in-time property of episodic recall to also be present in RL-based decision-making. When a past episode is cued in the present, decision-making is influenced by not only that particular episode but also episodes that followed the cued episode. In this way, the Contextual Episodic model is a temporally dynamic model of decision-making.

Subsequently, we proposed the Hybrid model to address the fact that human behaviour often incorporates elements from both model-based and model-free choice, and that contextual retrieval is omnipresent. Standard RL algorithms are often deployed simultaneously to optimize decision-making. Similarly, the Rescorla Wagner and Contextual Episodic algorithms may be combined to make more accurate choices since they provide different types of information. Furthermore, even when choices are made using a model-free method, different contextual cues trigger the recall of similar contexts with their associated episodes. In this way, episodes are always being recalled, and the more relevant they are to the current context, the stronger is their influence on current decisions. In our proposal, the Hybrid model is a weighted combination of model-free and episodic sources, which allows for a more flexible algorithm.

We tested this framework on both simulated and human data, and found that the Hybrid model is a better fit to human data choices. First, we designed an artificial two-armed bandit task and showed (unsurprisingly) that contextual cues bias the probability of choices only in models that incorporate both of these cues such as the Hybrid and Sampling models, but not in conventional models such as the Rescorla Wagner and Time Constants models. We further showed that the effect of a contextual cue in the Hybrid model lasts for the next two time steps after the presentation of the cue, while in the Sampling model this effect only lasts for a single further time step. This feature of the Hybrid model is a results of the contextual drift characteristic of TCM. Context slowly drifts through time; therefore, at any time point, the current context preserves information from previous time steps. When a cue is presented, the current context is updated and it is used by the Hybrid model to search in memory for similar contexts. This effect remains present during the next time steps, while the memory of the contextual cue is still present.

Furthermore, we showed that it is not only the cued episode that influences decisions in the Hybrid model, but also episodes that immediately followed the original presentation of the cued episode. Thanks to the memory property of TCM previously described, episodes that occurred right after the cued episode original presentation have similar contexts. For this reason, these episodes are also retrieved by our contextual similarity retrieval mechanism. An advantage of the Hybrid model is that it weights information from previous episodes based on relevance—measured as contextual similarity, thus more than one episode from the past can influence decision-making based on relevance rather than recency. The Sampling model combines the information from all episodes, and weights them by recency (except for the cued episode, which enjoys its own unique weight). It is this influence of relevance that distinguishes the Hybrid and Sampling models. In the latter, only the cued episode has a stronger influence on the action choice. On average, all other episodes have similar degree of influence as episodes would have in the Rescorla Wagner model. Thus, it is the influence of contextual relevance that distinguishes the Hybrid and Sampling models.

Nevertheless, this chapter also discussed the similarities between the Hybrid model and Sampling model because they both use episodic information and combine it with a form of recency based integration of previous episodes. In the case of the Sampling model, this is a consequence of sampling a value from the entire history of episodes weighted by recency, whereas in the Hybrid model, it is a consequence of the low pass filter nature of the TCM and the weighted contribution of the Rescorla Wagner model. The Hybrid model, however, can adjust a parameter that weights how much of this information is useful, which gives the Hybrid model another advantage: flexibility in its use of information. This extends beyond weighting episodes based on contextual similarity to switching between situations where either episodic information is more relevant or a model-free strategy is more optimal. For example, if a novel situation is encountered, the Hybrid model can easily increase the weight of episodic information. Conversely, when retrieving episodes and weighting them becomes an unnecessary and heavy load on working memory (i.e. the learner has mastered the task), the Hybrid model can switch partially or completely to a model-free system.

In this chapter, we showed that due to its unique features, the Hybrid model outperforms other models using simulated data (unsurprisingly since it was designed to test this model) and human data. These results suggest that - a temporal context model and episodic memory are present during decision making. While the temporal context model has been shown to robustly predict free recall in humans, this is the first time that this framework is used to model the retrieval of contextually relevant episodes to make decisions. A key distinction between free recall models and contextual episodic models—and hybrids—is that to optimize decision-making, contextual episodic models retrieve and weight all episodes to integrate the information of more than one episode, as opposed to the single episode selection that occurs in free recall models.

The fact that human choices are closer to those suggested by the Hybrid model suggests that at least some of the implicit assumptions humans make about the structure of the world are captured by this model. We do not claim that the Hybrid model is the algorithm used by humans, or that the conditions of the environment that this model assumes are exactly those assumed by humans. However, we have observed that some features of this model are used by humans, and therefore understanding the assumptions behind this model could be useful to understanding some of the assumptions made by humans.

An extension to this current work would be to explore how this weighting parameter is adapted dynamically and online. In our work, we fit the weight parameter w for each task, but in humans, this parameter may be dynamically adjusted from trial to trial as the subject explores the environment and gains more expertise. Learned, or expected levels of uncertainty may afford cognitive resources necessary for episodic retrieval, while unexpected uncertainty may require quick action-value assessments for optimal performance . Indeed, several neuromodulators have been associated to changes in environmental uncertainty [Angela and Dayan, 2005] and would be prime candidates for the biological substrate for adjustments to parameter w.

Our model also opens the door to further analysis related to the understanding of how memories are integrated during decision-making, problem solving and creative thinking. In our approach, model-free approaches were compared or combined with episodic-based learning. However, model-based algorithms may serve as more adequate recognition models in cases where inferences regarding the temporal causal structure are required for maximal reward. Future work should extend the Contextual Episodic model ideas to a model based framework. Our proposed framework is a proof-of-concept for a Contextual Episodic decision making algorithm. The definition of context could be expanded to include aspects; such as, semantics, emotions, and multiple contexts overlapping at different temporal time scales. Some of these aspects have already been addressed, and used to expand the notion of the TCM beyond temporal context [Polyn et al., 2009] [Talmi et al., 2019]. In the future, we would like to expand on this even further and see how these different features affect decision-making. In particular, we are interested in understanding biases in recall based on emotions. We would like to test for negative biases by adding extra features that code for positive and negative contexts, and test whether patients with psychiatric disorders exhibit these biases. These biases could be incorporated during context encoding, retrieval or weighting. Understanding when and how these biases affect decision-making would be a fruitful research direction.

Finally, an important direction for future work would be to expand these decision making models to problem solving and creative thinking, since these three cognitive computations share the core pillar of integrating previous memories to assess the present. The case of creative thinking is particularly interesting for us, since it requires integrating memories that are seemingly out of context, but turn out to be key to finding novel solutions. To build a model of creative thinking, we would need to expand not only our definition of context, but our mechanism of retrieval.

Decision-making algorithms are based on inductive biases, or inferences made by the learner about the structure of the world that generates observations and rewards. In the next chapter we explore this further, presenting a generative model to which the Contextual Episodic model and Hybrid models are matched. We first study the conditions of the world for which a Contextual Episodic model would be an appropriate inference algorithm. We then explore the scenario where a learner would combine this model with the Rescorla Wagner model. By describing a generative model for these learning algorithms, we present a formal normative approach for further study into the conditions for which this model would be an optimal strategy. In particular, we study when and why humans might use temporal episodic memory-based strategies for decision-making.

Chapter 3

Generative Model

3.1 Motivation

In the previous chapter, we introduced a learning framework based on episodic recall. We further introduced the concept of contextual similarity based recall, and we described how events from far in the past can be relevant and used for decision making in the present. We called this learning framework: Contextual Episodic. In this chapter, we introduce a generative model for which the Contextual Episodic learning framework is a suitable approximate inference model, and which therefore specifies an environment to which Contextual Episodic decision-making is likely to be effective.

A generative model is a statistical construct that describes the process by which observable variables are generated [Ng and Jordan, 2002], [Bernardo et al., 2007]. In the context of our experiments, it describes the process that generates the contextual cues and the bandits with their respective associated rewards. In other words, the generative model captures the environment the agent explores, and in this way, determines the requirements for appropriate inference.

Formally, we call the data, observable variables D, and describe the statistical model as the probability $P(D|\theta)$, where θ are themselves random variables that act as parameters in this stochastic generative process. This probability is called the Likelihood function or Likelihood of the data [Ng and Jordan, 2002]. We assume that a learning agent does not start off knowing the true values of θ; but only has a prior distribution over these values. However, the learning agent can compute these parameters using an inference model. An inference model computes the posterior probability $P(\theta|D)$ based on prior beliefs and conditional on the observed data D. Exact posterior inference is computed using Bayes' rule:

$$Posterior \propto Likelihood * Prior \tag{3.1}$$

where the Likelihood function is the probability given by the generative model $P(D|\theta)$ and the prior and posterior are probabilities taken over the model's parameters:

$$P(\theta|D) \propto P(D|\theta)P(\theta) \tag{3.2}$$

In the context of our bandit tasks, the learning agent wants to infer the value of these parameters, so that he or she can make predictions about future rewards. Specifically, after accumulating data from previous rewards (and other observations if there are any such), the learning agent can compute the posterior over the parameters that define the model of the environment, and use this learned parameterized model to predict future outcomes. Using these predictions, the learning agent can make rational decisions about future actions and choices. In Bayesian treatments of decision-making, rationality is defined as performing exact inference over the latent causes or generative model of an environment and guiding behaviour and choices accordingly.

Often there are constraints on the ability of agents to perform exact inference, due to physical limitations, intractable computations or ignorance of the exact generative model of the environment. Constraints such as these give rise to bounded rationality [Howes et al., 2009]. It is very important to acknowledge that even though exact inference is the optimal solution for rational decision making, often times biological and artificial agents are limited and thus approximations to inference are necessary. In this work, we introduce a generative model, for which exact inference is difficult, and thus we present our learning algorithms as suitable approximations.

3.1.1 Generative Model for Rescorla-Wagner

One example of an inference model in the reinforcement learning literature with an associated generative model is the Rescorla-Wagner (RW) algorithm [Rescorla et al., 1972] [Dayan et al., 2000] [Daw et al., 2008]. It has been shown that the RW algorithm is equivalent to the Kalman Filter (at least in the regime in which the Kalman gain has reached an asymptote), which performs exact inference in a generative model known as the Linear Gaussian State Space Model (LGSSM) [Sutton, 1992] [Dayan et al., 2000] [Daw et al., 2008]. Here, as a foundation for our own work, we describe the RW algorithm, and introduce a generative model for which this learning algorithm computes optimal inference. We show how this generative model is a LGSSM.

The Rescorla-Wagner model is a formal model of classical conditioning. It describes how animals learn associations or predictive relationships between stimuli (conditioned stimuli) and rewards or punishments (unconditioned stimuli) [Rescorla et al., 1972]. These predictions apparently guide behavioral responses [Rescorla et al., 1972] [Dickinson, 1980] [Mackintosh, 1983]. For example, during a particular classical conditioning experiment, animals might learn to predict that food comes after the sound of a bell, or that an electric shock comes after the presentation of a red light. Consequently, they salivate or freeze, respectively.

Formally, this model is described by the RW learning rule [Rescorla et al., 1972] (which is equivalent to the delta rule; Widrow & Hoff, 1960). This rule makes predictions of rewards given a stimulus. To derive this rule, we consider a vector $\mathbf{x}_t$ given on trial t, which represents the presence or absence of the conditioned stimuli. Often, the entries of $\mathbf{x}_t$ will be binary – but this is not necessary for what follows. The delivery of the unconditioned stimulus (the reward) is represented by the scalar r_t. The predictive relationship between $\mathbf{x}_t$ and r_t is captured by a set of parameters $\mathbf{w_t}$, and in this model, it is assumed to be linear: $V_t = \mathbf{x}_t \cdot \mathbf{w_t}$. The goal of a RW learning agent is to make adequate predictions of the reward it will receive upon presentation of the conditioned stimuli by acquiring an appropriate value for $\mathbf{w_t}$. RW is a learning rule that (expressed in modern terms) updates this value at each time step in the following way [Rescorla et al., 1972]:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha \delta_t \mathbf{x}_t \tag{3.3}$$

where α is the learning rate and δ_t is equal to the error between the reward received and the expected reward:

$$\delta_t = r_t - V_t = r_t - \mathbf{x}_t \cdot \mathbf{w}_t \tag{3.4}$$

In order to interpret the Rescorla-Wagner algorithm as an inference model, we consider a statistical approach to the prediction problem between the conditioned stimuli and the reward. To do this, we describe a space of statistical hypotheses about the data generation process or generative model of the data. This generative model has two components. One is the likelihood: a parameterized probability distribution over each data set $D = \{\mathbf{x}_t, r_t\}$ of stimuli-reward pairs: $P(D|\theta)$, where $\theta = \{\mathbf{w}_t\}$ in this model represent the latent variables $\mathbf{w}_t$ that characterize the generative process (by capturing the association between each stimuli in $\mathbf{x}_t$ and the reward r_t). The second component is the prior $P(\theta)$. For classical conditioning, it suffices that this model describes how rewards are generated on each trial conditioned on the stimuli observed [Daw et al., 2008]. In other words, the generative model for classical conditioning is a conditional generative model. It defines a conditional probability distribution of reward r_t given stimuli $\mathbf{x}_t$. A full generative model for data D would model the joint probability distribution of both variables jointly [Daw et al., 2008]. The conditional probability distribution of reward r_t given stimuli $\mathbf{x}_t$ is given by a Gaussian distribution with mean equal to $\mathbf{w}_t \cdot \mathbf{x}_t$ and observation variance σ:

$$P(r_t|\mathbf{x}_t, \mathbf{w}_t; \sigma) \sim \mathcal{N}(\mathbf{w}_t \cdot \mathbf{x}_t, \sigma) \tag{3.5}$$

Additionally, the prior $P(\theta)$ specifies that latent variables $\mathbf{w}_t$ evolve according to a first order Gaussian drift with mean centered around $\mathbf{w}_{t-1}$ and covariance equal to Σ_w:

$$P(\mathbf{w}_t|\mathbf{w}_{t-1};\Sigma_w) = \mathcal{N}(\mathbf{w}_{t-1},\Sigma_w) \tag{3.6}$$

The likelihood and prior jointly define a generative model for the delivery of reward, where the optimal inference algorithm is a form of the Rescorla-Wagner learning rule related to the Pearce-Hall rule [Dayan et al., 2000] [Pearce and Hall, 1980] [Sutton, 1992]. If we define the prior distribution over $\mathbf{w}_t$ at time $t = 1$ to be a Gaussian distribution with mean given by $\mathbf{w}_o$ and covariance Σ_{w_o} :

$$P(\mathbf{w}_{t=1}) = \mathcal{N}(\mathbf{w}_o,\Sigma_{w_o}) \tag{3.7}$$

Then this conditional generative model is equivalent to a Linear Gaussian State Space Model (LGSSM), for which it has been shown that the exact inference algorithm is the Kalman Filter [Dayan and Kakade, 2001]. This shows how the generative model for which the Rescorla-Wagner algorithm performs exact inference is equivalent to a LGSSM. To see this, observe that the Kalman Filter computes the posterior distribution of $\mathbf{w}_t$ given all previous observations with mean $\hat{\mathbf{w}}_t$ and the covariance matrix $\hat{\Sigma}_t$:

$$P(\mathbf{w}_t|r_1...r_{t-1};\mathbf{x}_1...\mathbf{x}_{t-1}) = \mathcal{N}(\hat{\mathbf{w}}_t,\hat{\Sigma}_t) \tag{3.8}$$

And updates it according to [Kalman, 1960] [Welch et al., 1995]:

$$\hat{\mathbf{w}}_{t+1} = \hat{\mathbf{w}}_t + \hat{\kappa}_t(r_t - \hat{\mathbf{w}}_t \cdot \mathbf{x}_t) \tag{3.9}$$

$$\hat{\Sigma}_{t+1} = \hat{\Sigma}_t - \hat{\kappa}_t\mathbf{x}_t^T\hat{\Sigma}_t + \Sigma_w \tag{3.10}$$

where κ is the Kalman gain vector:

$$\hat{\kappa}_t = \frac{\hat{\Sigma}_t\mathbf{x}_t}{\mathbf{x}_t^T\hat{\Sigma}_t\mathbf{x}_t + \sigma} \tag{3.11}$$

$$\tag{3.12}$$

These equations describe the Kalman filter updates for inference in the LGSSM. It can be seen that the update rule for the mean is equivalent to the Rescorla-Wagner learning rule when $\hat{\Sigma}_t \propto I$.

Intuitively, the Rescorla-Wagner rule averages past rewards based on recency. It can be seen from the generative model that indeed the random variable **w** gradually diffuses over time, so that values are more similar to those of nearby trials than to those that are further apart. For this reason, a recency-based inference model is adequate. A generative model helps us understand the requirements for inference and predictions, and likewise different inference algorithms imply learning in a different generative model.

3.1.2 Generative Model Proposal

Motivated by the Rescorla-Wagner model with its associated generative model, we would like to understand the conditions of the environment for which an episodic-based algorithm; such as the Contextual Episodic model or the Hybrid model, are adequate strategies for a reward-based learner. In last chapter, we showed that human decision-making had similarities with our algorithms; for this reason, we would like to understand the assumptions of the world necessary for an organism to choose these strategies. In order to draw statistically rigorous conclusions for the type of tasks or environments where using our algorithm would be adequate, we need to understand the structure of the environment in which it is trying to perform inference. Our Contextual Episodic algorithm does not weight rewards by recency, but rather by contextual similarity. This fact indicates us that the Contextual Episodic algorithm is not trying to perform inference in a LGSSM. Therefore, we need to propose a different generative model. Furthermore, the Contextual Episodic algorithm uses a temporal context model (TCM) [Howard and Kahana, 2002] that integrates information from events using a context vector, which is updated as a low pass filter of the previous contexts, but it is also updated by new inputs. These inputs can be similar to the previous context, to a context far in the past, or to be completely new. We explore the necessary characteristics for the corresponding generative model of this inference process. In other words, we explore the conditions under which being able to learn and integrate all the described contextual information would be necessary to perform inference. Consequently, we conclude that our generative model will have to generate data that changes slowly in time, but also suddenly can change to previous or new situations.

Here, we elaborate these requirements and propose a suitable generative model. For that, we first review a previous generative model that uses the Temporal Context Model [Howard and Kahana, 2002]. Then, we review the Dirichlet Process [Teh, 2010] [El-Arini, 2008]. In particular, we review the Infinite Hidden Markov Model (IHMM) [Beal et al., 2002], because we use this model for our generative model. Finally, we introduce our generative model, and analyze the implications of different parameter regimes for approximate inference with the Contextual Episodic algorithm. In this way, we show different situations when our new model is an adequate inference algorithm.

3.2 Literature Review

3.2.1 Previous Model

We begin this literature review with an overview of a conditional generative model of human memory performance in free recall developed by Socher et. al [Socher et al., 2009]. Free recall is studied in subjects that memorize a list of words, and then are asked to recall these words. Models of verbal memory assume that words are formed and retrieved based on assimilated semantic representations. In this model, Latent Dirichlet Allocation (LDA) [Blei et al., 2003] is used to represent this latent semantic structure [Socher et al., 2009]. LDA represents the meaning of documents as a distribution over topics, which themselves are a distribution over words. Free recall is then modeled by combining the latent representations provided by LDA with a slowly changing temporal latent context[Socher et al., 2009]. The Temporal Context Model (TCM) [Howard and Kahana, 2002] is used to model this latent temporal structure. In essence, this conditional generative model assumes that memories of words are formed as distributions over topics that are assimilated into a slowly a changing latent context. Using this conditional generative model, Bayesian inference can be used to make predictions about recalled words [Socher et al., 2009]. The goals of this model are closely related to our intent in this chapter: formulate a generative model for temporal and contextual-based recall inspired by TCM. The main difference between this work and our work is that the work by Socher et. al proposes a conditional generative model based on TCM and LDA [Socher et al., 2009] which, like the RW analysis, uses inputs as a fixed part of the generative process (details below). On the other hand, we will propose later in this chapter a full generative model, where a TCM-based algorithm is used as an approximate inference model. Here we describe the different components of this conditional generative model. In the next section, we describe our proposed full generative model.

Socher et al. consider both a study phase and a recall phase. In experiments examining free recall, which are the focus of the TCM, subjects first have to memorize a list of words (study phase), and then they need to recall these words (recall phase). Socher's model is based on TCM, but it re-interprets it as a dynamic latent variable model, where the context vector represents a distribution over topics and the contextual drift can be seen as a sequence of mixtures of topics [Socher et al., 2009].

Here we describe the two phases. We label the context during study phase $\mathbf{c}^s$, and the context during recall phase $\mathbf{c}^r$. The study phase specifies the trajectory of mental context by first drawing the initial mental context from a Gaussian [Socher et al., 2009]. The subscript n refers to the current studied word, and it is used to identify the context and probabilities during the study phase:

$$P(\mathbf{c}^s_o) \sim \mathscr{N}(\mathbf{0}, \sigma I) \tag{3.13}$$

then, using the new studied words as inputs, the mental context drifts according to the following equation:

$$P(\mathbf{c}^s_n) \sim \mathscr{N}(\mathbf{h}^s_n, \sigma I) \tag{3.14}$$

where $\mathbf{h}^s_n$ evolves with the following equation

$$\mathbf{h}^s_n = \eta_1 \mathbf{c}^s_{n-1} + (1-\eta_1)\log(\mathbf{p}^s_n) \tag{3.15}$$

where $\mathbf{p}^s_n \sim \beta^s_{w_n}$. $\mathbf{p}^s_n$ is the posterior probability of each context given the current studied word, and $\beta^s_{w_n}$ is the distribution over words given a context containing the current studied word w_n

$$\tag{3.16}$$

These equations show how the mental context drifts when the subject studies new words. The two forces governing this drift are the previous mental context $\mathbf{c}^s_{n-1}$ and the posterior probability $\mathbf{p}^s_n$. The parameter η_1 controls the contribution of each of these forces [Socher et al., 2009].

The recall phase specifies the distribution of recalled words and the dynamics of the drifting context during this recall process. First, we describe how subjects recall words after the study phase. At time step t, a recalled word is generated from a mixture of ”paths”: semantic and episodic [Socher et al., 2009]. The semantic path recalls words according to an LDA generative model using the current context as the distribution over topics. This implies that a topic is drawn from this distribution, and then a word is drawn from this topic. Formally, word w is recalled via the semantic path with a probability induced by the current context [Socher et al., 2009]. Here the index t refers to the time step of the recalled word.:

$$P_{se}(w) = \pi(\mathbf{c}^r_t)\beta_w \tag{3.17}$$

In this equation, π represents a function that maps vectors onto the simplex, where positive vectors sum to one, and β_w is distribution over words. Then, $P_{se}(w)$ represents the marginal probability of recalling word w given the current recalled context.

The episodic recall path draws words from the set of studied words, with probability equal to a weighted sum of δ functions defined at each study word, where the weight for each function is based on the contextual similarity between the current context and the context when each word was studied. This contextual similarity is measured by the function $d(\cdot,\cdot)$, which corresponds to the negative KL-divergence. In the equation below, ε is a parameter that controls the curvature of this similarity function. Therefore, the probability of recalling word w via the episodic path is equal to [Socher et al., 2009]:

$$P_e(w) = \frac{\mathbf{u}_{t,w}}{\sum_i \mathbf{u}_{t,i}} \tag{3.18}$$

where $\mathbf{u}_t$ is obtained by summing over all studied words:

$$\mathbf{u}_t = \sum_{n=1}^{N} \frac{\delta_{s,w_{sn}}}{d(\pi(\mathbf{c}_t^r),\pi(\mathbf{c}_n^s))^{\varepsilon}} \tag{3.19}$$

A recalled word is generated from a mixture of these two paths. This mixture is determined by a mixing proportion γ. Therefore, a word is drawn during recall according to [Socher et al., 2009]:

$$w_{rt} \sim Mult(\phi_t) \tag{3.20}$$

where:

$$\phi_t(w) = \gamma P_{se}(w) + (1-\gamma)P_e(w) \tag{3.21}$$

During this recall phase, context drifts according to the following equation:

$$P(\mathbf{c}_{t+1}^r) = \mathcal{N}(\mathbf{h}_t^r, \sigma I) \tag{3.22}$$

where:

$$\mathbf{h}_t^r = \eta_2 \mathbf{c}_t^r + \eta_3 log(\mathbf{p}_t^r) + \eta_4 \mathbf{c}_{n(w_{r,t})}^s \tag{3.23}$$

where, similar to the study phase, $\mathbf{p}_t^r$ is is the posterior probability of each context given the current recalled word.

These equations are very similar to the evolution of context during the study phase. The only difference is that, during recall, context is also driven by the retrieved context (the context that was present when the recalled word was studied). Formally, this corresponds to η_4, which maps a recalled word to its index when it appeared during the study period [Socher et al., 2009].

This model is a conditional generative model of free recall. It uses input words during the study phase as part of the generative model. It does not specify the statistics of the environment that would generate these words. It uses the input words to construct probabilities over topics. Furthermore, it uses the temporal context model as part of the generative model. In this thesis, we propose a TCM inspired algorithm as an inference algorithm, and then in this chapter, we will propose a full generative model for which such a contextual and temporal algorithm is an appropriate approximate inference method. In the following section, we present this full generative model of episodic events applied to a reinforcement learning task. The goal of inference in our model is to predict the values of actions based on previous rewards, and previous and current observations. In this way, we want our inference algorithm to guide agents to make adequate choices.

3.2.2 Dirichlet Processes

The Dirichlet process is a Bayesian non-parametric model, which defines a distribution over distributions [Teh, 2010], [El-Arini, 2008], [Ferguson, 1973]. This means that each draw from the Dirichlet process is also a distribution. These distributions are discrete, and cannot be described using a finite number of parameters. For this reason, these models are called non-parametric [Teh, 2010]. The Dirichlet process is specified by a base distribution and a concentration parameter:

$$G \sim DP(\alpha, H) \tag{3.24}$$

where α is the concentration parameter, and H is the base distribution.

The Dirichlet process draws distributions around this base distribution in a similar way that a normal distribution draws samples around its mean. The scaling parameter specifies how concentrated or spread out the discrete distributions drawn from the DP are. A higher value of α means that distributions are less concentrated or more spread out. Intuitively, the Dirichlet Process (DP) is an infinite dimensional generalization of the Dirichlet distribution, and it is the conjugate prior for infinite, non-parametric distributions; such as, infinite mixture models or infinite Hidden Markov Models (HMMs). In this section, we describe this infinite generalization of HMMs: the Infinite Hidden Markov Model.

3.2.2.1 Infinite Hidden Markov Model

The Infinite Hidden Markov Model was proposed by Beal et. al [Beal et al., 2002] to extend the Hidden Markov Model (HMM) to have infinite number of states. Standard HMMs are used to model sequential data; such as, speech, music, or protein sequencing. They define a probability distribution over observed sequences by describing the evolution of another sequence of unobserved or hidden states. The observed sequences have unknown temporal dynamics, and are conditionally independent given hidden states. On the other hand, hidden states have Markov dynamics: the hidden state at time t only depends on the hidden state at time $t-1$. Therefore, conditioned on this state, it is independent of the previous state sequence. We define the hidden states as Y_t and the observations as o_t. To model the hidden states, we only to define the transition matrix, with the ij_{th} entry equal to $P(Y_t = j|Y_t = i)$. These hidden states emit observations with emission probability matrix, with the iq_{th} entry equal to $P(o_t = q|Y_t = i)$. Since the hidden sequence is independent of the past given the previous state of the sequence, the observed variables are independent of their history given the value of the hidden state [Beal et al., 2002].

HMMs have important limitations for modeling sequential data. First of all, the modeler needs to specify in advance the structure, such as number of parameters of the model, of the sequence, and knowing this structure is often not possible a priori. Second, it is often unreasonable to assume that the data was generated by a set of discrete states. For this reason, iHMMs propose that data can be generated by countably infinite hidden states. iHMMs are a non-parametric extension of HMMs to infinite possible transitions. Therefore, they do not require the experimenter to specify the structure of the task in advance, and the number of states can grow depending on the complexity of this task. This infinite realization is achieved thanks to the usage of Dirichlet processes, which make it be possible to integrate out the infinities. We describe iHMMs below, and show the equations and parameters needed for drawing samples and optimization [Beal et al., 2002].

The infinite Hidden Markov Model extends the HMM model by considering each row of the transition and emission probability matrices of the HMM as a Dirichlet Process. One of the advantages of using DPs is that we can, we integrate out the Dirichlet prior and get a set of conditional probabilities with only three hyper-parameters. These probabilities depend only on the three hyper-parameters $(\alpha, \beta, and \gamma)$ corresponding to the two Dirichlet process, and on the number of transitions from each state to another one following the transition matrix and emissions or observations from each state following the emission matrix [Beal et al., 2002].

Before we present the set of equations for these conditional probabilities, we introduce the variables n_{ij} and m_{iq} that count the number of transitions from state $Y = i$ to state $Y = j$, and the number of emissions of symbol $o = q$ from state $Y = i$, respectively. Similarly, to the HMM, each state Y_t is hidden, while each emission o_t corresponds to the observation at each time step.

The variable n_{ij} is defined as:

$$n_{ij} = \sum_{t'=1}^{t-1} \delta(Y_{t'}, i)\delta(Y_{t'+1}, j) \tag{3.25}$$

The variable m_{iq} is defined as:

$$m_{iq} = \sum_{t'=1}^{t-1} \delta(Y_{t'}, i)\delta(o_{t'}, q) \tag{3.26}$$

We further introduce the variables $n_i^{\diamond}$ and $m_q^{\diamond}$ that respectively count the number of times state $Y = i$ transitioned to new state it has not visited before, and the number of times state $Y = i$ emitted new symbol [Beal et al., 2002].

We now define the equations for the the generation of the hidden state sequences in an iHMM. States that have already appeared up to time t in the sequence are labeled states 1 to k. The three hyper-parameters $\alpha, \beta, and \gamma$ each control how the hidden state sequence is generated. α controls the prior tendency to remain in a state. β controls the tendency to explore transitions to new states. Finally, γ controls the expected number of represented hidden states. Each state can transition to itself, to a state it has transitioned before, to a state that has appeared in the sequence but it has not transitioned to it before, or it can transition to a completely new state. The probability of a state remaining the same is proportional to the sum of number of transitions it has had to itself before and a parameter α. This probability determines the time-scale of evolution of the iHMM [Beal et al., 2002]:

$$P(Y_{t+1} = i | Y_t = i; \mathbf{n}, \beta, \alpha) = \frac{n_{ii} + \alpha}{\Sigma_{j'=1}^{K} n_{ij'} + \beta + \alpha} \tag{3.27}$$

The state can also transition to a state from the history of states it has visited in the past. This probability leads to common transitions, and it is proportional to the number of times it has transitioned to each state in its past [Beal et al., 2002]:

$$P(Y_{t+1} = j | Y_t = i; \mathbf{n}, \beta, \alpha) = \frac{n_{ij}}{\Sigma_{j'=1}^{K} n_{ij'} + \beta + \alpha} \tag{3.28}$$

$$j \in 1, ..., K \tag{3.29}$$

With a certain probability, the state can transition to an oracle. The probability of the transition to this oracle is proportional to a parameter β [Beal et al., 2002]:

$$P(oracle | Y_t = i; \mathbf{n}, \beta, \alpha) = \frac{\beta}{\Sigma_{j'=1}^{K} n_{ij'} + \beta + \alpha} \tag{3.30}$$

Once in the oracle, the state can transition to a state that it has never transitioned to in the past, but that has previously appeared in the trajectory - a state that has appeared in the sequence of hidden states, but not as a transition following the current state but some other state in the trajectory. This transition probability depends on the number of times the state has used the oracle to make a transition $n^{\diamond}$ [Beal et al., 2002]:

$$P(Y_{t+1} = j | Y_t = i; \mathbf{n}^{\diamond}, \gamma) = \frac{n_j^{\diamond}}{\Sigma_{j'=1}^{K} n_{j'}^{\diamond} + \gamma} \tag{3.31}$$

The second alternative once in the oracle is that a state can transition to a completely new state - a state that has never been visited in the past. This probability is proportional to the parameter γ [Beal et al., 2002]:

$$P(Y_{t+1} = j | Y_t = i; \mathbf{n}^{\diamond}, \gamma) = \frac{\gamma}{\Sigma_{j'=1}^{K} n_{j'}^{\diamond} + \gamma} \tag{3.32}$$

The generation of hidden states is controlled by the three hyper-parameters: α, β, γ. The parameter α determines how quickly the iHMM changes states, the parameters β and γ control how much trajectories deviate from typical trajectories, and how frequent new states are visited. These new transitions or visited states generate jumps in the trajectory, which is the characteristic that we are looking for in this model. Furthermore, the parameter γ determines the variety of states visited from the countably infinite states [Beal et al., 2002]. Figure 3.1 shows the different state trajectories that can be generated by different parameters using the iHMM. We use the iHMM to model the agent's dynamics i.e. the locations in the world that it visits (Figue 3.2 shows how the agent visits different locations each time step). We use the index of the different states to represent locations in the environment i.e. state 1 corresponds to location 1 (index = 1). We are interested in a model that can generate these trajectories, because we want to design an agent that can visit locations sequentially, but can also jump to new locations or visit previous locations. For example, Figure 3.1 (c) shows a trajectory that visits locations sequentially, while Figure 3.1(b) shows some sequential trajectories interrupted by discontinuous visits to other locations.

The emission transitions are similar in nature to the transition probabilities, with the exception that there is no such a thing as ”self-transition” - probability to transition to the same state. At each state there is a probability of emitting a symbol

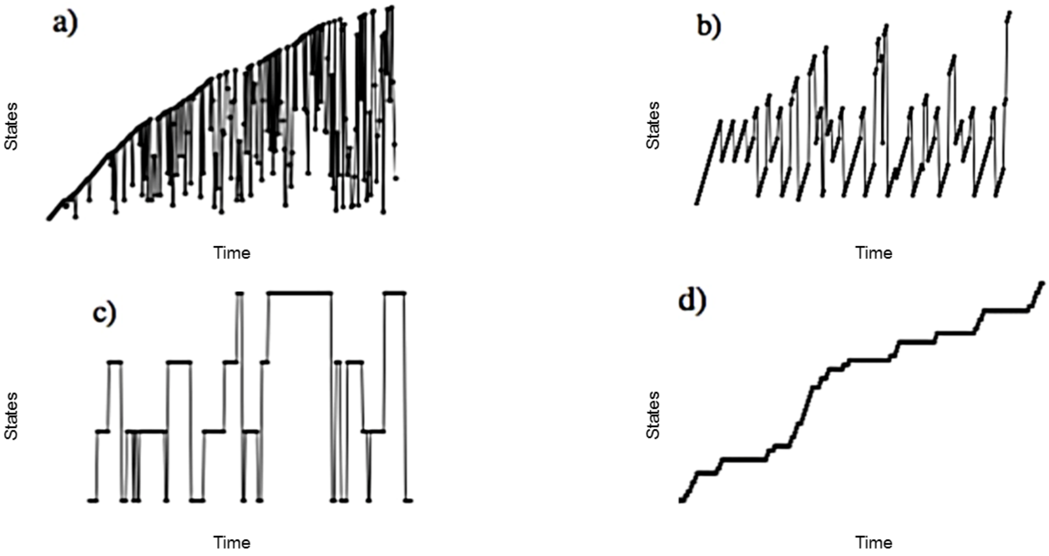

Figure 3.1: Infinite Hidden Markov Model: State Transitions generative mechanism. (a-d) Sampled state trajectories of length T - 250 (time along horizontal axis) (a) $\alpha = 0.1$, $\beta = 1000$, $\gamma = 100$, visits many states. (b) $\alpha = 0$, $\beta = 0.1$, $\gamma = 100$, retraces multiple trajectory segments. (c) $\alpha = 8$, $\beta = 2$, $\gamma = 2$ visits few states. (d) $\alpha = 1$, $\beta = 1$, $\gamma = 10000$, has strict left-to-right transition dynamics [Beal et al., 2002]

.

from the history of symbols generated by that state. That probability is proportional to the number of times each symbol was emitted in the past m_{iq} [Beal et al., 2002]:

$$P(o_t = q|Y_t = i; m_{iq}, \beta^e) = \frac{m_{iq}}{\Sigma_q m_{iq} + \beta^e} \tag{3.33}$$

Similar to the transition probabilities, each state can transition to an oracle with probability proportional to the parameter β^e. This parameter controls the frequency of each state generating a new symbol [Beal et al., 2002]:

$$P(oracle|Y_t = i; m_{iq}, \beta^e) = \frac{\beta^e}{\Sigma_q m_{iq} + \beta^e} \tag{3.34}$$

With a probability proportional to the times this state has used the oracle (m_q^o), the state can generate a symbol that has been generated in the past, but not from this state [Beal et al., 2002]:

$$P(o_t = q | Y_t = i; m_q^o, \gamma^e) = \frac{m_q^o}{\Sigma_q m_q^o + \gamma^e} \tag{3.35}$$

And with probability proportional to γ^e, it can generate a completely new symbol that has not been generated by this or any other state in the past [Beal et al., 2002]:

$$P(o_t = q | Y_t = i; m_q^o, \gamma^e) = \frac{\gamma^e}{\Sigma_q m_q^o + \gamma^e} \tag{3.36}$$

The parameters β^e and γ^e regulate the frequency of generating new sequences of symbols [Beal et al., 2002].

This model was later derived by Teh, Jordan, Beal and Blei [Teh et al., 2005] in terms of Hierarchical Dirichlet Processes (HDP). The HDP is a set of Dirichlet Processes (DPs) coupled through a shared random base measure (this corresponds to the base distribution from the definition of DPs). This shared base is itself drawn from another DP, and thus the term hierarchical [Teh, 2010]. Specifically, the base distribution of the Dirichlet Process $G \sim DP(\alpha_g, G_o)$ has also a Dirichlet prior such that: $G_o \sim (\gamma_g, H)$. Both these two DPs can be expressed in the following manner:

$$G_{oi}(\phi) = \sum_{j=1}^{\text{inf}} \beta_j \delta_{\phi_j}(\phi) \tag{3.37}$$

$$\forall : \phi_j \sim H \tag{3.38}$$

where $\beta \sim GEM(\gamma)$. This form of expressing DPs is known as the stick-breaking construction for DPs [Van Gael et al., 2008]

$$G_i(\phi) = \sum_{j=1}^{\text{inf}} \pi_{i,j} \delta_{\phi_j}(\phi) \tag{3.39}$$

$$\forall : \phi_j \sim H \tag{3.40}$$

To understand how HDPs are an equivalent characterization of the iHMM, we simply need to identify $\pi_{j,i}$ as the transitions probabilities $P(Y_t = j|Y_{t-1} = i)$ and $\delta_{\phi_j}(\phi)$ as the parameters of the emission distributions for o_t given state $Y_t = j$.

The motivation for introducing a hierarchical DP is that the draws from that Dirichlet prior are unique discrete distributions. In order to introduce coupling across transitions, so we can have common trajectories in the iHMM, a hierarchical Dirichlet prior is needed. In this way, the second DP draws from from a discrete distribution, which itself was drawn from the first DP. Draws from a discrete distribution can have the same value, which leads to the desired coupling.

3.3 Generative Model

In this section, we describe our proposed generative model. The design of this generative model was inspired by the unique properties of the learning algorithms we studied in the previous section. Both the Contextual Episodic and the Hybrid model (which is just a combination of the Contextual Episodic and the Rescorla Wagner) use the information from the environment in a novel way. Here, we describe a generative model of the environment according to which the learning mechanisms of these two algorithms are suitable to make good predictions. First, we give an intuitive explanation of the goals of this generative model, then we formally describe the model. In short, the goal of this generative model is to provide a statistical framework for events that evolve successively, but from time to time jump forward to a completely new event or backwards to an event that happened in the past. Furthermore, each event is paired with a contextual cue. In such an environment, an episodic memory would be necessary to store and recall specific events, and a contextual based recall and weighting would be necessary to select the events with the information that is most relevant for the current decision.

We design an environment that could later be used to in an experimental setting with human subjects. Our environment is a sequence of two-armed bandit task trials, each paired with a distinct picture. These pictures define the context, and are used as contextual cues. Each trial consists of two bandits associated with a reward.

The associated rewards are different on every trial. They vary over time following a Gaussian drift, and represent the slowly changing state of the world. The pictures are unique on each trial. However, any trial can repeat itself later in the sequence of trials. When a trial repeats itself, both a similar picture and similar associated rewards from that trial recur. The reason why this picture and rewards are only similar and not exactly the same is because we assume that the world continues to evolve over time, and thus a repeated trial is never exactly the same. After this repeated trial, the following trials can be the same trials that followed this repeated trial in the past, or completely new trials. In our model, pictures and bandits with their associated rewards are paired. Another important note about our generative model is that even though pictures and bandits with associated rewards can repeat themselves during the sequence of trials, these pictures and bandits are never exactly the same. We assume that pictures and bandit's rewards are changing every time step. The state of the world drifts gradually, but the agent's observations depend on the location of the agent in the world. The agent is able to jump to different states in the world each time steps. It can jump to an unexpected new state and to a previous state. Bornstein's task only probed memories from the past. Figure 3.2 gives an intuition of these characteristics.

To our knowledge, there is no existing model that fulfills the characteristics just described. The main mathematical difficulties for achieving these characteristics are the generative model must be able to generate contextual cues (unique pictures per trial) as well as continuous (in time) bandits' rewards. Furthermore, both these discrete and continuous states must be able to have, on occasions, discrete discontinuous jumps to previous or new states. In order to better understand the required processes of this model, we show a cartoon of this generative model in Figure 3.2. Here, an agent explores different locations in the world, and observes a distinct picture and obtains a reward at each of these locations.

The environment observed by the agent at time *t* depends on the state of the world at time *t* and the location of the agent in the world. The state of the bandits are hidden from the observer until an action is taken and a reward is received or

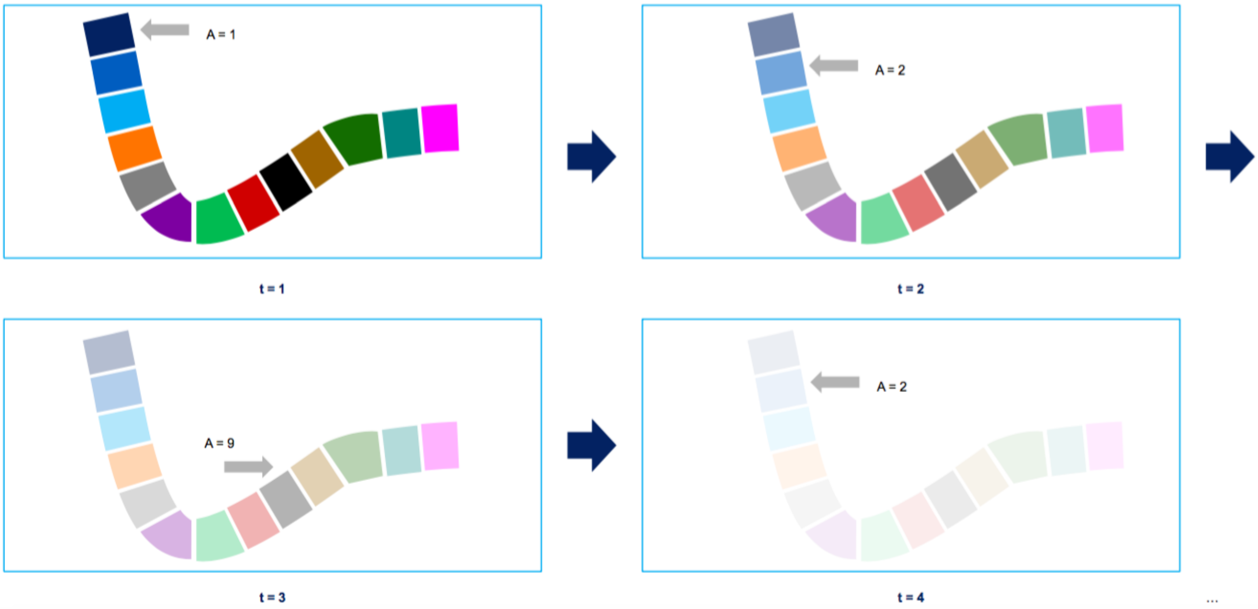

Figure 3.2: Toy model of the Generative model. It describes the temporal dynamics of the state of the world and of the agent in the world. Each square is one time step. At each time step the model of the world changes slightly. The colors at each location, and correspond to different values of rewards. The agent, A, changes locations each time step. The location is indicated by the value that the variable A takes at each time step.

not. Following this analogy, we separate the problem in two: how the world or environment evolves, and the location of the agent in this world. The state of the world consists of a set of pictures and bandits with associated rewards. Both the pictures and the bandits' rewards have temporal dynamics. The agent has its own separate dynamics which determine its location index in the world. The location of the agent determines his or her observations. Furthermore, the dynamics of the agent in the world create the discontinuous ”jumps” that we described earlier. The state of the world (pictures and rewards) drift slowly over time, and it is the agent the one that takes discrete steps in this world. These step can be sequential or the agent can take a step backwards at the beginning of the state of the world or forward to a new state. This process is shown in Figure 3.2.

To model this, we initialize the state of the world as a set of pictures and values of rewards associated with two bandits per trial. The state of the world changes evolves in time following a Gaussian drift. The location of the agent in the world is modeled as a location index Y_t. Initially, the agent is located at location index $Y_{t=1} = 1$, which corresponds to the first picture and first set of bandits of the state of the world. Then, the agent's dynamics are determined by the state transitions of an iHMM. As described above, the iHMM is an Hierarchical Dirichlet process, during which sequential trajectories as well as different type of jumps are possible. In the Experiments section, we show how we use the iHMM to generate different dynamics of the agent.

3.3.1 Formal Definition

To define formally our generative model and show that it satisfies the conditions described above, we define the state of the world (set of pictures and rewards) as $S_t = [Sp_t, Sr_t]$. Both Sp_t and Sr_t are vectors, and they represent the set of pictures and the set of rewards respectively. The initial state of the world is drawn from:

$$P(S_0) = \mathcal{N}(S_o, Q_o) \tag{3.41}$$

where $S_o = [Sp_o, Sr_o]$ are the initial mean values for the set of Pictures and the set of Rewards, and $Q_o = [Q_{o_p}, Q_{o_r}]$ are the covariance matrices of these sets respectively. The state of the world S_t drifts over time following an Ornstein Uhlenbeck process [Finch, 2004]. Without loss of generality, the set of pictures, assigned to all possible locations the agent could visit, are treated as being Gaussian, and at each time step, they are corrupted by Gaussian noise, and the values of the associated rewards also drift following this Gaussian process. At each time step, the state of the world is drawn from:

$$P(S_t|S_{t-1}) = \mathcal{N}(AS_{t-1}, Q) \tag{3.42}$$

where, $A = (A_p, A_r)$, the state transition matrices between states, and $Q = (Q_p, Q_r)$ the covariance matrices.

The location of the agent in the world is denoted by Y_t, a scalar index. This location indicates what picture from the pictures vector Sp_t and what reward from the rewards vector Sr_t are observed by the agent. We define the observed picture and reward received at time t as P_t, and R_t. This means that the state of the world is constantly changing, and observations depend on where in this world the agent is located. Furthermore, because the state of the world is changing over time, even if the agent were to come back to the same location, the state of the world would have drifted and the agent wouldn't observe the exact same picture and reward. The indices for the location of the agent in the world are generated by the sequence of hidden variables from an iHMM model. The indexes are equal to the hidden state sequence of the iHMM. For our model, we do not use the emissions of the iHMM and use only the sequence of hidden states. Every time step the iHMM generates a new hidden state. We use the this hidden state as the location index for our agent. Consequently, we generate the trajectories of Y_t, using equations [4.23 to 4.28].

This process allows the agent to follow common trajectories, but from time to time, also jump to a new state or to a previous state. Once we have generated the state of the world and the location of the agent in the environment, pictures and rewards are generated based on this state and location at time t. Pictures are generated from a probability distribution given by a δ function defined at the entry from the pictures vector Sp_t given by the index location of the agent:

$$P(P_t|S_t,Y_t) = \delta(P_t,Sp_t(Y_t)) \tag{3.43}$$

Rewards are drawn from a Gaussian distribution with a mean equal to the entry from the rewards vector Sr_t given at location of the agent and co-variance σ_r :

$$P(R_t|S_t,Y_t) = \mathcal{N}(Sr_t(Y_t),\sigma_r) \tag{3.44}$$

Using this generative model, we can construct different scenarios with different requirements for learning. Our proposal is that this generative model is able to generate an environment for which the characteristics of the Contextual Episodic or the Hybrid model are necessary. In particular, we want to understand the characteristics of the environment for which a pure Contextual Episodic strategy, or a mixed strategy; such as the Hybrid model, are appropriate. For that, we look at scenarios where each of these algorithms have the highest performance. We also report the performance of the Rescorla Wagner model in our analysis to study the other end of the spectrum of the Hybrid model. In this way, we want to understand how the changing characteristics of the world should lead suitably adaptive agents to go from exhibiting model free to episodic based memory strategies, or to choose a Hybrid strategy - possibly when it is uncertain what the environment requirements are. A second aim of this chapter is to understand the situations when an episodic based reinforcement learning algorithm would be required, and therefore the situations that have so far been under explored by the current reinforcement learning literature.

Below, we describe different scenarios that exemplify different potential environments. To generate these different scenarios, we vary the parameters that describe the dynamics of the agent in this world. The location of the agent and how it explores the world determine the observations of the world it obtains and thus the learning strategy this agent adopts. We assume all agents are at least attempting to maximize rewards. We describe the process of choice and reward maximization in the results section where we explore the performance of agents using different strategies to collect rewards in these different scenarios. Here, we focus on explaining the different scenarios. First, we show the initialization values of the generative model. Afterwards, we describe each scenario individually. The task consists of 300 trials, but Sp_o is initialized to 377 unique unit length pictures (to account for the possibility of further locations - since in theory the environment can have infinite locations), and $Sr_o = (Sr_{o1}, Sr_{o2})$ are initialized such that they are different from each other and at each spatial location as described in Figure 3.2. To achieve this, we draw the initial rewards at each location (here we label each location i) in the following way:

$$Sr_{o1}(i) = sin(\frac{i}{60} + 10) \tag{3.45}$$

$$Sr_{o2}(i) = cos(\frac{i}{60} - 5) \tag{3.46}$$

By drawing the rewards in this manner, we achieve not only that each location has

a different reward, but that the reward at locations far apart will be distinguishable. Thus, if the agent jumps to a further location, the agent will receive a significantly higher or lower reward. Figure 3.16 and Figure 3.4 shows how rewards vary in different locations, and how nearby locations are similar and locations far apart are more different. This design was made to make results clearer, when agents jump or are reminded of non-sequential trials. It is important to remember that this are the initial values of the rewards at each location, and that as the state of the world drifts, so do these rewards.

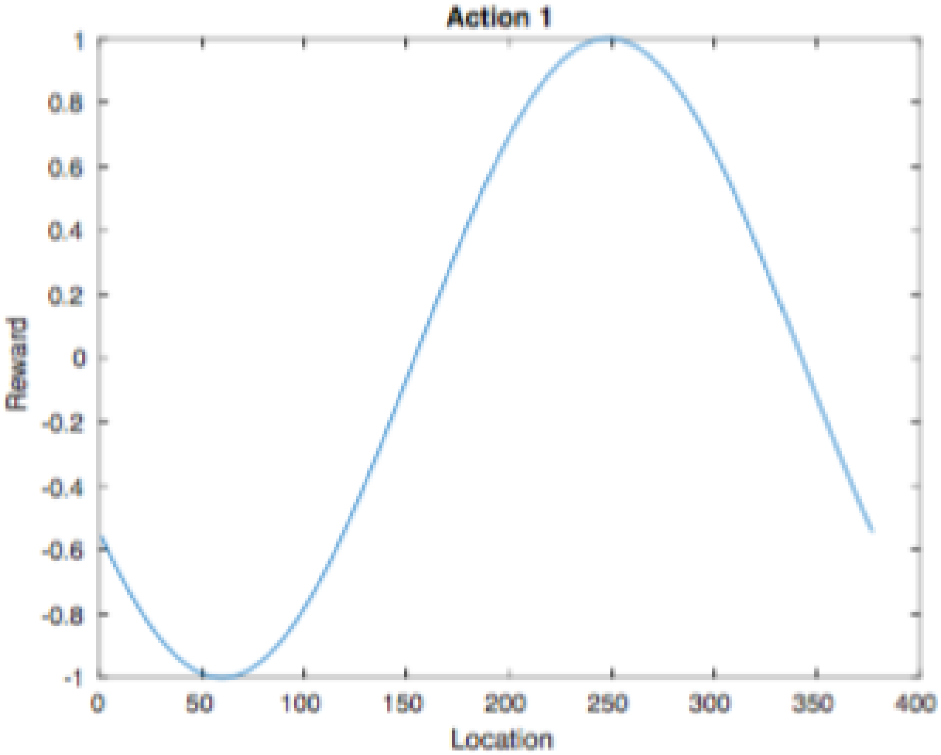

Figure 3.3: Initial values (time t = 0) of rewards given at all different locations in the environment for taking action 1.

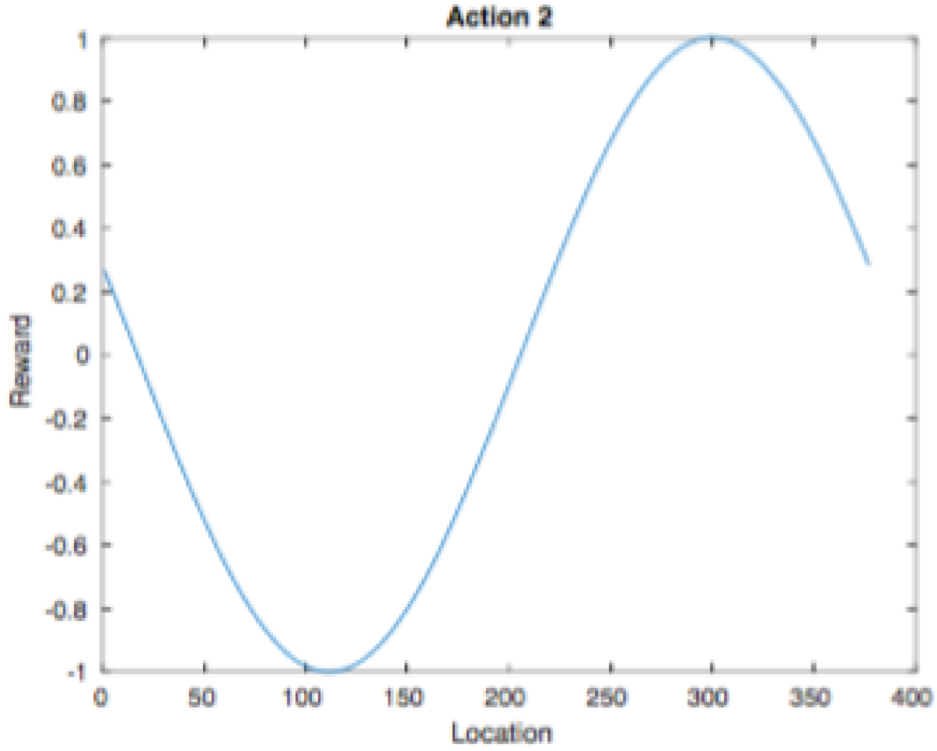

Figure 3.4: Initial values (time t = 0) of rewards given at all different locations in the environment for taking action 2.

3.3.1.1 Scenario 1

In the first scenario, the agent does not change locations. It only observes how the rewards and picture in that location change over time. This scenario is different from the other scenarios because the dynamics of the agent are held constant. The reason for this modification is that we wanted to create a scenario that resembles the generative model of a Rescorla-Wagner algorithm. In this scenario, both the rewards and the picture at the agent's location change on each trial as a Gaussian random walk or corrupted by Gaussian noise respectively. Therefore, the agent observes a slowly changing world with no contextual cues - the picture is the same, just corrupted by noise each time step.

Agent's Dynamics	
$Y_{t=1}$	$\forall t$

Table 3.1: Description of the parameters governing the agent's dynamics for all time steps i.e how the agent explores different locations in the environment at each time step. In this situation, the agent remains in one location.

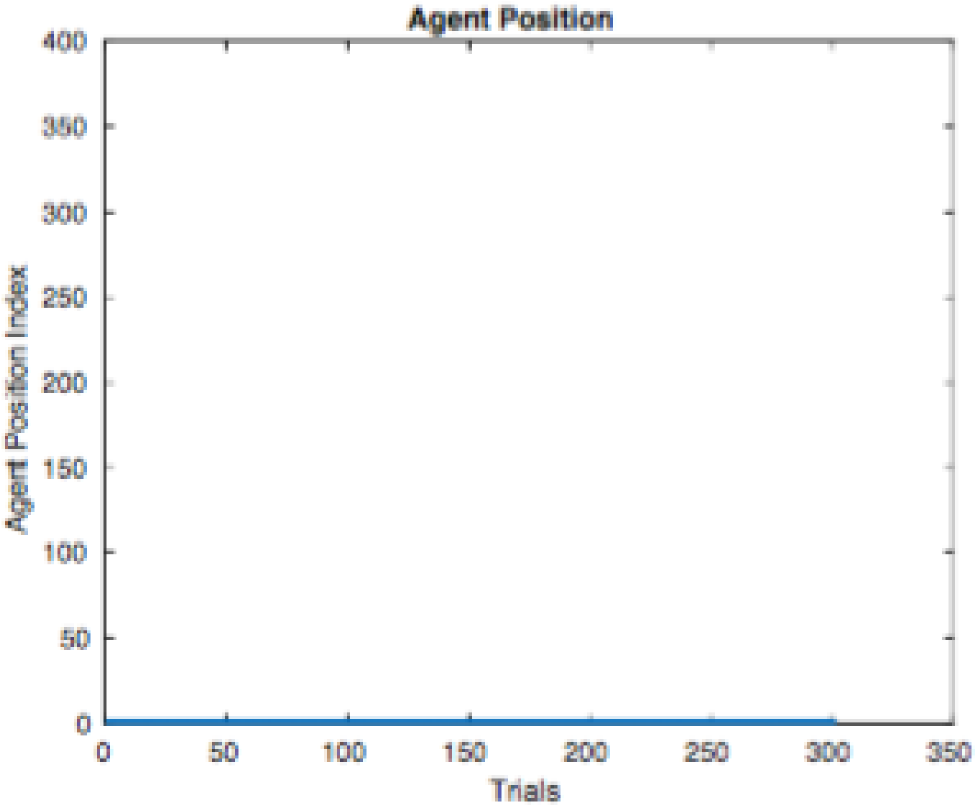

Figure 3.5: Agent's Location in the environment for all time steps. Y axis shows the index of location in the environment, and x axis shows the time step.

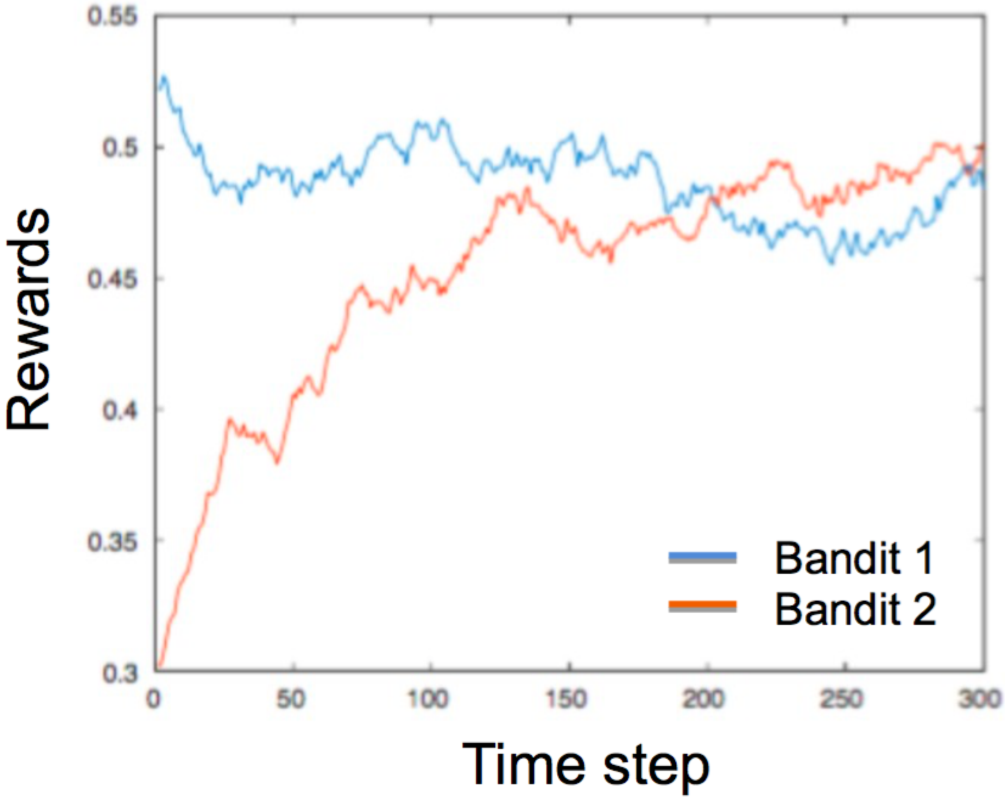

Figure 3.6: Bandits' rewards received by the agent at each time step as it explores different locations in the environment

3.3.1.2 Scenario 2

In this and the subsequent scenarios, we introduce more new or repeated episodes, where we expect an episodic based RL to perform better. In this scenario, the dynamics of the agent are fairly constant. The agent remains most of its time observing one picture (which is slowly changing), and gradually changing rewards. However, from time to time, the agent jumps to a state further along the trajectory in the state of the world and observes a new picture and new associated rewards. After those jumps, the agent always comes back to its original position and continues to observe a slowly changing world with just one picture. To avoid confusion with the parameters from the learning model, we label the hyper-parameters from the generative model with the subscript g.

Agent's Dynamics	
α_g	10
β_g	0.1
γ_g	0.01

Table 3.2: Description of the parameters governing the agent's dynamics for all time steps i.e how the agent explores different locations in the environment at each time step.

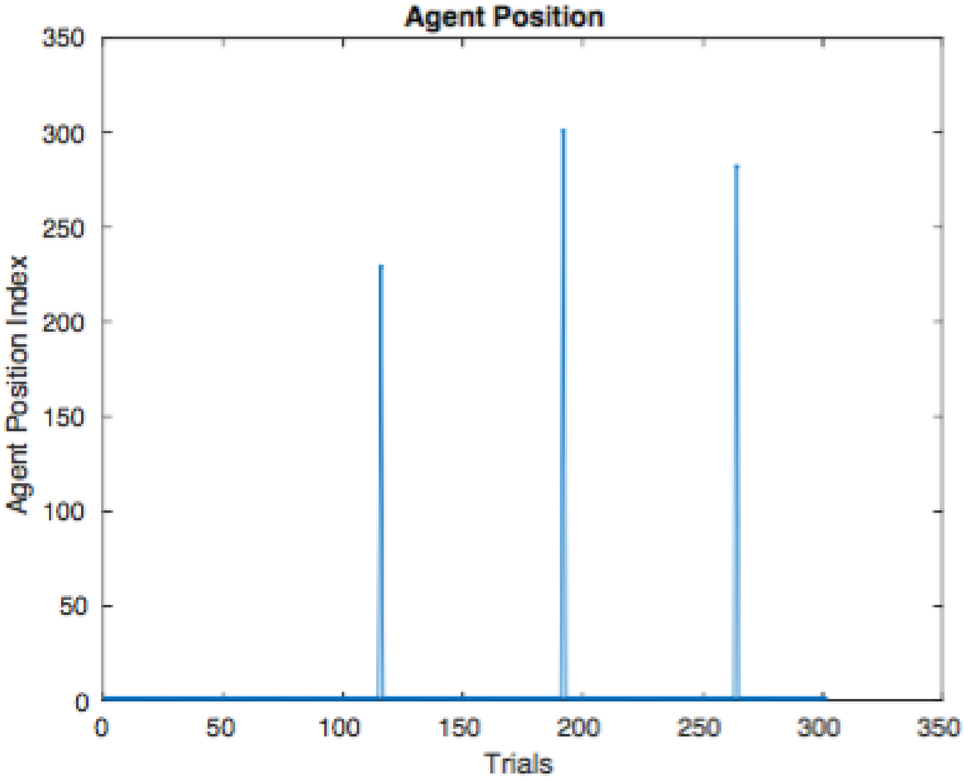

Figure 3.7: Agent's Location in the environment for all time steps. Y axis shows the index of location in the environment, and x axis shows the time step.

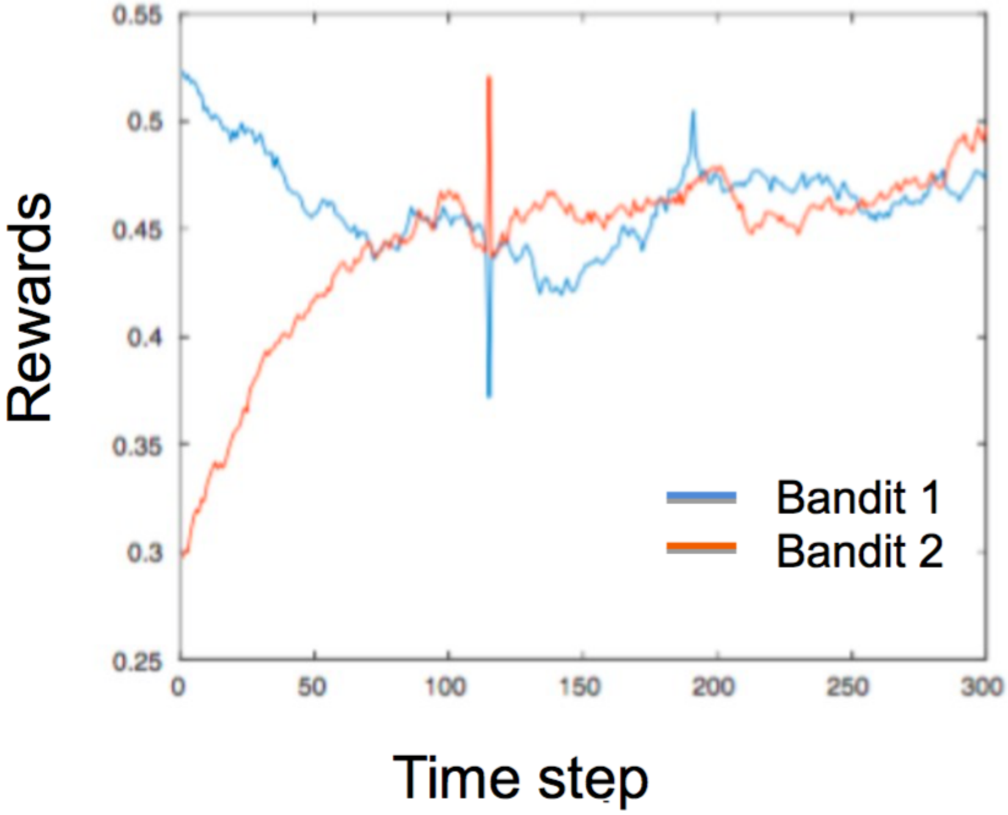

Figure 3.8: Bandits' rewards received by the agent at each time step as it explores different locations in the environment

3.3.1.3 Scenario 3

This scenario consists of the agent exploring the world sequentially. For most of its trajectory, the agent transitions from index i to index i+1. As with the previous scenario, from time to time, the agent jumps to a new state and then it comes back to the same sequential trajectory. These dynamic are seen on the figure below.

Agent's Dynamics	
α_g	0.001
β_g	1000
γ_g	500

Table 3.3: Description of the parameters governing the agent's dynamics for all time steps i.e how the agent explores different locations in the environment at each time step.

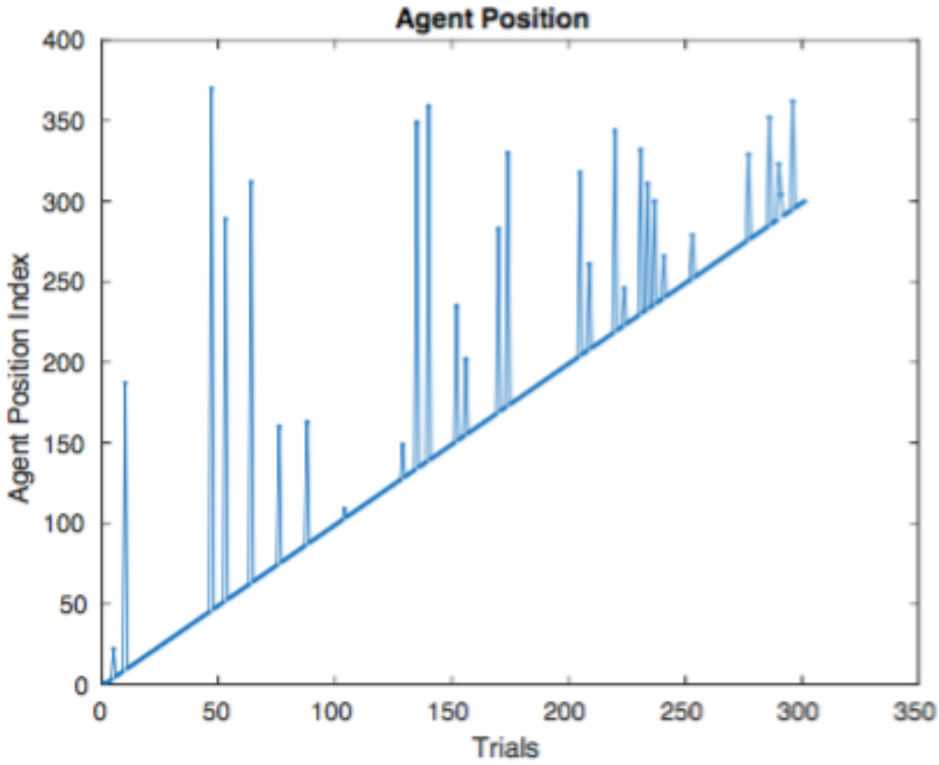

Figure 3.9: Agent's Location in the environment for all time steps. Y axis shows the index of location in the environment, and x axis shows the time step.

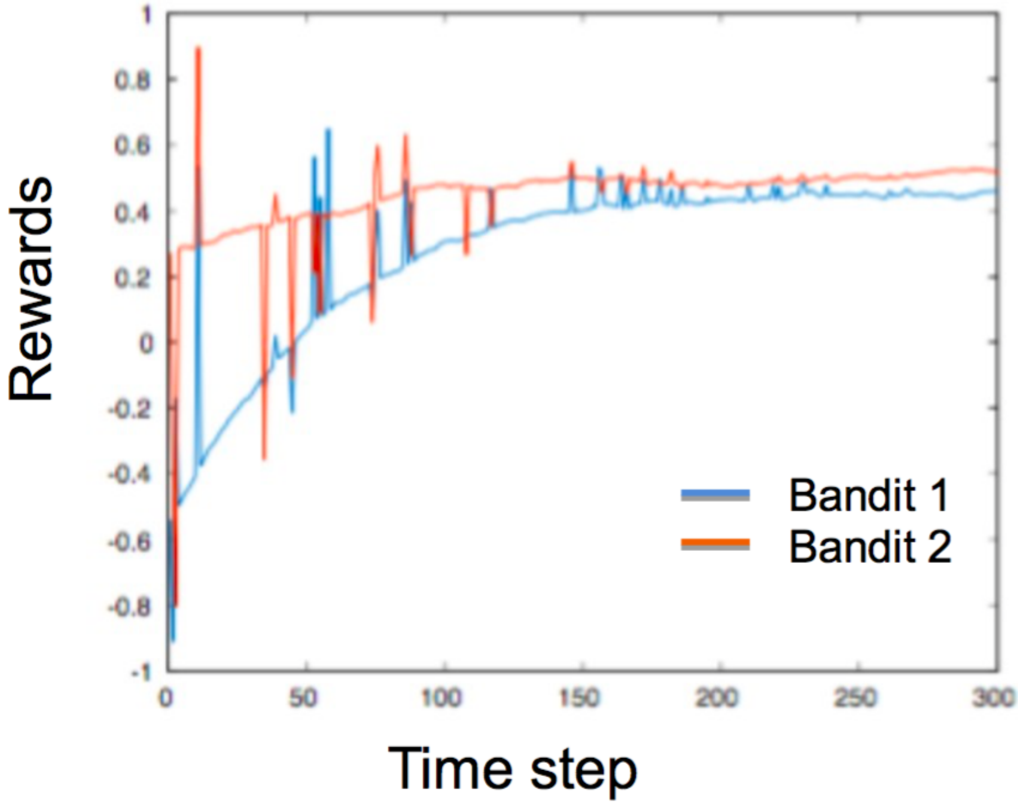

Figure 3.10: Bandits' rewards received by the agent at each time step as it explores different locations in the environment

3.3.1.4 Scenario 4

In this scenario, the agent continues to explore the world sequentially, but it jumps to new and previous states more frequently. In this scenario, the agent will not observe a gradually changing world. We expect that the agent will more heavily rely on its episodic memory to maximize reward.

Agent's Dynamics	
α_g	0.8
β_g	100
γ_g	10000

Table 3.4: Description of the parameters governing the agent's dynamics for all time steps i.e how the agent explores different locations in the environment at each time step.

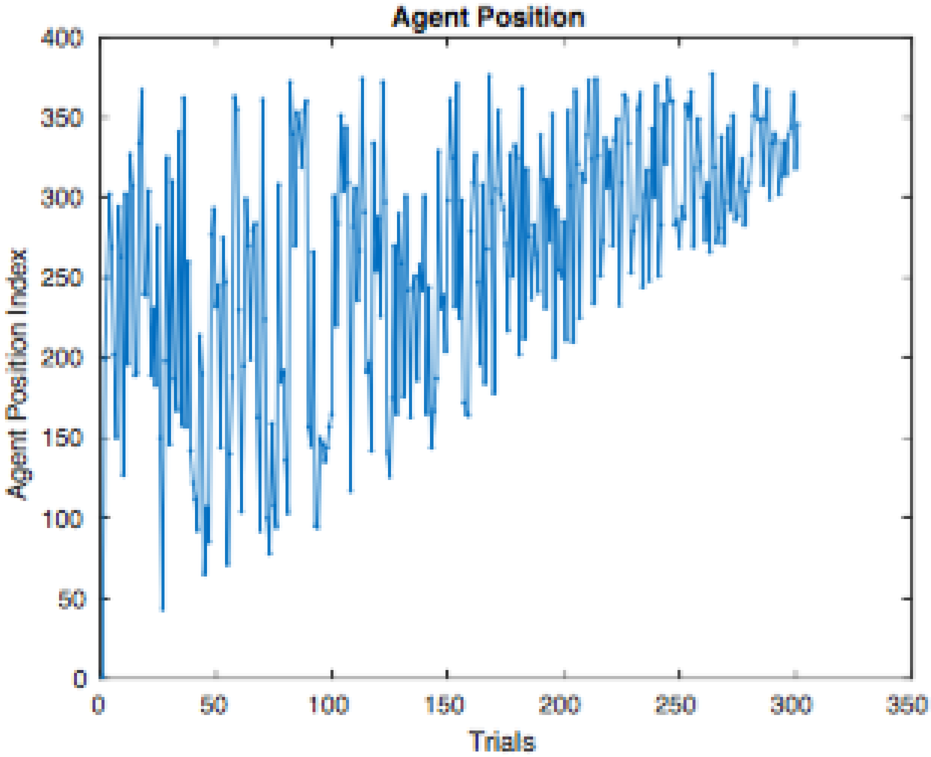

Figure 3.11: Agent's Location in the environment for all time steps. Y axis shows the index of location in the environment, and x axis shows the time step.

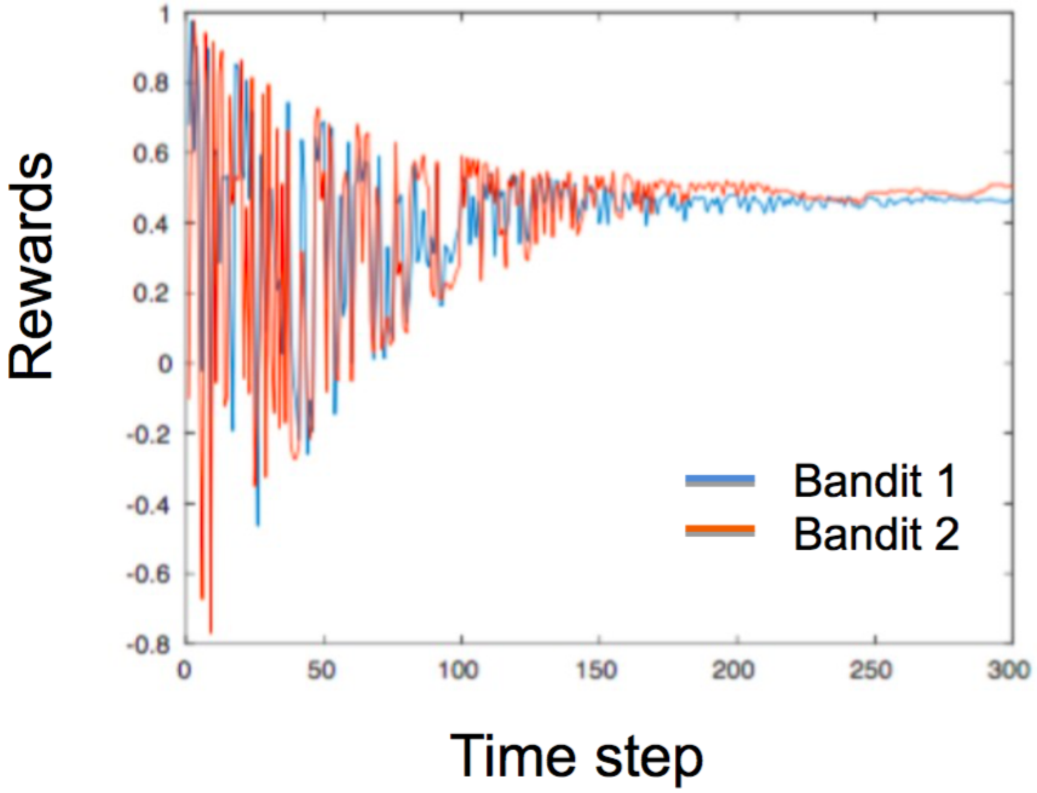

Figure 3.12: Bandits' rewards received by the agent at each time step as it explores different locations in the environment

3.4 Contextual Episodic Model as Inference Approximation

Exact Bayesian inference is a computational framework that defines the optimal performance of agents that learn the structure of the world in order to make decisions or guide actions [Geisler and Diehl, 2003]. We use Bayesian inference as an ideal to which we believe rational agents aspire to. However, capacity constraints, computational costs, and specific task requirements often make optimal inference not achievable. Additionally, agents might have different prior assumptions of the world that make their optimal inference look sub-optimal to outside observers. Exact Bayesian inference might not only be difficult for real world agents to compute, but it is also often times intractable for machine learning algorithms. The reason for this is that many real world problems or data sets require computing intractable integrals in order to estimate the posterior distribution. For these reasons, our goal is to compute exact inference, but when this is not possible, we choose an approximation.

It is the case for our generative model that exact inference is intractable. The reason for this is that it requires computing intractable sums and integrals over all possible states of the agent and all possible states of the world respectively. These integrals and sum grow in size at each time step. Furthermore, solving this equation would require discretizing the continuous variables, which would increase computational complexity even further. It can also be seen that the non-stationarity of the world, as well as the inter-dependency between variables, make inference even more complicated. Finally, inference in iHMMs involves intractable sums over possibly infinite latent state trajectories, which can grow in size each time step. MCMC methods, which are a class of sampling algorithms, can be used to compute intractable posterior integrals. These algorithms have a theoretical guarantee of convergence to the true posterior. However, they suffer from the curse of dimensionality, which in the case of our generative model (due to its increasing complexity at each time step), makes convergence to the true posterior - in a finite amount of time - not feasible. For this reason, in this chapter, we propose an approximate inference algorithm to this exact posterior.

When exact inference is not possible, we can compute an approximation to the exact posterior. One common alternative for approximate inference are variational approximations [Ormerod and Wand, 2010]. These methods are fast and deterministic alternatives to exact inference or sampling methods such as MCMC. They approximate the posterior with a more tractable function. Depending on the nature of the approximation, these methods vary in accuracy [Jordán et al., 1999] [Fox and Roberts, 2012], [Blei et al., 2017]. In our situation, one obvious variational approximation would be that the posterior be computed as two separate processes:

$$P(Y_t, S_t | P_t, R_t) = Q_1(Y_t | P_t, R_t) Q_2(S_t | P_t, R_t) \tag{3.47}$$

This approximation would be highly inaccurate. The problem is that the continuous process (state of the world) provides the vector, and the discrete process (agent dynamics) provides the index that specifies which entry from the state of the world vector is selected at each time step. The index selects what information from the state of the world vectors is used to generate the picture and the reward. At every time step, it is important to know the values of both variables. Both processes are interdependent and coupled. Furthermore, the output of each process depends on its previous time steps, and the previous time steps of the other process. For these reasons, any approximate inference method that separates the continuous and discrete variables would be inaccurate. Since one of the problems with exact inference is that it is difficult to make estimates that mix the continuous and discrete processes, any approximate algorithm would need to separate these processes. In this way, any of these approximations would suffer from the same problems of the approximation just described.

To address this issue, we propose that our Contextual Episodic algorithm is an adequate approximate inference algorithm for this generative model. This algorithm is based on a model of human episodic memory, which is embedded in a Reinforcement Learning framework. First of all, we propose that it is reasonable to construct an approximate inference algorithm based on previous brain-inspired algorithms. The reason for this is that the human brain has found ways to solve intractable computations in real time, and it is only natural to draw inspiration from it. Second, our generative model was constructed to satisfy the requirements postulated as necessary for an episodic-based and reward based decision making algorithm to be necessary. For this reason, it has some embedded characteristics that help it learn and make inferences from this generative model.

Now, we elaborate on the reasons why we propose our Contextual Episodic algorithm as a good approximate inference for our generative model. To do that, first, we need to understand the goal of the agent, and how this agent can use available information to achieve this goal. The goal of the agent is to understand the latent dynamics that give rise to the pictures and rewards it observes/ receives - respectively, and use this information to make predictions about future rewards. Specifically, the agent needs to infer the dynamics governing its own location in the world, the dynamics of the state of the world, and how these two dynamics influence the rewards he or she receives. In our generative model, both the location of the agent in the world and the state of the world itself have their own dynamics. The agent makes the assumption that both the pictures and the rewards are unique to each location each time step. Thus, knowing the location of the agent in the world gives information regarding its associated picture and rewards. Consequently, if the agent observes a picture, and it is able to properly infer its location, it can use its episodic memory to retrieve the associated rewards with that location. Therefore, in our generative model of the world, the inference problem faced by the agent can be narrowed down to inferring the location of the world given the pictures observed. The one caveat to this is that the agent must also take into account that the state of the world is changing. So, the agent can perfectly infer its location from the picture it observes. Both pictures and rewards are gradually drifting over time. Thus, the observations at the current time step are always slightly different from past observations at the same locations.

The Contextual Episodic algorithm can be used to solve this inference and prediction problem. A contextual episodic learning agent, learns a temporal context variable based on pictures observed at each location. The agent uses this context variable to estimate its location in the world, and make predictions about rewards accordingly. The agent uses this context variable to search for similar contexts in the past. The Contextual Episodic agent has a degree of uncertainty regarding which trial is the best to recall and use, due to changing nature of the state of the world. For this reason, it recalls and weights different episodes rather than just one episode. Due to the continuity of the world, the agent expects to weight heavily the most recent episodes. However, it also knows that from time to time it can be in a different location with no resemblance to the recent past. To reconcile these two assumptions, the agent weights all retrieved episodes based on their contextual similarity. In this way, recent episodes are still weighted heavily due to the low pass filter nature of the context variable - the current context is similar to its recent past, but also episodes that occurred far in the past but are relevant in the present (similar in context) are also included. The agent also retrieves and weights all rewards associated with these relevant episodes. It makes a prediction about the reward at a new location by integrating all relevant previous rewards, and weighting them by the contextual similarity of their associated episodes. In short, a contextual episodic learning agent uses observed pictures to infer its location, which is hidden to the agent, in the form of a context variable. Then, it uses stored episodic memories (stored rewards with associated context) to make predictions of future rewards.

In order to see how the Contextual Episodic model operates, we designed a simple task with one memory probe, and used it to show the effect of that probe on the weighting function and the Q values' estimates. We compared the weighting function and the Q values' estimates with the standard Rescorla Wagner model. Our goal is to show how the Contextual Episodic model uses information from the cue to weight previous episodes accordingly, and how this has different effects on the Q values compared to the Rescorla Wagner's estimates. The task consisted of 20 trials and one memory probe of trial 3 at trial 17. Rewards were between 0 and 1 and they were drawn from a normal distribution with mean equal to 0.3 (Keeping only positive numbers, and re-sampling if the number drawn was negative). In order to make the visualization of the probe more evident, we changed the rewards given at trial 3, 4, 17 and 18 in the following way: at trial 3 and 17 the rewards given for action 2 were equal to 0.05 (far lower than the average reward for all other trials), and the reward given for action 1 at trial 4 and 18 were equal to 0.95 and 0.65 respectively (far greater than the average for all other trials). Our goal with this task was to see clearly the effect of remembering trial 3 at trial 17, and to test whether trial 4 also had an effect on trial 18. We designed the task such that it would be useful for an agent to remember this information, and made the rewards at trials 17 and 18 the same as the rewards at trial 3 and 4. The rewards for all trials can be seen in Figure 3.13 and Figure 3.14.

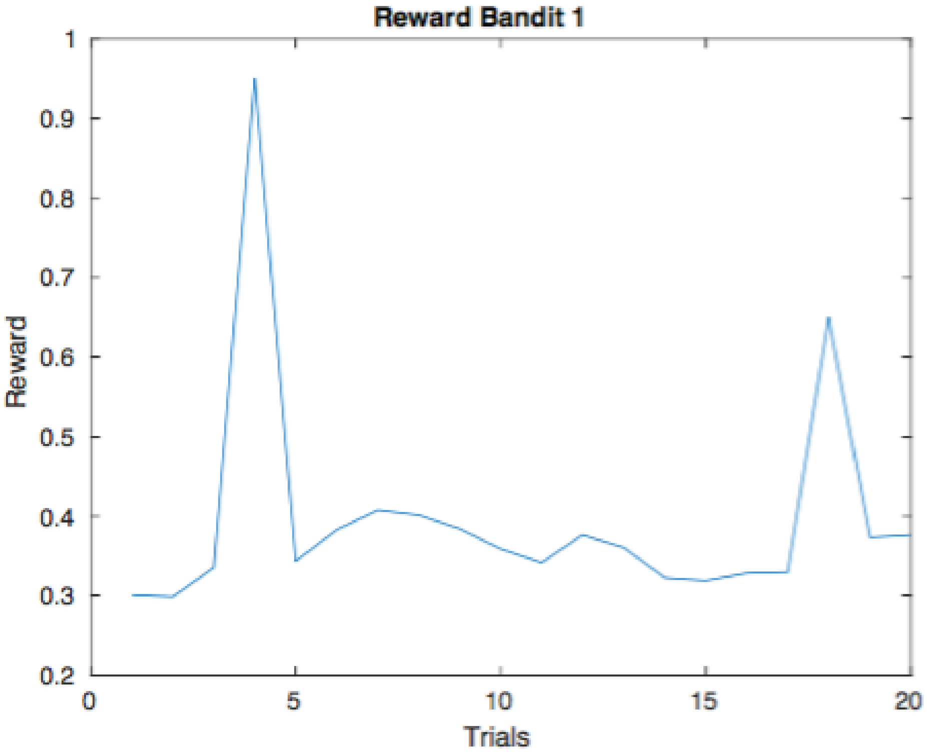

Figure 3.13: Simple Task. Rewards for Action 1 for all trials

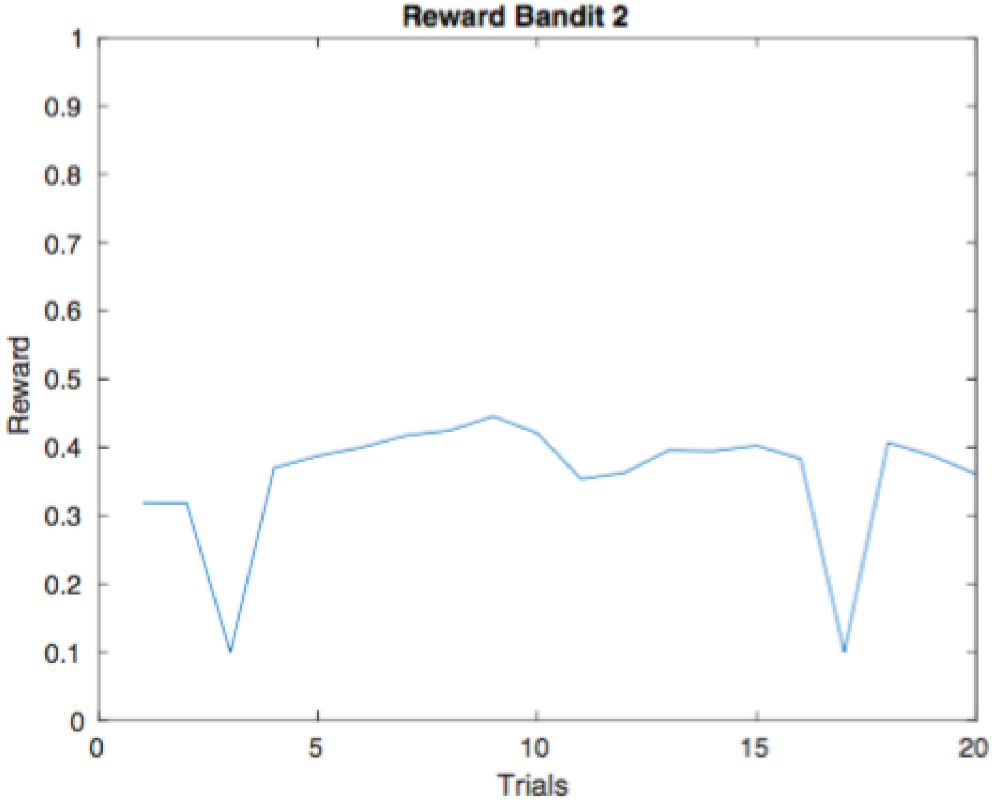

Figure 3.14: Simple Task. Rewards for Action 2 for all trials

Now, we show the effects of the memory probe at trials 17 and 18 on the weighting of previous episodes by the Contextual Episodic model and the Rescorla Wagner model. We chose these two trials, because we expect to see the effects of the memory probe at these trials. The memory probe of trial 3 was presented at trial 17. Thus, if the learning agent was using the probe as information, this effect would be seen in the weighting function at this trial. Furthermore, if the learning agent was using a temporal model, then the effect of this probe would also be seen in the following trial. We show the weighting functions for action 2 (at trial 17) and action 1 (at trial 18), because the learning agent selected these actions at trials 3 and 4. Consequently, the information of these trials will only be present in the weighting functions of these actions at trial 17 and 18 respectively. We only update the weight for observed trials.

In Figure 3.15, we show the difference in weighting of the probed event (trial 3) and most recent trial (trial 16) at trial 17 (action 2), and of the event following the probe (trial 4) and the most recent event (trial 17) at trial 18 (action 1) between the Contextual Episodic and RW models. It can be seen that the Rescorla Wagner puts most of the weight on the most recent trial. This effect corresponds with the fact that the RW is a recency based model and does not incorporate the information from memory probes. On the other hand, the Contextual Episodic model takes into account the information from the memory probe and weights this trial the most. It can also be seen that this model uses temporal information, and at trial 18, it puts some weight trial 4.

In the following figure, we show how these differences in weighting functions impact the predictions made by each model. More specifically, we show the differences in Q values computed by each model at trial 17 and 18. We show that the Rescorla Wagner's predictions are less accurate than those of the Contextual Episodic model. Because it doesnt use the information from the probes, at trial 17 it overvalues action 2 and at trial 18 it undervalues action 1. It is important to note that the Contextual Episodic predicts values closer to the actual rewards; however, there is still some difference between the Q value and the actual reward due to the influ-

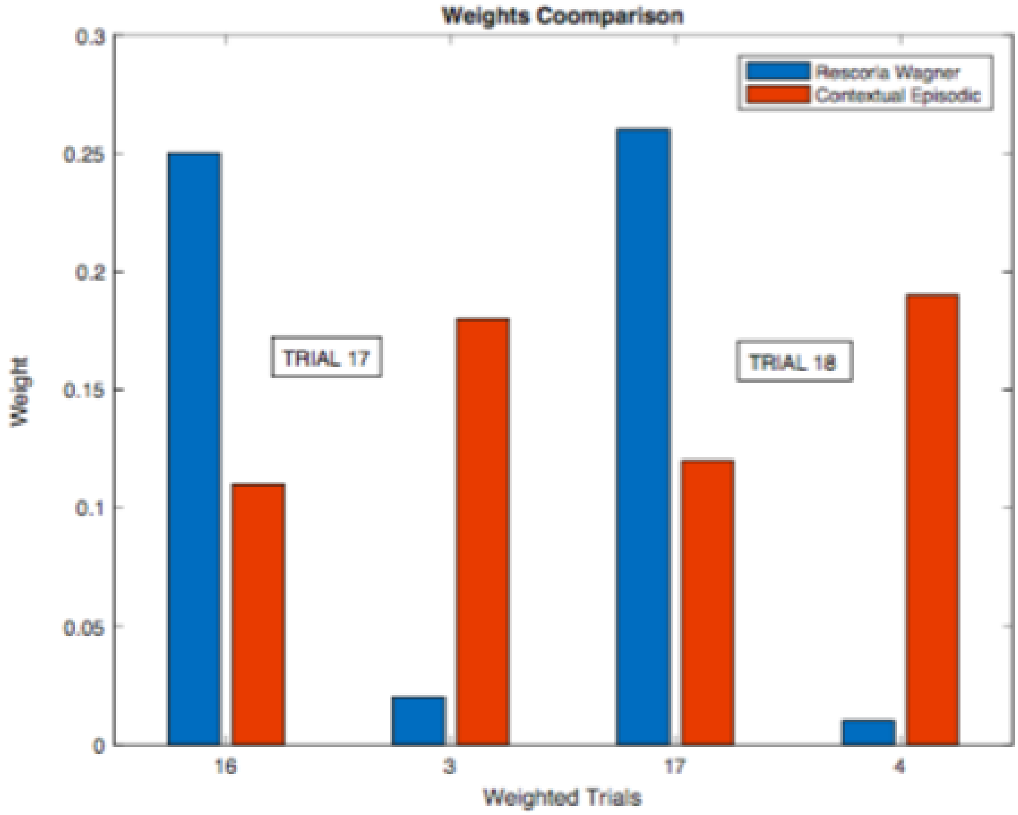

Figure 3.15: Comparison of the weight assigned by the Rescorla Wagner and Contextual Episodic models to the most recent trial, the probed trial and the trial following the probed trial at the trial when the probed was shown (trial 17) and at the following trial (trial 18)

ence of previous trials in the computation of the Q value. Even if the Contextual Episodic model weights the probed trial the most, it still considers the information from the most recent experiences. In our situation, this extra consideration diminishes the accuracy of the prediction.

In conclusion, this figure shows how the Contextual Episodic model uses the information of the probe to make predictions, and how this information is used based on contextual similarity (obtained through the weighting functions shown above). In an environment where context gives information about rewards, the Contextual Episodic model is a more suitable model for inference and prediction than the Rescorla Wagner model. The difference in predictions made by these two models can make a big difference in rewards obtained by an agent; particularly, when repeated trials have unusual rewards. For humans, this is analogous to our

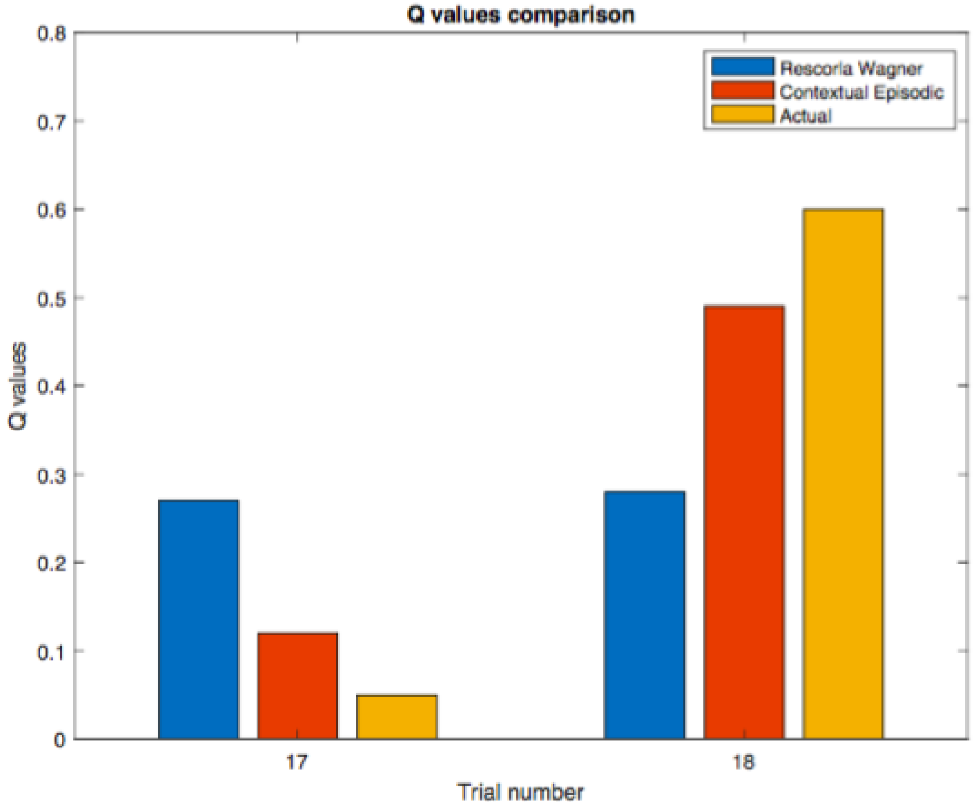

Figure 3.16: Q values predictions at the trial when the probe was shown (trial 17) and at the following trial (18) by the Rescorla Wagner and Contextual Episodic models. Next to these values, we show, for comparison, the value of the actual reward received at these trials (called Actual in the label). Neither predictions is perfect due to the effect of other trials in the computation, but the Contextual Episodic model computes a better estimate of the actual reward than the Rescorla Wagner model.

memories of unusual events, which help us avoid traps or seek opportunities.

This example shows the Contextual Episodic model at work, how it differentiates from the Rescorla Wagner model, and how its weighting strategy allows it to make more accurate predictions. In the next section, we design experiments where we show the scenarios where the weighting functions of the different models (Contextual Episodic and Rescorla Wagner) are useful. We also show when the Hybrid strategy turns out to be the best strategy.

3.5 Experimental Results

Here we test the performance of the Contextual Episodic, the Hybrid and the Rescorla-Wagner models in the four scenarios described previously. Our goal is to understand which of these models would perform the best in different situations, and with this understand when and why a learning agent would use each of these strategies. Particularly, we are interested in discovering the conditions of the world for which the Hybrid model would have the highest performance. In the previous chapter, we saw that the Hybrid model was the algorithm that fit human data the best. For that reason, we can say that from all algorithms that we have discussed so far, the Hybrid algorithm resembles best the assumptions that humans make to guide their decisions. By understanding the characteristics of the environment for which the Hybrid model is the most adequate algorithm, we can gain some insights into the assumptions of the world that humans have.

In order to achieve these goals, we trained our algorithms to maximize cumulative reward using the pictures generated by our generative model as contextual cues, and the rewards as the outcomes from two bandits. Because the generative model is stochastic for each given set of parameters, we use Bayesian optimization to compute the best fit parameters. The learning agent chooses one bandit and receives a reward. Over time, this agent will learn the value of each action and make choices accordingly. How the agent learns these values is the topic of our analysis. The three models we have described are the strategies that we test in this section. The equations for computing these values using each of this algorithms were described in the previous chapter. For reference, see Chapter 3 equation 3.9 for the Rescorla Wagner model, equation 3.30 for the Contextual Episodic model and equation 3.32 for the Hybrid model. For each scenario, we trained each algorithm over 100 tasks (each tasks consisted of 300 trials or time steps), and optimized the parameters using Bayesian optimization. Below we show the optimized parameters as well as the accumulated rewards by each algorithm on each scenario.

3.5.1 Scenario 1

The dynamics of the agent in this scenario are described by Figure 3.5. The optimized parameters for this scenario are the following:

Model	Parameters
Rescorla Wagner	$\alpha = 0.26(0.12)$ $\beta = 6.2(1.01)$
Hybrid	$\alpha = 0.33(0.08)$ $\beta = 5.7(1.2)$ $\gamma = 0.29(0.02)$ $w = 0.67(0.25)$
Contextual Episodic	$\gamma = 0.37(0.18)$ $\beta = 7.8(1.58)$

Table 3.5: Learned parameters for data from Scenario 1 using the Rescorla Wagner, Hybrid and Contextual Episodic Models

Using these parameters, the Rescorla-Wagner algorithm had the highest performance. It was able to accumulate the highest reward, because it keeps a running average of received rewards weighted by recency, and uses this average to make predictions about the next reward. In an environment in which rewards are changing slowly, this strategy is optimal. The Hybrid model is the second best algorithm, and the parameter w that maximizes the performance of this algorithm in this scenario is close to 1. The closer the w is to 1, the higher the contribution of the Rescorla-Wagner algorithm. For this scenario, this value of w makes sense, since the agent is trying to maximize performance.

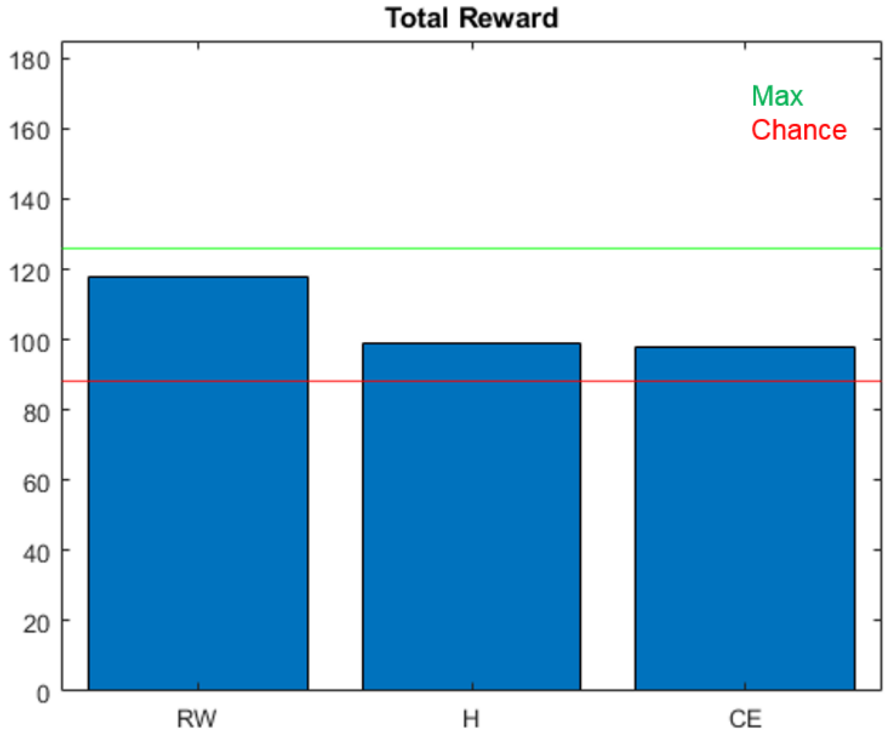

Figure 3.17: Total rewards collected by the Rescorla Wagner, Hybrid and Contextual Episodic models using data from Scenario 1. Each bar shows the mean value of total rewards collected over 100 tasks (each tasks consisted of 300 trials or time steps). The red line (Chance = 88.2) shows the total reward that collected by an agent that selects between actions with equal probability. The green line (Max = 126.5) shows the maximum total reward collected by an agent that knows the action that gives the highest reward at each time step. There was a statistically significant difference between the total rewards collected by each model over the 100 tasks as determined by one-way ANOVA (F(4,495) = 27, p <0.001). A Tukey post hoc test revealed that the difference between the total rewards collected between the Rescorla Wagner model (118 +/- 4, p <0.001), the Hybrid model (99 +/- 2, p <0.001) and the Contextual Episodic model (98 +/- 5, p <0.001) were statistically significant.

3.5.2 Scenario 2

The dynamics of the agent in this scenario are described by Figure 3.7.The optimized parameters for this scenario are the following:

Model	**Parameters**
Rescorla Wagner	$\alpha = 0.48(0.28)$ $\beta = 6.55(0.34)$
Hybrid	$\alpha = 0.41(0.05)$ $\beta = 4.02(1.75)$ $\gamma = 0.57(0.86)$ $w = 0.68(0.16)$
Contextual Episodic	$\gamma = 0.69(0.75)$ $\beta = 5.1(0.92)$

Table 3.6: Learned parameters for data from Scenario 2 using the Rescorla Wagner, Hybrid and Contextual Episodic Models

In this scenario, both the Rescorla-Wagner and the Hybrid model performed better than the Contextual Episodic model. The Hybrid model performed slightly better than the Rescolar-Wagner model. The parameter w that weights the contribution of each model in the Hybrid model was fit to $w = 0.68$, which shows a major contribution from the Rescolar-Wagner model with some contribution from the Contextual Episodic model. This result makes sense in the case of Scenario 2, where the agent makes few discrete jumps, but it mainly remains in its original position. In this position, the agent observes how the world changes gradually, while when the agent makes discrete jumps, it observes different discrete states of the world. At the beginning of the trial, these jumps are completely unfamiliar to the learning agent, but as more discrete jumps happen, the agent can integrate information from previous jumps to make better informed decisions. For these later jumps, the contribution of episodes become more useful. As we will see in the following scenarios, as jumps become more frequent, the performance of the Contextual Episodic model increases.

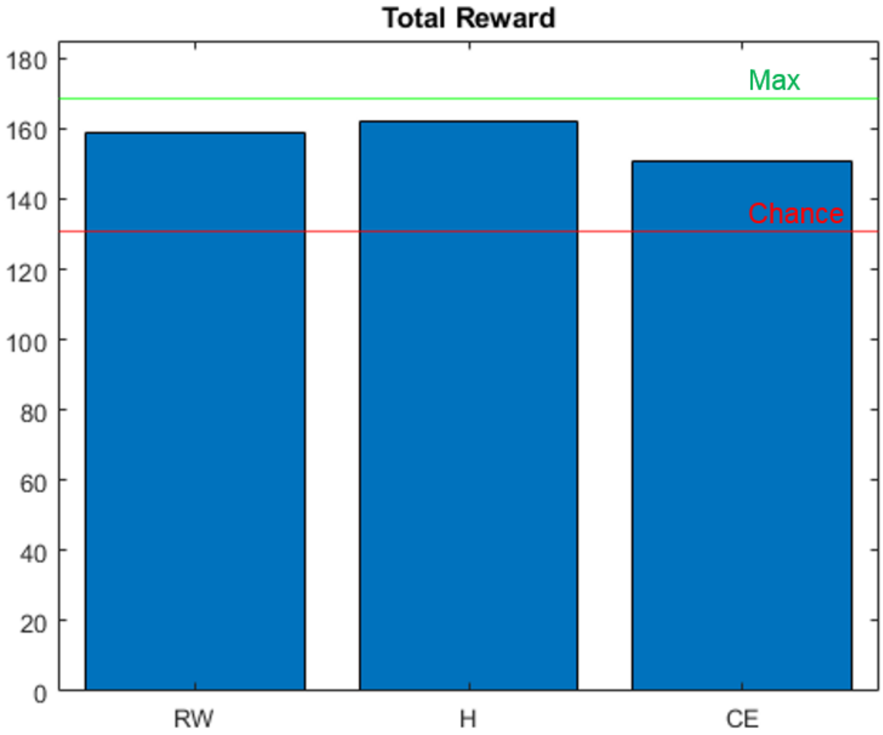

Figure 3.18: Total rewards collected by the Rescorla Wagner, Hybrid and Contextual Episodic models using data from Scenario 2. Each bar shows the mean value of total rewards collected over 100 tasks (each tasks consisted of 300 trials or time steps). The red line (Chance = 131.2) shows the total reward that collected by an agent that selects between actions with equal probability. The green line (Max = 168.5) shows the maximum total reward collected by an agent that knows the action that gives the highest reward at each time step. There was a statistically significant difference between the total rewards collected by each model over the 100 tasks as determined by one-way ANOVA (F(4,495) = 35, p <0.001). A Tukey post hoc test revealed that the difference between the total rewards collected between the Rescorla Wagner model (159 +/- 4, p <0.001), the Hybrid model (162 +/- 4.2, p <0.001) and the Contextual Episodic model (151 +/- 3.8, p <0.001) were statistically significant.

3.5.3 Scenario 3

The dynamics of the agent in this scenario are described by Figure 3.9.The optimized parameters for this scenario are the following:

Model	**Parameters**
Rescorla Wagner	$\alpha = 0.41(0.15)$ $\beta = 8.02(0.66)$
Hybrid	$\alpha = 0.41(0.32)$ $\beta = 9.2(1.07)$ $\gamma = 0.63(0.21)$ $w = 0.45(0.18)$
Contextual Episodic	$\gamma = 0.71(0.65)$ $\beta = 8.2(0.93)$

Table 3.7: Learned parameters for data from Scenario 3 using the Rescorla Wagner, Hybrid and Contextual Episodic Models

In this scenario, the Hybrid model performed best. The parameter w was fit to $w = 0.45$. This result shows that both models had a significant contribution to the performance of the Hybrid model. In this scenario, the agent moves in the environment continuously with the exception of some discrete jumps. The number of jumps are higher in this scenario than in the previous scenarios, which explain the higher contribution of the Contextual Episodic model. However, the Rescorla Wagner model is also able to perform well, because there is some gradual continuity in the way the agent explores the world, and therefore in the values of the rewards received.

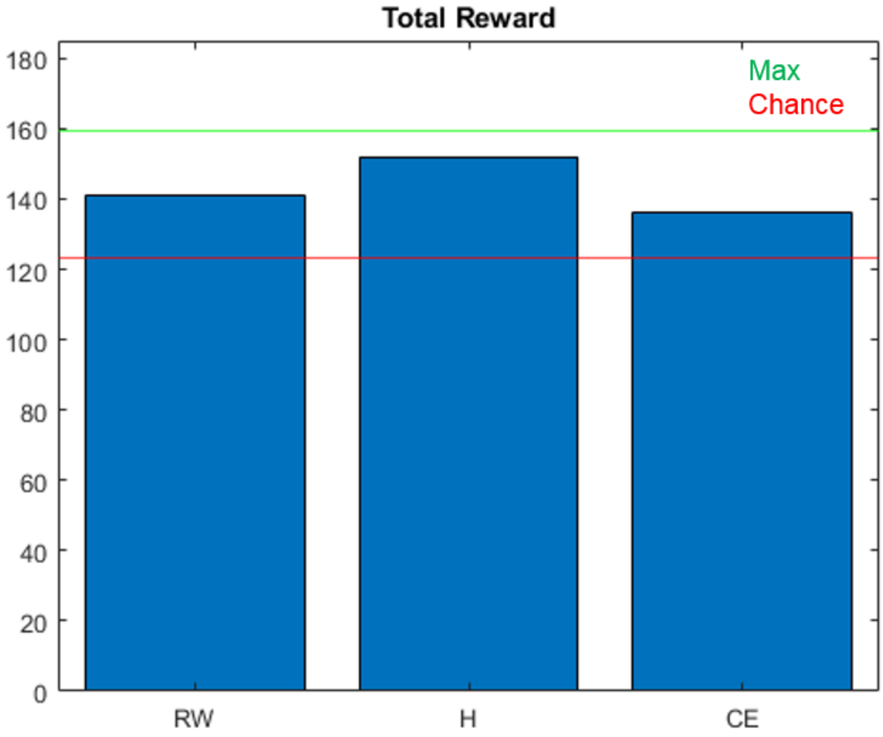

Figure 3.19: Total rewards collected by the Rescorla Wagner, Hybrid and Contextual Episodic models using data from Scenario 3. Each bar shows the mean value of total rewards collected over 100 tasks (each tasks consisted of 300 trials or time steps). The red line (Chance =123.48) shows the total reward that collected by an agent that selects between actions with equal probability. The green line (Max = 159.25) shows the maximum total reward collected by an agent that knows the action that gives the highest reward at each time step. There was a statistically significant difference between the total rewards collected by each model over the 100 tasks as determined by one-way ANOVA (F(4,495) = 29, p <0.001). A Tukey post hoc test revealed that the difference between the total rewards collected between the Rescorla Wagner model (141 +/- 3.2, p <0.001), the Hybrid model (152 +/- 4.9, p <0.001) and the Contextual Episodic model (136 +/- 4, p <0.001) were statistically significant.

3.5.4 Scenario 4

The dynamics of the agent in this scenario are described by Figure 3.11.The optimized parameters for this scenario are the following:

Model	**Parameters**
Rescorla Wagner	$\alpha = 0.71(0.56)$ $\beta = 7.56(0.83)$
Hybrid	$\alpha = 0.49(0.39)$ $\beta = 7.8(1.02)$ $\gamma = 0.81(0.12)$ $w = 0.15(0.17)$
Contextual Episodic	$\gamma = 0.67(0.52)$ $\beta = 8.8(1.12)$

Table 3.8: Learned parameters for data from Scenario 4 using the Rescorla Wagner, Hybrid and Contextual Episodic Models

In this final scenario, the Contextual Episodic performs best. This scenario shows that when events are discontinuous, and recent events do not hold the most relevant information, the learning agent can turn to its episodic memory to make accurate decisions. Episodes are recalled based on contextual similarity, and are weighted accordingly at the time of decision. This flexibility allows the Contextual Episodic model to perform best in scenarios, where recent events provide with little information. This is the case for scenario 4, where there are a lot of discontinuous jumps. One important thing to note is that this discontinuous jumps, as seen in Figure 3.11, often have similarities to previous jumps earlier in the trial.

The goal of this chapter was to introduce a generative model for an episodic-based Reinforcement Learning algorithm. In particular, for the Contextual Episodic and the Hybrid models, and to show the characteristics of the world for which either model would be the most suitable. We have shown that this generative model can produce a spectrum of scenarios, where a purely Rescorla Wagner model or a Contextual Episodic model are the best choices. But more importantly, we showed that the generative model can generate scenarios, where a mixed strategy, like the Hybrid model, is necessary. It was our goal to show a model of the world for the Hybrid strategy, which was the one that fit human data the best in the previous chap-

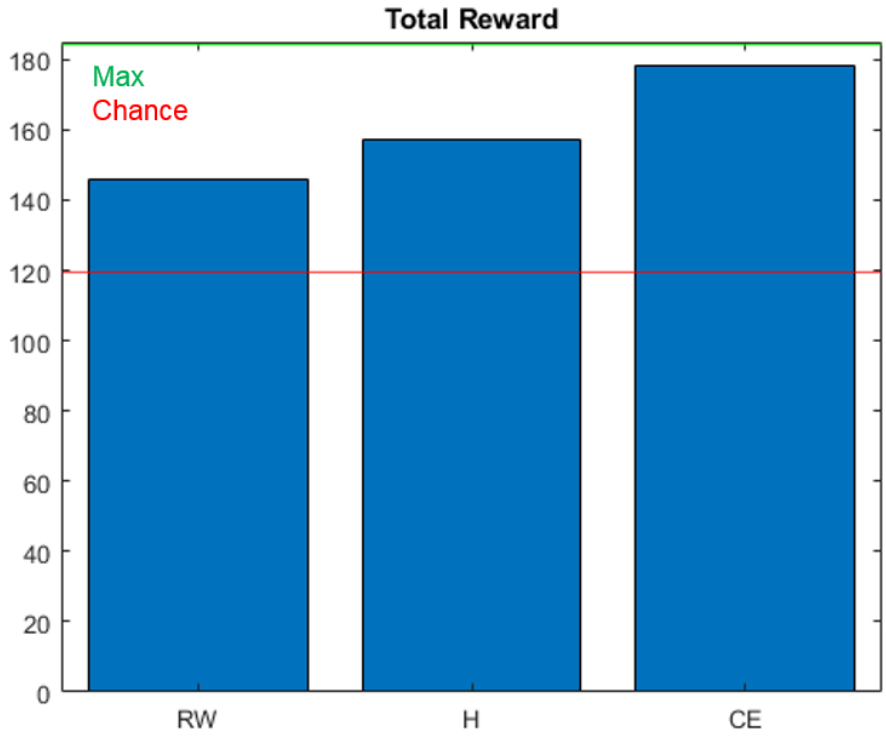

Figure 3.20: Total rewards collected by the Rescorla Wagner, Hybrid and Contextual Episodic models using data from Scenario 2. Each bar shows the mean value of total rewards collected over 100 tasks (each tasks consisted of 300 trials or time steps). The red line (Chance = 119.5) shows the total reward that collected by an agent that selects between actions with equal probability. The green line (Max = 184.4) shows the maximum total reward collected by an agent that knows the action that gives the highest reward at each time step. There was a statistically significant difference between the total rewards collected by each model over the 100 tasks as determined by one-way ANOVA (F(4,495) = 31, p <0.001). A Tukey post hoc test revealed that the difference between the total rewards collected between the Rescorla Wagner model (146 +/- 2.7, p <0.001), the Hybrid model (157 +/- 3, p <0.001) and the Contextual Episodic model (184.43 +/- 4.8 , p <0.001) were statistically significant.

ter. In this way, we wanted to understand the constraints and assumptions about the world made by humans when making decisions.

It is important to note that, presumably with more data, the Hybrid model would have been able to fit the data in all scenarios at least as well as the Contextual Episodic (in Scenario 1), and the Rescorla Wagner model (in Scenario 2), by setting the value of w to 0 and 1 respectively. This implies that the Hybrid model would have been able to learn that the Hybrid strategy is not optimal, and put all weight in one strategy.

We have also shown that our algorithms can perform adequate approximate inference in our generative model. It would be an interesting question for future work to understand how closely our algorithms approximate exact inference.

3.6 Conclusions

Generative models define statistical hypotheses regarding the generation of a particular data set. They define the probability distribution of observing data given some underlying, latent (hidden) causes. The counterparts to these models are their statistical inverses known as recognition models, which are used to infer the probability distribution over these underlying latent causes, for which they are also known as inference models. In this chapter, we propose a generative model for which the Contextual Episodic or Hybrid models (described and justified in chapter 3, partly via their link to the TCM [Howard and Kahana, 2002] and kernel based similarity, which [Jäkel et al., 2009] [Shahbazi et al., 2016]) are reasonable approximate inference algorithms, or recognition models.

The generative model proposed captures relevant characteristics for which an episodic memory would be necessary and therefore provides a formal test ground to explore the advantages of endowing RL algorithms with a memory of this sort. The main characteristics of this generative model parallel characteristics of the real environment that are often ignored in laboratory experimental settings. For example, with this generative model, random new events can happen unexpectedly, and circumstance similar to those from past events may reoccur. These two characteristics are quite common in day to day life, but have not been extensively tested in laboratory scenarios for decision-making, or substantially implemented in computational models.

Bayes-optimal inference in this generative model is intractable. As a consequence, this might not be the most efficient goal for adequate probabilistic reasoning [Dasgupta et al., 2017], [Gershman et al., 2015], [Lieder and Griffiths, 2019]. Instead, when considering the costs of computations and complexity of the real world, humans might use a variety of approximations or heuristics. Unfortunately, various conventional methods that converge or approximate the exact posterior appear to be unfeasible. In particular, MCMC methods, often a gold standard for accuracy, would be very computationally expensive and likely struggle to sample new states in the agent's trajectory. We justified the Contextual Episodic algorithm as a suitable approximate inference algorithm by showing how a learner that uses this algorithm can learn to infer the state of the world and make good predictions about future reward.

We then explored different parameter settings within this generative model to examine when a learner could usefully employ an episodic memory based strategy. For simplicity, we focused on the case of immediate rewards without considering the additional complexities that arise with prospective planning. We studied the transition between a purely model-free strategy using the Rescorla Wagner algorithm and a Contextual Episodic strategy, with the Hybrid model used a mixed strategy for problems where the agent has not gathered enough information or it is not clear whether to use purely cached values or episodes. Particularly, we were interested in understanding the circumstances under which a model that includes an episodic memory and has higher complexity, such as the Contextual Episodic or the Hybrid models, would be required for better performance–measured in accumulated rewards by the learning agent. In order to make better predictions, the inference model used by the learning agents needs to fit the characteristics of the generative model. This inferential complexity was rewarded when the environment is equally complex, as shown by the different scenarios where an agent would choose either of the strategies described. This work sets the stage for future studies to address meta-learning considerations that arise in this framework, such as learning parameter w, or learning when and how to select the strategies discussed in this chapter. In our experiments, we kept parameter w fixed, but it is likely that in a closer approximation of human behaviour, such a parameter would be dynamic and adapt as the agent collects more information.

One of the benefits of having a generative model is that it suggests new experimental directions, which can offer a more stringent examination of the mechanisms of storage and recall of episodes during decision-making. In particular, Bornstein's experiment 2 used contextual cues that did not represent any meaningful information about the structure of the world. The memory probes in this experiment were not correlated with the underlying structure of reward generation; nevertheless, it was hypothesized that the learner would use this picture, retrieve the information from that previous trial and use it to make a choice. It was shown that subjects who were not aware of the relevance of this memory probe treated this memory probe as a cue and retrieved the cued information to make a choice. This result suggests that humans believe that events that happen in similar context are similar themselves, thus previous information from a similar context can be used to make decisions when that context reoccurs. We would go further and suggest that human episodic memory is indeed optimized for situations in which contextual cues indicate the repetition of a past event. In our generative model, contextual cues provide useful information about the environment, and memory probes are paired with their associated rewards from that state of the world.

We suggest that in order to test the predictions from this chapter, it would be desirable to generate data from the scenarios described and use it to test behaviour of human subjects. An experiment can be designed where individual pictures are generated, and human subjects have to make a choice between two bandits with two different rewards. Specifically, we could test whether humans use the Rescorla-Wagner model when data is generated by scenarios 1, the Contextual Episodic when data is generated by scenario 4, and the Hybrid model when data is generated by scenario 3. To test this, we could collect data from human participants and perform model selection using these data to uncover the model that was used by the participants.

Collins and Frank influential work on the integration of working memory and reinforcement learning [Collins and Frank, 2012] speak directly to the frameworks proposed in Chapters 1 and 2. Collins and Frank showed that these systems can interact or compete for resources, and that humans use a mixture of these strategies, which appear to interfere with each other during learning and decision-making. They showed that a hybrid model between a Rescorla-Wagner algorithm and a capacity-limited working memory-based reinforcement learning algorithm fit their experimental human data the best. It is possible that most learning algorithms might need to include this component if they are to provide accurate models of the brain and behaviour. It is definitely still an open question for future work to understand how a working memory component could be included in our proposed algorithms.

One of the critical aspects that Collins and Frank stress is the constraint that arises from the limited capacity of working memory. We believe that understanding the physical implementation and limits of working memory is fundamental to properly developing algorithms and understanding cognitive computation. In our models, both Contextual Episodic and Hybrid, rely on working memory capacity. The number of episodes that can be retrieved by the Contextual Episodic model is constrained by working memory capacity. How well the Hybrid model can combine different sources of information is also constrained by working memory capacity. In general, most cognitive computations rely on working memory and are constrained by its capacity. Thus, understanding the neural substrates and limitations of working memory would be key to understanding human cognition, and developing accurate algorithms. In the following chapter, we explore the neural mechanisms of working memory. We focus on one alternative theory for the neural substrate of working memory that has the potential to expand our current understanding of its capacity.

Chapter 4

Redefining the Neural Substrates of Working Memory: A Mechanistic Approach

4.1 Motivation

Working memory is a theoretical construct used in various disciplines including cognitive psychology, neuroscience and computational neuroscience. Although there has yet to be a generally agreed definition, the term working memory usually refers to the core executive function that is responsible for the transient holding, processing, and manipulation of information [Baddley et al., 2003], [Baddeley, 2012]. This definition was first used by cognitive psychologist George A. Miller [Miller, 1956],[Cowan et al., 2004]. Based on lesion and imaging studies, working memory is associated with the dorsolateral prefrontal cortex, posterior parietal cortex, and hippocampus [Metcalfe et al., 2013]. However, transient holding of information has been observed in many other areas, including the motor cortex [Fuster and Alexander, 1971], [Alexander and Fuster, 1973], [D'Esposito et al., 2000], [D'Esposito, 2007]. It is a substantial open challenge to understand how recurrent neural circuits can act as a memory buffers, despite fast forgetting by individual neurons. Previous work has primarily focused on understanding memory lifetimes of recurrent circuits with fixed synapses with no intrinsic dynamics of their own. However, synapses are rich dynamical systems in their own right, and recent experimental findings [Stokes, 2015], [Lundqvist et al., 2016], along with previous theoretical proposals [Mongillo et al., 2008], [Buonomano and Maass, 2009] implicate these characteristics as supporting short-term memory.

In this thesis, we propose to address this controversy using theoretical analysis, and to test whether including synapses as part of the neural substrate of working memory could have a positive impact on memory capacity. Specifically, we propose that working memory is maintained in both the persistent activity of neurons and the dynamics of synapses. As a stimulus is presented, neural activity causes changes in the networks' synapses. During a delay period, information is maintained for the duration of time that the network is capable of maintaining that sustained activity plus the duration of time that the changes in the synapses last. For these reasons, we hypothesize that neuronal activity and dynamical synapses are jointly responsible for holding information in working memory. The advantage of this combination is that each component has different decay time constants, and consequently can increase working memory capacity.

Recently, an information-theoretic upper bound for the memory lifetime of short term memory was derived for linear recurrent networks with static synapses. Specifically, it was shown that any such linear network can, at most, achieve a memory life time proportional to the number of neurons in the network. Furthermore, it was shown that only a delay line, or any network that is equivalent to a delay line up to a unitary transformation, can saturate this bound [Ganguli et al., 2008]. Here, we extend this information-theoretic analysis to the case in which the neural substrate of working memory includes neurons and dynamical synapses. We aim to understand the role of dynamic synapses with short-term plasticity on memory performance. By linearizing a non-linear network, we study how short term plasticity modifies the effective connectivity matrix of the network to change the memory performance. We test this framework using different architectures, focusing on networks with very poor memory performance.

In the following sections, we review relevant literature from previous models of working memory, and techniques from non-linear dynamical systems and information theory. We then show that dynamical synapses indeed modify the internal structure of these networks and improve their memory performance by using the techniques described earlier.

4.2 Models of Working Memory

Working memory is essential for complex cognition, learning and guided behaviour [Miyake and Shah, 1999]. The controversy with respect to its definition lies in the fact that it is often confused with short term memory - the ability to hold information for short periods of time. The subtle difference between these two concepts is that working memory involves information that has been held for a short period of time and is actively been attended to or used to guide actions ([Baddeley and Hitch, 1974] [Baddeley, 2012], [Daneman and Carpenter, 1980]. Nevertheless, the terms short term memory and working memory are often used interchangeably in the literature, because of the conceptual overlap between them. Furthermore, there is still an open debate on whether these two concepts should exist as separate processes or whether short term memory should be considered a component of working memory [Aben et al., 2012]. In this section, we do not elaborate in the controversy between these two definitions. Instead, we focus on explaining how the most influential models of working memory have characterized this system. We then move on to explain the concept of working memory capacity, and finish with an overview of different algorithmic proposals for the implementation of working memory in neural systems. This sets the stage for our proposal for incorporating dynamical synapses.

The most famous model of working memory is the multi-compartment model introduced by Baddeley and Hitch [Baddeley and Hitch, 1974], [Baddeley, 2012]. The original model consists of three components: the central executive, the phonological loop and the visuo-spatial sketchpad. The central executive controls the flow of information from and to the other two systems. The phonological loop specializes in phonological information; such as, languages and sounds, while the visuo-spatial sketchpad specializes in visual and spatial information. Years after the original proposal, another component, the episodic buffer, was added. The episodic buffer provides a temporary storage of information in the form of single episodic representations [Baddeley, 2000]. As a whole, this model describes working memory as the information retained in short term memory, which is attended and coordinated by the central executive. It represents working memory as a separate memory system with specific parts for each type of information [Baddeley and Hitch, 1974] [Baddeley, 2012][Baddeley, 2000].

Another way of modeling working memory was introduced by Cowan [Cowan, 1999] . In his model, working memory consists of a subset of activated representations from long term memory. Therefore, unlike the previous model, working memory is not considered to be a separate system. Instead, it suggests that working memory is characterized by attention. The model consists of two levels. The first level consists of the activated representations in long term memory. The second level consists of those of these representations that are attended. This second level is called the focus of attention and it has a limited capacity of about 4 representations [Cowan, 1999].

O'Reilly and Frank developed model about the interactions between the pre-frontal cortex and basal ganglia [O'Reilly and Frank, 2006]. In this model, the pre-frontal cortex maintains active information during working memory tasks, while the basal ganglia selectively gates information in and out. In this way, this model explains how information is quickly updated and used for problem solving and decision making. Besides the network model, O'Reilly and Frank proposed that the pre-frontal cortex and basal ganglia interaction operates according to an actor-critic system to learn the best way to use information during working memory tasks [O'Reilly and Frank, 2006].

For a long time, working memory capacity or the number of items that can be maintained in working memory [Wilhelm et al., 2013] was believed to be equal to four [Cowan, 1999], [Cowan, 2016], [Cowan, 2001] or seven [Miller, 1956]. However, current research suggests that working memory capacity might depend on a variety of different factors such as the type of information been processed, and the amount of training or familiarity of the subject [Miller, 1956]. Another important concept that is relevant for capacity is that of ”chunks” [Miller, 1956], [Cowan, 2001], which are the real units of storage in memory, and which compress information. Different types of information are more easily compressed than others, because different aspects regarding the content of the chunks matter including the category of the information used (words, numbers, letters), the length of each category, the complexity of each item (for example, phonological complexity), the lexical status of the contents (whether the items are familiar to the subject: compressing the names of your family members under the chunk ”relatives” is a lot easier than compressing an equal number of names in Chinese). Furthermore, rehearsal aids compression, as well as individual differences and abilities matter [Cowan et al., 2000], [Service, 1998], [Gobet and Simon, 2000]. Understanding how to compress information to improve working memory capacity is an important open question in psychology and cognitive science, albeit not one that we consider further here.

In this thesis, we focus on understanding the physical limits of working memory capacity. These limits are not only very important for understanding working memory, but also provide insights into to how information is processed in neural circuits. In order to evaluate these physical limits, we first need to understand what the neural substrate of working memory are. According to Marr, this study falls into the implementational level of analysis. We consider that for working memory this level of analysis is very important, because it provides the requirements and constraints that we need to have when thinking of algorithms or computations that use working memory.

Several models have been developed to understand the neural substrates of working memory. Joaquin M. Fuster and Garrett Alexander observed an increase in firing frequency of nerve cells in the pre-frontal cortex and in the nucleus medialis dorsalis during a delay response task [Fuster and Alexander, 1971]. In this task, monkeys were presented with a stimuli for a brief period of time, and then asked to perform an action based on this stimuli after a short delay. The firing rates of activated neurons remained elevated throughout the delay period even when no external cue was presented. Figure 4.1 shows results from these delay experiments with a 30 seconds delay window. In this figure, persistent activity during the delay in the pre-frontal cortex (A) and in the nuclues medialis dorsalis (B) can be observed. This observation indicates that information is still present and been held in working memory in the period prior to the animal taking an action.

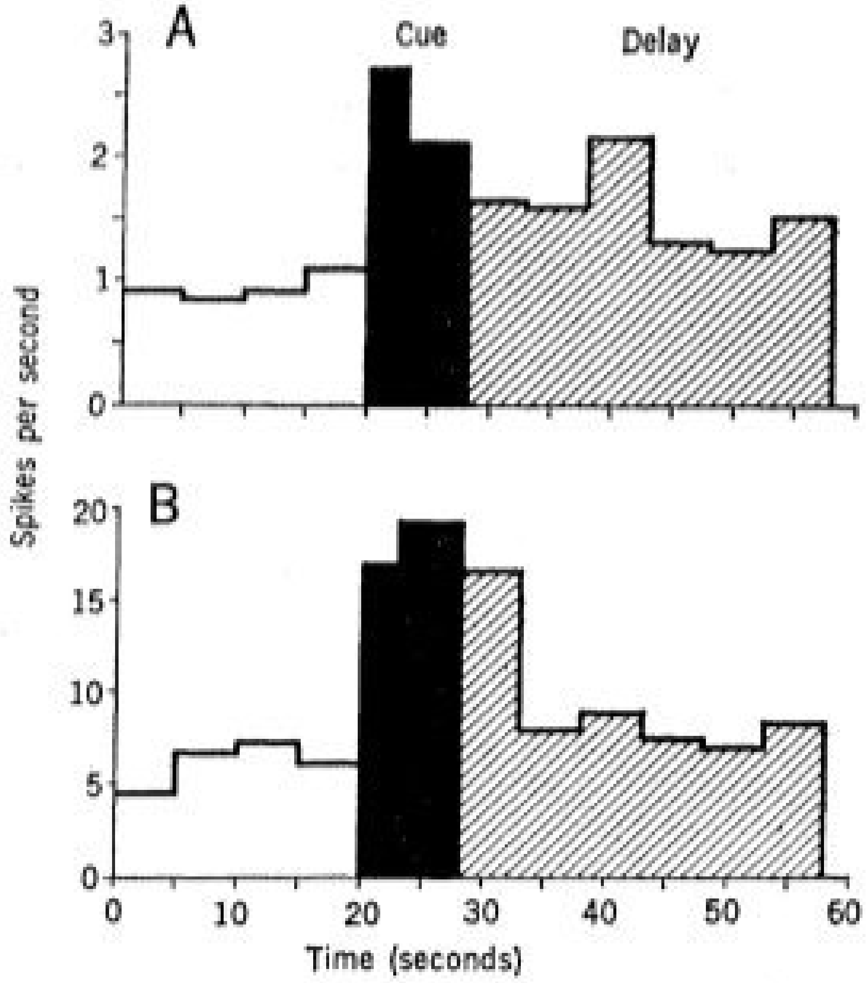

Figure 4.1: Average firing during delay response task from [Fuster and Alexander, 1971]. Average firing of two units during a five delayed response trials with 30 second delays (a) unit in prefrontal cortex (b) unit in nucleus medals dorsalis

Thanks to this experiment, Fuster and Alexander hypothesized that persistent activity, which was correlated with the narrowing of attention by the animal on information held in temporary memory storage, might be the mechanism through which neural systems retain information [Fuster and Alexander, 1971]. This experimental observation gave origin to the persistent activity theory of working memory. This theory suggests that short term memory maintenance relies on sustained elevated firing rates in a population of neurons. Many different mechanisms to explain this sustained firing activity have been developed. They include recurrent excitation within cell assemblies, synfire chains and single-cell bistability [Durstewitz et al., 2000], [Durstewitz and Seamans, 2006].

Recurrent excitation within cell assemblies is the most widely accepted proposal for sustained elevated firing. Within these models, the most famous model is the Hopfield model. This model stores discrete items in the synaptic weight matrix of the network and retrieves them as fixed point attractors [Hopfield, 1982]. A fixed-point attractor is an equilibrium point of a dynamical system that is represented by a particular point in the dynamic space. In this model, neurons that encode the same item are wired together and form a cell assembly. Each working memory activation corresponds to the activation of one pattern stored, and to the network dynamics associated with one equilibrium point.

The synfire chain hypothesis postulates that groups of neurons connected via feedforward links can propagate waves of synchronous excitatory activity in circulatory loops. In this way, information continues to be propagated and thus retained in the neural network [Hertz and Prügel-Bennett, 1996], [Goedeke and Diesmann, 2008].

Single-cell bistability proposes that neurons can be maintained two different stable states - resting and spiking. This phenomena is mediated by the non-linear current-voltage relationship of NMDA receptors, which can produce two stable fixed points of the neuron's membrane potential [Durstewitz and Seamans, 2002], [Seamans et al., 2001]. In this way, elevated firing rate can arise and be maintained as a fixed point of each neuron. The advantage of this mechanism is that it does not need previous synaptic learning, and therefore can store novel information faster than the two previously proposed models.

Recent experimental evidence has partly challenged the notion that stable persistent activity consititutes the neural substrate of working memory. Lundqvist et al. showed that working memory is highly dynamic and driven by oscillations. In this work, it was shown that activity in working memory consists of brief and variable bursts, and consequently, the observed sustained activity in previous experiments is mainly an due to trial and population averaging effects [Lundqvist et al., 2016].

Stokes has also postulated that working memory is mediated by a dynamic code besides persistent activity [Stokes, 2015]. His work showed that persistent activity was not critical for maintaining information during the delay period of a working memory task. He defined the term ”activity-silent” to refer to the activity still present in a neuronal assembly, but absent from the activity of the neurons. He designed a task that consisted of a delay working memory task with the addition of a disrupting stimuli during the delay period. Figure 4.2 (a) shows the standard working memory task from [Funahashi et al., 2004]. A monkey was trained to saccade in the direction of a presented stimuli after a delay period. Figure 4.2 (b) shows Stoke’s dual task, where the monkey is presented with a distractor in the delay period. Stokes recorded the activity of the neurons during the presentation of the stimuli, during the delay period with the distractors and during the onset of the guided response. The data collected showed that neuronal activity associated with the original stimuli decayed during the delay period due to the presence of the distractor. Figure 4.2 (c) shows the delay period, and how persistent activity was lost when the distractor was presented. However, once the cue that signals the saccade was given, this activity increased. Figure 4.2 (c) also shows how then the neural activity encoding the stimuli started to ramp up. This finding shows that even if the neural activity corresponding to the encoding of a stimuli is gone, information regarding the stimuli can still be present - potentially in some hidden states like the network’s synaptic components. For this reason, the animal was able to perform the task, as well as the neural activity was able to retrieve this information still ”silently” present in working memory.

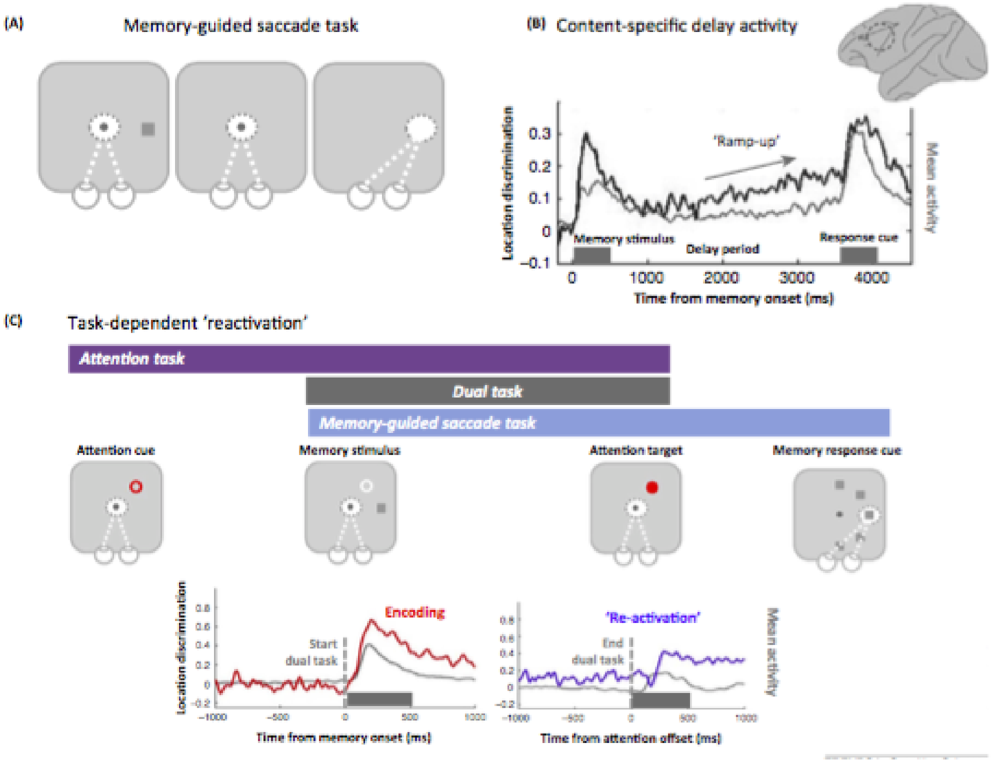

Figure 4.2: Working memory delay activity dual-task: working memory task is interrupted during the delay period with an attention task. Content-specific activity is abolished by the memory task and reactivated at the end of the dual task, reflecting a shift in focus to complete the working memory task upon receiving the cue signal [Stokes, 2015].

Stokes' hypothesis is that the neural substrate for this ”silent” working memory is the synaptic weights of the network [Stokes, 2015]. Information remains present in the modified synapses of the neural circuit. These experimental observations came as no surprise to the theoretical neuroscience community. Previously, theoretical models have argued for the importance of synaptic activity and dynamics in working memory. Hempel et al. suggested that synaptic augmentation (short-term synaptic enhancement of 40-60 % in synaptic transmission which lasts seconds to minutes) can enhance the ability of neuronal circuits to sustain persistent activity in response to transient inputs [Hempel et al., 2000] by boosting the level of recurrent excitation. In this way, synaptic enhancement was proposed as relevant factor in maintaining working memory. Hempel et. al [Hempel et al., 2000] showed that multiple forms of synaptic plasticity at excitatory synapses in rat medial prefrontal cortex ; in particular synaptic facilitation and synaptic depression, lead to synaptic augmentation - a longer lasting form of short term enhancement. They proposed and showed in computer simulations that this synaptic enhancement could led to enhancement in the capacity of a circuit to sustain persistent activity after a transient stimuli. Dynamic synapses increase the duration of reverberations in neural activity after stimulus presentation Mongillo et al. [Mongillo et al., 2008] took the role of synaptic dynamics one step further. They proposed a theoretical model where short term memory is stored in the dynamics of synapses. In this model, dynamic synapses are not a facilitator, but are the neural substrate of working memory. They suggested that, during encoding, neuronal activity changes synaptic efficacy, leaving a longer lasting memory trace [Mongillo et al., 2008]. We might think that the ”silent” memory observed by Stokes refers to the memory trace mediated by activity-dependent short term plasticity. We use Mongillo et al.'s model as our network model to study the effects of synaptic dynamics on memory capacity. We describe their equations in the following section.

4.3 Network Model

We study how short-term plasticity of synapses modifies the internal structure of recurrent neural circuits. To do this, we use a previously proposed model of short-term plasticity [Mongillo et al., 2008]. The model has two synaptic variables defined by the vectors $\mathbf{u}_t$ and $\mathbf{x}_t$, which refer to the synaptic variables corresponding to all neurons in the network. The utilization parameter (the fraction of resources utilized by each spike) is represented by $\mathbf{u}_t$, and refers to the residual calcium levels in the pre-synaptic terminal. $\mathbf{U}$ represents the increment of $\mathbf{u}$ produced by a spike. The amount of resources (neurotransmitters) available in each synapse are represented by $\mathbf{x}_t$.

The dynamics of $\mathbf{u}_t$ are given by :

$$\frac{d\mathbf{u}_t}{dt} = \frac{\mathbf{U}-\mathbf{u}_t}{\tau_f} + \mathbf{U}(1-\mathbf{u}_t)\delta(t-t_{spike}) \tag{4.1}$$

After each spike (at time t_{spike}), an amount equal to $\mathbf{u}_t\mathbf{x}_t$ is used, and $\mathbf{x}_t$ is reduced.

Because the amount of resources available are depleted at each spike, the dynamics that govern how fast these resources are depleted (the dynamics of $\mathbf{x}_t$ driven by spikes - equation 4.2) determine the dynamics of synaptic depression. On the other hand, $\mathbf{u}_t$ is increased with each spike due to influx of calcium, which increases the probability of release (probability of spike). For this reason, the dynamics of $\mathbf{u}_t$ (equation 4.1) determine the dynamics of synaptic facilitation. And the dynamics of $\mathbf{x}_t$ are given by:

$$\frac{d\mathbf{x}_t}{dt} = \frac{1-\mathbf{x}_t}{\tau_d} - \mathbf{u}_t\mathbf{x}_t\delta(t - t_{spike}) \tag{4.2}$$

Given these equations, both variables can saturate due to continuous spiking or come back to their baseline values ($\mathbf{x} = 1$ and $\mathbf{u} = \mathbf{U}$) with time constants τ_d (for synaptic depression) and τ_f (for synaptic facilitation) respectively once spiking stops. Depending on the value of the time constants, synapses are said to be facilitating (larger τ_f) or depressing (larger τ_f) [Mongillo et al., 2008].

In this thesis, we study the behaviour of dynamic synapses in a rate network model. For that, we average the dynamics of synapses over different realization of spike trains. We denote $\mathbf{r}_t$ the rate of all neurons in the network at time t. And obtain the following equations:

$$\frac{d\mathbf{u}_t}{dt} = \frac{\mathbf{U}-\mathbf{u}_t}{\tau_f} + \mathbf{U}(1-\mathbf{u}_t)\mathbf{r}_t \tag{4.3}$$

$$\frac{d\mathbf{x}_t}{dt} = \frac{1-\mathbf{x}_t}{\tau_d} - \mathbf{u}_t\mathbf{x}_t\mathbf{r}_t \tag{4.4}$$

A rate network with dynamic synapses evolves with the following dynamics:

$$\tau_m\frac{d\mathbf{r}_t}{dt} = -\mathbf{r}_t + \mathbf{W}\mathbf{u}_t\mathbf{x}_t\mathbf{r}_t + \mathbf{v}s_t \tag{4.5}$$

Here, $\mathbf{W}$ is the connectivity matrix between the neurons in the network, s_t represents a scalar, time-dependent signal that drives the recurrent network, and $\mathbf{v}$ is a time independent unit length vector of feed-forward connections from the signal into the neurons in the network. This model consists of three non-linear coupled differential equations describing the dynamic evolution of the recurrent rate network and the two synaptic variables.

4.3.1 Linearization

To facilitate our analysis, we linearize this non-linear network. Linearization is a dynamical systems technique used to study non-linear systems. It is performed by finding a linear approximation to a function at a given fixed point of the system. A fixed point is a point in the system where the net change is equal to zero. It is assumed that the behaviour within a small range of this equilibrium point can be approximated by a linear model. This linear function is approximated by the first order Taylor expansion of the non-linear system around this point [Strogatz, 1994]

Given a non-linear dynamical system described by:

$$\frac{d\mathbf{y}_t}{dt} = \mathbf{F}(\mathbf{y_t}) \tag{4.6}$$

The linear system is written as the first order Taylor expansion, where $\mathbf{y_o}$ is the fixed point of the system and J is the Jacobian matrix:

$$\frac{d\mathbf{y}_t}{dt} = \mathbf{F}(\mathbf{y_o}) + J(\mathbf{y_o})(\mathbf{y}_t - \mathbf{y_o}) \tag{4.7}$$

The Jacobian matrix is the matrix of all first-order partial derivatives:

$$J_{i,j} = \frac{\partial \mathbf{F}(y_i)}{\partial \mathbf{y_j}} \tag{4.8}$$

4.4 Fisher Memory Measures

Ganguli et al. studied the memory properties of generic high-dimensional dynamical systems. They used Fisher information theory to study the fundamental limits and properties of these systems to retain memory traces of previous inputs. In their work, they focused on linear recurrent neural circuits driven by noisy time dependent inputs, and derived theoretical upper bounds for memory capacity. Besides these theoretical bounds, they proposed memory measures called Fisher memory measures to quantify how much of a signal's history is encoded in the current state of a network. Here we describe these measures, and introduce the memory upper bounds [Ganguli et al., 2008]. Before that, we first define the linear network that was used for this analysis.

Ganguli et. al [Ghahramani, 2010] defined fisher memory measures for a discrete time linear network with dynamics given by:

$$\mathbf{r}_t = \mathbf{W}\mathbf{r}_{t-1} + \mathbf{v}s_t + z_t \tag{4.9}$$

In this equation, z_t is a zero mean gaussian white noise with covariance $\langle z_{t_1}, z_{t_2} \rangle = \varepsilon \delta_{t_1,t_2}$.

Fisher memory measures are useful measures of the efficiency with which the network state $\mathbf{r}_t$ encodes the history of the input signal. Because there is noise in this system, a given past signal $s_{t-k}|k \geq 0$ induces a conditional probability distribution $P(\mathbf{r_t}|\mathbf{s})$ [Ganguli et al., 2008]. For the purpose of this analysis, the history of the signal is defined as a temporal vector $\mathbf{s}$, where the kth component (s^k) is equal to $s_{t-k}|k \geq 0$. The measures consist of the Fisher memory matrix (FMM), the Fisher memory curve (FMC) and the Total Memory. The FMM between $\mathbf{r}_t$ and the past signal can be defined defined as:

$$F_{k,l}(\mathbf{s}) = \left\langle -\frac{\partial^2}{\partial \mathbf{s}^k \partial \mathbf{s}^l} \log P(\mathbf{r_t}|\mathbf{s}) \right\rangle \tag{4.10}$$

In this chapter, we change the notation from [Ganguli and Latham, 2009]. We use F to define the memory measures instead of J, which was used in Ganguli's work. The reason for this is that later in the chapter, we introduce a Jacobian matrix J, and we wanted to avoid confusion between these variables. This matrix is constructed based on the definition of Fisher information, where we consider the signal a parameter and the conditional probability of the current state given this signal the likelihood function given this parameter. Then, we can interpret the FMM as the Fisher information that the state of the network carries about the signal. In other words, the amount of information the current state of the network carries about the signal at a given time. This information measure quantifies the memory lifetime of the signal in the recurrent network. For this reason, it is called a memory measure [Ganguli et al., 2008]. We want to be able to apply these measures to our recurrent network equation 4.5. This network was defined using continuous differential equations. In order to apply these memory measures to our network, we will discretize these equations.

The most important terms of the FMM are the diagonal elements, which represent the information present in the network from a pulse signal k time step in the past. These diagonal elements capture the decay of memory and form the Fisher memory curve: FMC. Ganguli et al. named this curve $J_{k,k}$. In this thesis, we renamed it $F_{k,k}$. The off-diagonal terms of the FMM represent the interference from the signal at other time steps. Therefore, these terms represent how much of the signal is lost at each time step. The FMC can be written as [Ganguli et al., 2008]:

$$F_{k,k} = \mathbf{v}'\mathbf{W^{k}}'\mathbf{C}_t^{-1}\mathbf{W^{k}}\mathbf{v} \tag{4.11}$$

In this equation, $\mathbf{C}_t$ is the covariance matrix, and is equal to:

$$\mathbf{C}_t = \varepsilon \sum_{k=0}^{\infty} \mathbf{W}^k \mathbf{W}^{k\prime} \tag{4.12}$$

It can be seen that the FMC only depends on the time lag k, and on the type of network connectivity $\mathbf{W}$. For this reason, the connectivity matrix of a network defines its memory properties. The FMC is also a measure of the signal to noise ratio (SNR: amount of signal relative to amount of noise) present in the network. The kth component of the FMC represents the fraction of the original SNR of a pulse presented to the network k time steps in the past. The area under the curve of the FMC represents the total memory capacity of the network F_{tot} or the total SNR present in the network at time t about the entire history of the signal, relative to the original SNR. F_{tot} is defined as [Ganguli et al., 2008]:

$$F_{tot} = \sum_{k=0}^{\infty} F_{k,k} \tag{4.13}$$

Here we also changed the name of F_{tot} from Ganguli's paper in order to maintain consistency with the FCM $F_{k,k}$. The measures described above provide the tools to study memory lifetimes of recurrent circuits. Using these measures, Ganguli et al. derived theoretical bounds for two classes of network connectivities: normal (matrix with orthogonal basis of eigenvectors) and non-normal [Ganguli et al., 2008]. A real square matrix **W** is said to be normal if it commutes with its transpose such that[Asllani et al., 2018]:

$$\mathbf{W}\mathbf{W}^{'} = \mathbf{W}^{'}\mathbf{W} \tag{4.14}$$

Furthermore, the spectral theorem states that a matrix is normal if and only if it is unitarily similar to a diagonal matrix. In linear algebra, two square matrices **W** and **D** (here, a diagonal matrix) are unitarily similar if there exists a unitary matrix **U** such that[Asllani et al., 2018] :

$$\mathbf{W} = \mathbf{U}\mathbf{D}\mathbf{U}^{'} \tag{4.15}$$

This statement is the same as saying that **W** is diagonalizable by unitary matrix **U**.

For normal matrix **W**, the diagonal entries of **D** are its eigenvalues, and the columns of **U** are its eigenvectors. It follows that the eigenvectors of a normal matrix are orthogonal. On the other hand, a matrix is non-normal if it is not diagonalizable by a unitary matrix, and therefore, its eigenvectors are not orthogonal to each other. This statement is thus equivalent to the condition that a for a non-normal matrix W [Asllani et al., 2018]:

$$\mathbf{W}\mathbf{W}^{'} \neq \mathbf{W}^{'}\mathbf{W} \tag{4.16}$$

What distinguishes normal and non-normal networks is whether they are diagonalizable or not. It is important to note that through the Schur-decomposition, we can see whether a given matrix is normal or non-normal. The Schur-decomposition holds for any square matrix **W**, allowing us to write it as unitarily equivalent to an upper triangular matrix, whose diagonal elements are the eigenvalues of **W**. As we stated before, if the matrix is normal, the only non-zero elements of this upper triangular matrix will be the diagonal terms. In this way, if we perform the Schur decomposition of any matrix, we can see whether the matrix is normal or not by looking at the result of this unitary transformation. In the experiments in this chapter, we use this technique to analyze whether a network is normal or not. The Schur decomposition states that for a normal matrix **W** can be decomposed as equation 4.16. For a non-normal matrix W, it states that:

$$\mathbf{W} = \mathbf{U}\mathbf{T}\mathbf{U}^{'} \tag{4.17}$$

where **U** is a unitary matrix and **T** is an upper triangular matrix.

We note that both **D** and **T** are upper triangular matrices, but we refer as diagonal matrices to those with non-zero elements strictly in the diagonal. D and T are known as Schur Forms, and we address them as such in the rest of the text. The characteristics and differences between these two types of matrices have implications for dynamical systems. In our situation, we use these matrices as network connectivity matrices. We call the networks with normal connectivity matrices, normal networks, and the networks with non-normal connectivity matrices, non-normal networks. In the case of linear normal networks, one can write the solution to the system as a linear combination of exponentially relaxing modes, whose characteristic time scales are given by the inverse of each corresponding eigenvalue. In these way, the eigenvalues and eigenmodes are responsible for driving the long term dynamics of these networks. In the case of linear non-normal networks, more complex patterns can emerge. In these networks, eigenvectors do not form an orthonormal basis, so the canonical characterization of a linear system by its spectrum is unreliable. Eigenvalues can become extremely sensitive to noise and consequently, their physical meanings are diminished. Furthermore, small disturbances can undergo a transient phase and be strongly amplified. Here, we define the memory bounds for these two connectivities as were derived in the work by Ganguli et al. [Ganguli and Latham, 2009]:

For normal networks: $F_{tot} = \frac{1}{\varepsilon}$

For non-normal networks: $F_{tot} = \frac{N}{\varepsilon}$

where ε is the noise injected to the rate network equation 4.9.

The main results from [Ganguli et al., 2008] are that F_{tot} represents how much of the entire signal history can be remembered by a given network. It defines the memory lifetime of each network. When the memory lifetime is proportional to the number of neurons in the network, this network has extensive memory. Ganguli et. al showed that only non-normal networks can achieve extensive memory, and normal networks can only have a memory lifetime proportional to 1. The reason for this is that normal networks are diagonalizable, whereas non-normal networks can be unitarily transformed into lower triangular matrices, which imply a hidden feed-forward connectivity. Furthermore, normal networks' dynamics are driven solely by their eigenspectrum, while non-normal networks are capable of transient expansion and compression [Kerg et al., 2019] These transient dynamics have been shown to have many computational advantages over normal matrices [Hennequin et al., 2012].

Ganguli's results imply that when a signal enters a network with normal connectivity, the signal does not get amplified. In this way, the signal gets lost over time due to the increase in noise from interactions between neurons and corruption from previous time-steps of the signal. Furthermore, if one were to optimize for remembering the signal, one would have to choose between remembering only the recent past or the remote past. For the signal to be preserved in time, a special type of amplification is required. It was concluded that at least super-linear amplification is necessary. The results above also show that non-normal networks can have extensive memory - a memory lifetime proportional to the number of neurons in the network. Networks that saturate this bound are said to achieve extensive memory. Non-normal networks are the only networks capable of having this memory capacity. The reason for this is that they have a hidden feed-forward architecture that allows them to transiently amplify the signal, and maintain a high fraction of the original SNR over time. When Ganguli et al. derived these bounds, they also established the conditions for memory performance related to network connectivity [Ganguli et al., 2008]. The main points are outlined below:

1. Only networks with a hidden feed-forward architecture can achieve a memory capacity greater than 1. Only non-normal networks have this property. Hidden feedforward connectivity can be observed from the Schur Forms of a network. If the network is upper triangular, with non zero elements outside the diagonal, then it can have a feedforward structure. This finding agrees with previously proposed observations that neural circuits may actually behave like feedforward networks in the brain [Ganguli and Latham, 2009].

2. Super-linear transient amplification, that lasts for at least a time of order N, is necessary for a network to achieve extensive memory. Network amplification refers to the gain of the initial input signal as it passes through the network. Signal gain consists of scaling the input signal by a factor, and it occurs thanks to the signal been passed on to different neurons each time step. The delay line is the network structure that can achieve the most signal amplification compared to noise amplification. Mathematically, network amplification is defined as [Ganguli and Latham, 2009]:

$$A(k) = \mathbf{W}^k \mathbf{v}^2 \tag{4.18}$$

   For normal networks, this amplification monotonically decreases, while for non-normal networks it can transiently increase. Therefore, only non-normal networks can satisfy this property [Ganguli and Latham, 2009].

3. Furthermore, the only non-normal network architecture that can achieve extensive memory is the delay line, or any network that can be unitarily transformed to a delay line. The reason for this is that the delay line is the only

network that can achieve the least noise amplification for a given amount of signal amplification [Ganguli and Latham, 2009].

Another important concept for memory performance is the input connectivity. For normal networks, any type of network connectivity achieves the same performance, since we showed that memory lifetime for normal networks can be at most O(1) (order 1). However, for non-normal networks, it is important, because the direction in which the input is fed can have a great impact in how it is amplified. An important concept when analyzing the best direction to feed the input in the network is the Spatial Fisher Information : F_s. (We also changed the name from J_s in Ganguli's paper to F_s for consistency with our notation). F_s measures the information in the network's spatial degrees of freedom about the entire signal history[Ganguli and Latham, 2009], and is defined as:

$$F_{si,j} = \Sigma_{k=0}^{\infty}[\mathbf{W}^{k'}\mathbf{C}_t^{-1}\mathbf{W}^k]_{i,j} \tag{4.19}$$

In general, placing the signal in the direction of the eigenvector of the largest eigenvalue of the spatial Fisher information F_s counterbalances noise propagation inside the network and maximizes total information. The reason for this is that in the case of non-normal matrices, F_s does not have a trivial spatial structure, and consequently the largest principal component contains the largest amount of information. It has been shown that by placing the signal in this direction, one can guarantee that the largest memory capacity will be obtained [Ganguli et al., 2008]. For the analysis we perform in this chapter, we use this input connectivity that maximizes network performance.

Using, F_s, the total Fisher information is equal to:

$$F_{tot} = \mathbf{v}'F_s\mathbf{v} \tag{4.20}$$

Inspired by these theoretical findings, we ask the question of whether adding dynamic synapses to a linear recurrent network with normal connectivity matrix could change its connectivity and as a consequence increase its memory capacity. From the recent experimental observations during delay working memory experiments, we know that synaptic variables are key for memory capacity. For this reason, we hypothesize that models of recurrent neural circuits have failed to achieve extensive memory, because they do not included all the relevant variables in the neural circuit. Often time, neural circuits made up of neurons and static synapses do not have transient amplification. We hypothesize that dynamic synapses can modify the structure of the circuits. It has been argued that dynamic synapses can increase the amount of time a memory trace is present in a neural circuit [Maass and Markram, 2002] [Stokes, 2015] [Mongillo et al., 2008], as well as prevent memories from been over-written by new memories [Lahiri and Ganguli, 2013] [Zenke et al., 2017] [Amit and Fusi, 1994] [Poole et al., 2017]. For these reasons, we believe that adding dynamic synapses produces structural changes in the network of neurons and synapses that increase memory performance. In the results section, we show how adding dynamic synapses does change network structure. For that, we linearize a non-linear network with dynamic synapses, and obtain an effective connectivity matrix which includes neurons and dynamic synapses as variables. We describe this procedure in the next section.

4.5 Model Implementation

We use Mongillo's model of a linear recurrent neural network with synaptic variables to test the hypothesis described in this chapter. First, we discretize the system of differential equations so that our analysis is parallel to that performed by Ganguli [Ganguli and Latham, 2009] . The resulting equations.

$$\mathbf{r}_t = \mathbf{r}_{t-1} + \frac{dt}{\tau_m}(-\mathbf{r}_{t-1} + \mathbf{W}\mathbf{u}_{t-1}\mathbf{x}_{t-1}\mathbf{r}_{t-1} + \mathbf{v}s_t + z_t) \tag{4.21}$$

$$\mathbf{u}_t = \mathbf{u}_{t-1} + dt(\frac{U - \mathbf{u}_{t-1}}{\tau_f} + U(1 - \mathbf{u}_{t-1})\mathbf{r}_{t-1}) \tag{4.22}$$

$$\mathbf{x}_t = \mathbf{x}_{t-1} + dt(\frac{1-\mathbf{x}_{t-1}}{\tau_d} - \mathbf{u}_{t-1}\mathbf{x}_{t-1}\mathbf{r}_{t-1}) \tag{4.23}$$

Because this model is non-linear and coupled, we linearize it to be able to use the Fisher memory measures described earlier. Furthermore, this procedure allows us to construct a Jacobian matrix (or the new connectivity matrix between all variables: neurons and synapses), and use it as the effective connectivity matrix of the new network. As long as perturbations to the system are small, this new linearized network is a good approximation of the original network. The Jacobian of the system is equal to the derivatives of the system evaluated at a fixed point. We call the Jacobian matrix the effective connectivity matrix, because we believe that information is effectively being propagated in the new network formed by the neurons and the synapses.

At the fixed point, all variables in the network reach their steady state values. $\mathbf{r}_0$ corresponds to the steady state value of all the neurons in the networks at the fixed point. This value is obtained empirically. For the synaptic variables, the steady state values are given by the following equations :

$$\mathbf{u}_o = U\frac{1+\tau_f\mathbf{r}_o}{1+U\tau_f\mathbf{r}_o} \tag{4.24}$$

$$\mathbf{x}_o = \frac{1}{1+\mathbf{u}_o\tau_d\mathbf{r}_o} \tag{4.25}$$

We compute the Jacobian matrix with respect to the network (neurons and synapses): J, and with respect to the input: J_s. We do this in order to have a consistent linearized system, where J is the effective network connectivity matrix :

$$\mathbf{J} = \begin{vmatrix} J_{rr} & J_{ru} & J_{rx} \\ J_{ur} & J_{uu} & J_{ux} \\ J_{xr} & J_{xu} & J_{xx} \end{vmatrix} \tag{4.26}$$

where:

$$J_{rr} = \frac{1}{\tau_m}(-I + \mathbf{W}\mathbf{u_o}\mathbf{x_o}) \tag{4.27}$$

$$J_{ru} = \frac{1}{\tau_m}(\mathbf{W}\mathbf{r_o})\mathbf{x_o}) \tag{4.28}$$

$$J_{rx} = \frac{1}{\tau_m}(\mathbf{W}\mathbf{r_o}\mathbf{u_o}) \tag{4.29}$$

$$J_{ur} = (U(1 - \mathbf{u_o})) \tag{4.30}$$

$$J_{uu} = -\frac{1}{\tau_f} - U\mathbf{r_o} \tag{4.31}$$

$$J_{ux} = \mathbf{0} \tag{4.32}$$

$$J_{xr} = -\mathbf{u_o}\mathbf{x_o} \tag{4.33}$$

$$J_{xu} = -\mathbf{r_o}\mathbf{x_o} \tag{4.34}$$

$$J_{xx} = -\frac{1}{\tau_d} - (\mathbf{r_o}\mathbf{u_o}) \tag{4.35}$$

and J_s is the effective input connectivity://

$$\mathbf{J}_s = \begin{vmatrix} J_{rs} \\ J_{us} \\ J_{xs} \end{vmatrix} \tag{4.36}$$

$$J_{rs} = \mathbf{v} \tag{4.37}$$

$$J_{us} = 0 \tag{4.38}$$

$$J_{xs} = 0 \tag{4.39}$$

We use these matrices to construct the new linearized system:

$$\mathbf{Z}_t = \mathbf{Z}_{t-1} + dt(\mathbf{J}\mathbf{Z}_{t-1} + \mathbf{J}_s s_t) \tag{4.40}$$

where $\mathbf{Z}_t$ is a vector with rows equal to the perturbations of all variables of the system from the steady state :

$$\mathbf{r}_t - \mathbf{r}_o \tag{4.41}$$

$$\mathbf{x}_t - \mathbf{x}_o \tag{4.42}$$

$$\mathbf{u}_t - \mathbf{u}_o \tag{4.43}$$

We implemented the model above with the parameters previously used in [Mongillo et al., 2008] to study the effects of short term synaptic plasticity in recurrent networks memory capacity. These parameters are shown in the table below:

Parameters	
Number of neurons	100
τ_d	0.20 (ms)
τ_f	1.5 (ms)
τ_m	0.15 (ms)
U	0.2

Before we proceed with the analysis of memory performance, we show how well the linearized system matches the behaviour of the non-linear system under small perturbations. We remind the reader that linearization only works under small perturbations. First, we drive the non-linear network with an input pulse, and find its steady state. At this steady state, we linearize the network using the technique described above. Finally, we test the response of both these networks under a small perturbation: an input pulse of 0.15 Hz with gaussian noise with zero mean and variance equal to 0.001. We show in Appendix E how both networks respond similarly under this perturbation. For this simulation, we used the symmetric connectivity matrix, but the Orthogonal network response in the same manner. In the following section, first, we show our analysis using the parameters defined here, and afterwards, we show how memory performance is affected when these parameters change.

4.6 Network Structure Analysis

Using the linearized network described above, we compute the Fisher memory measures for two different type of connectivity matrices: Orthogonal and Symmetric. We chose Orthogonal and Symmetric connectivity matrices, because they are normal matrices, and ideal to test our hypothesis that adding synaptic variables changes the effective connectivity of the network and can improve memory performance. Here we present the results of our analysis. First, we analyze the structure of the networks before and after adding dynamic synapses, and then we show the implications of these changes for memory performance.

4.6.1 Symmetric Network

First, we analyze the eigenspectrum of the effective connectivity matrices before and after including dynamical synapses. We call them 'network with only neurons' or 'network with static synapses', and 'network with dynamical synapses'. We present the results for the effective connectivity matrices and parameters defined earlier. We start our analysis with a network with symmetric connectivity. This network is a normal network, and it is characterized by having a flat eigenspectrum (all eigenvalues are real) in the complex plane. Figure 4.3 shows the distribution of eigenvalues of a symmetric matrix. Figure 4.4 shows how the distribution of eigenvalues changes after including dynamic synapses in the system (The eigenvalues from the effective connectivity matrix from the linearized system with dynamic synapses).

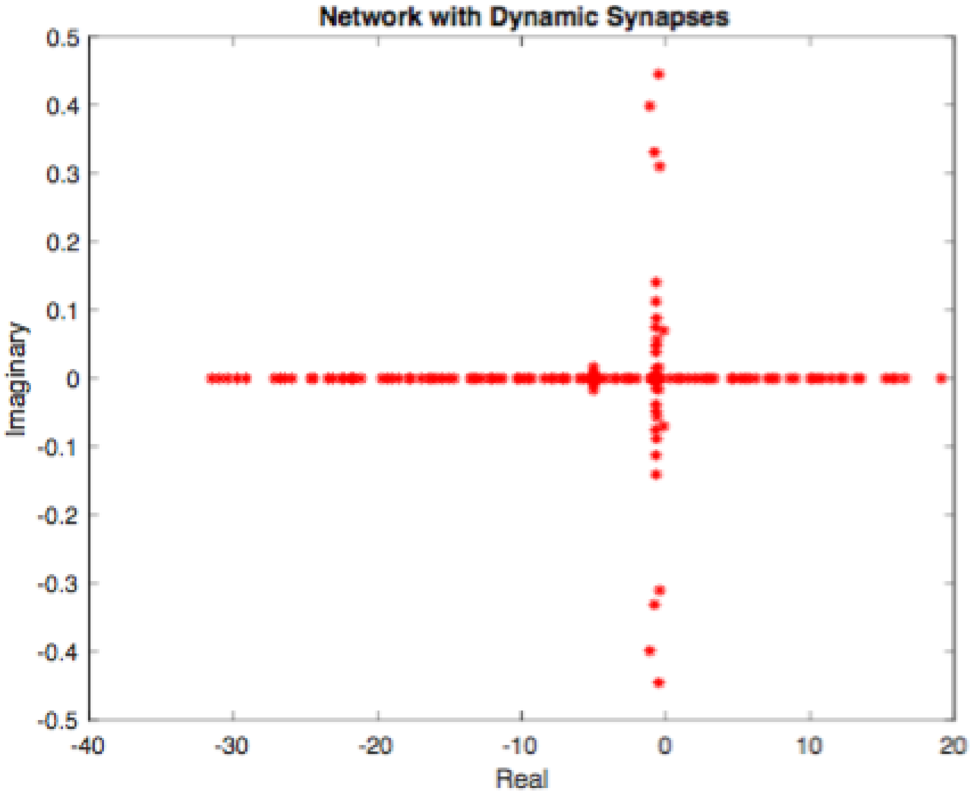

Figure 4.3: Eigenspectrum: Symmetric Network with dynamic synapses

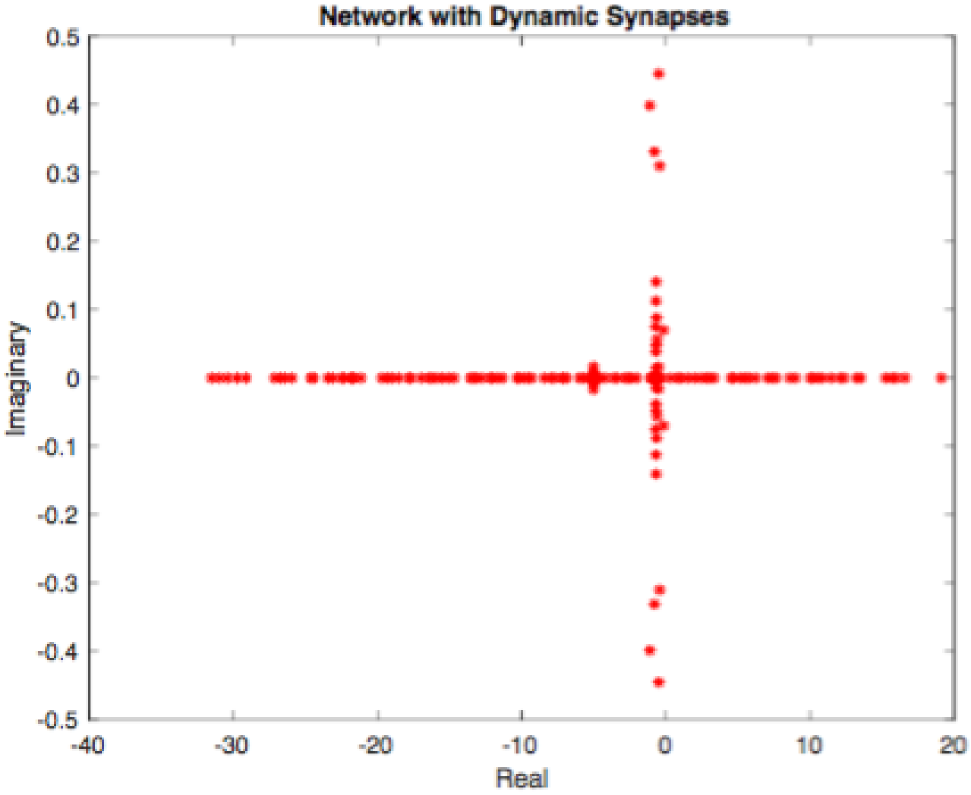

Figure 4.4: Eigenspectrum: Symmetric Network with dynamic synapses without dynamic synapses

From these results, we can see that indeed adding synaptic variables modifies the internal structure of the connectivity matrices. It can be seen that the addition of dynamic synapses causes the eigenspectrum to deviate from the spectrum of a symmetric matrix. This analysis shows that dynamic synapses indeed can cause structural changes to the network. However, in order to see whether these changes imply that the network is no longer normal, we perform the Schur decomposition of the connectivity matrix before and after the addition of dynamic synapses. We analyze the resulting Schur Form for each of these matrices. We know that only if the Schur Form is diagonal, the matrix is a normal matrix. Figure 4.5 shows the resulting Schur Form for the symmetric connectivity matrix before the addition of dynamic synapses and Figure 4.6 shows the symmetric connectivity matrix after the addition of dynamic synapses. A zoom version of the schur form for network with dynamic synapses can be seen in Appendix F, where the entries along the delay line can be seen more clearly.

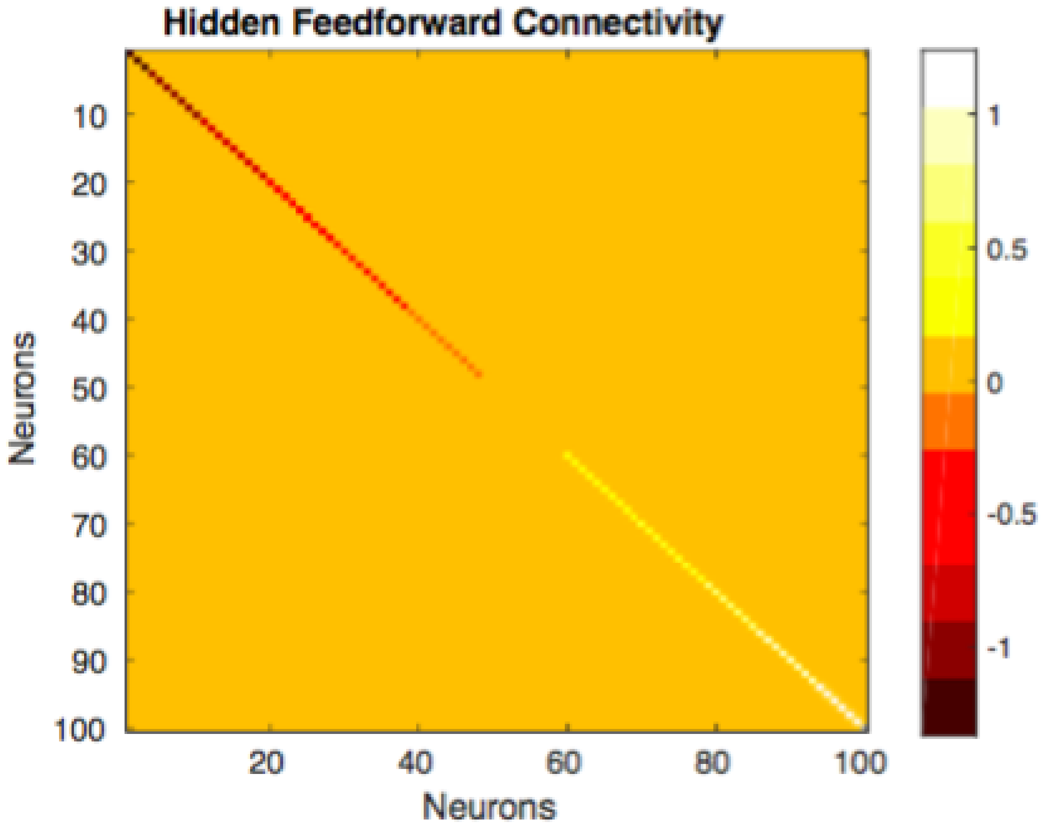

Figure 4.5: Schur Decomposition: Symmetric Network with dynamic synapses

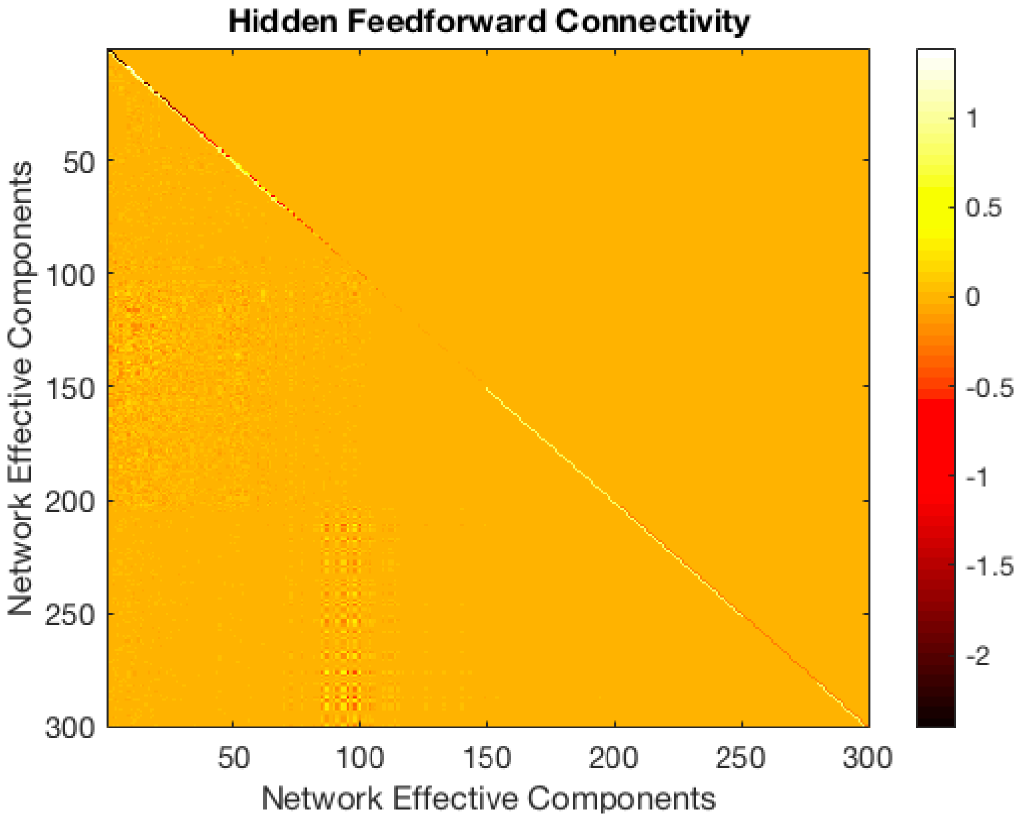

Figure 4.6: Schur Decomposition: Symmetric Network without dynamic synapses

We can see that the network with dynamic synapses is no longer a normal matrix. The diagonal matrix from the network without dynamic synapses is transformed into a lower triangular matrix when dynamic synapses are added. Deviation from normality increases the potential for a network to achieve a memory capacity proportional to the number of neurons in the network or extensive memory [Ganguli et al., 2008]. Figure F.1 simply shows the connectivity matrix between the effective elements of each network. For the case of the network with no dynamic synapses, it shows the connectivity matrix between neurons. For the network with dynamic synapses, it shows the connectivity matrix between all effective components of the network: neurons and synapses. The colorbar is a way to show which values are comparatively higher, and thus driving the network. The background color yellow represents entries equal to zero. For both matrices shown, most entries are equal to zero. In the case of the matrix from the network with static synapses, the only non-zero elements are in the diagonal. Some of these elements have values less than zero (red color), and higher than zero(light yellow). In the matrix corresponding to the network with dynamic synapses, the diagonal entries are color red (less than zero), while the lower triangular entries are bright yellow (greater than zero) - it is for this reason that the plot looks ”lighter”. Bright yellow entries appear more opaque (due to the colorbar selection), but they correspond to a higher entry value. As we explained earlier, a lower diagonal matrix has a hidden feed-forward architecture. This implies that this network could be capable of achieving a higher memory performance. In Ganguli’s work, it was shown that this hidden structure is necessary to achieve extensive memory, if it translates into an adequate network amplification. The condition found by [Ganguli et al., 2008] is that this network amplification needs to be at least super-linear. We show the network amplification, defined in equation 4.20 for the symmetric connectivity network with and without dynamic synapses in Figure 4.7.

As can be seen, the network with dynamic synapses 4.7 (red) can transiently amplify an input signal. Figure 4.7 shows how the network amplifies the signal. This signal amplification allows the network to have a higher memory capacity than

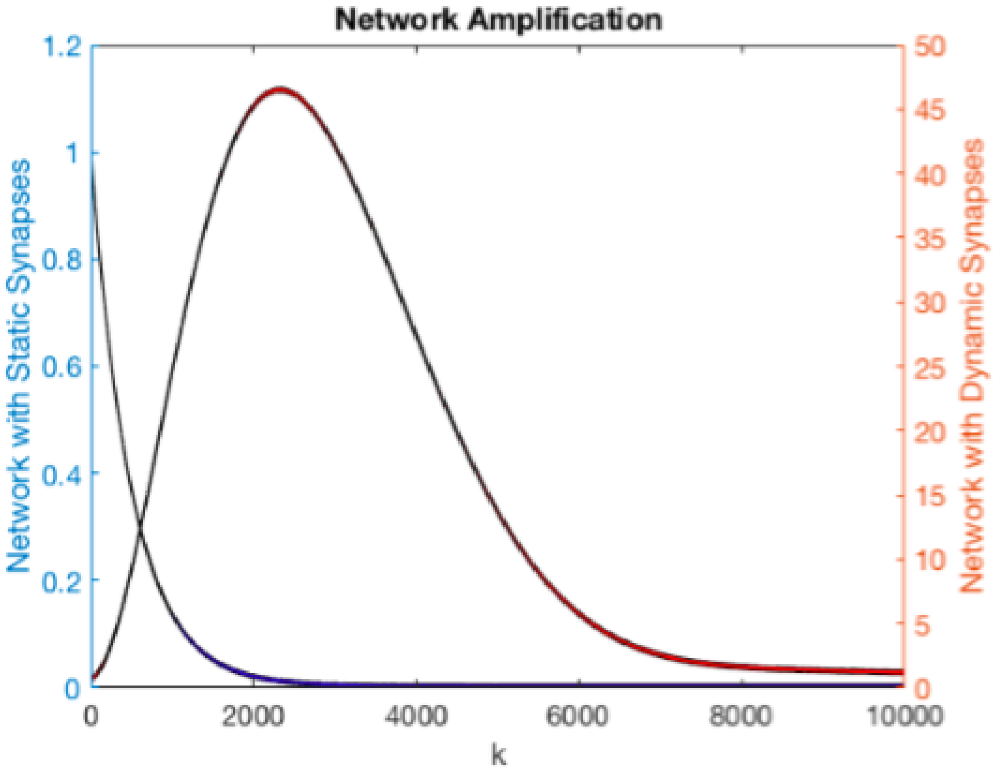

Figure 4.7: Network Amplification: Symmetric Network with dynamic synapses (red) and without dynamic synapses (blue)

its static synapses counterpart. Network amplification, in the case of the network with dynamic synapses, was computed using all network's components: neurons and dynamic synapses. The reason for this is that the hypothesis of this chapter is that an input signal is stored in the neuron's activities and in the dynamic synapses - each with their respective time scales. For this reason, how the signal is amplified over time involves both these components. For this reason, the values of network amplification are much higher in the network with dynamic synapses. However, it is not the values what is the most important feature for us, but the shape of the curves. Furthermore, the signal that can be read from a network is present on the activity of the neurons only. In order to compute the information content available or present in the network at time t, we use the Fisher memory measures described in the previous section using only neurons as elements for both the network with dynamic synapses and the network with static synapses. We show the results of the

Fisher Memory Curve in Figure 4.8, which shows how much of the signal is present k time steps after it was originally presented. The results shown for the FMC and Network Amplification are for 50 realizations of each network. The equation for the Fisher Memory Curve was described in equation 4.12.

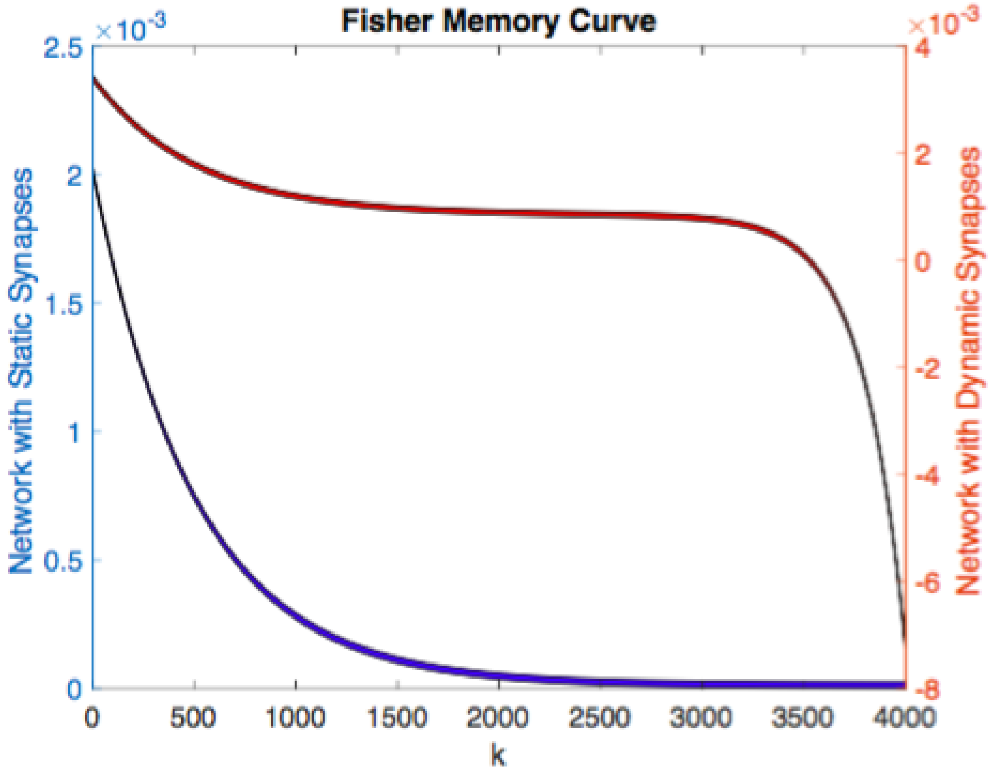

Figure 4.8: Fisher Memory Curve: Symmetric Network with dynamic synapses (red) and without dynamic synapses (blue)

Another important measure that arises from the Fisher Memory Curve is the Total Memory, which is the area under the curve of the FMC. The Total Memory measures the amount of information present in the network from the entire history of the input signal at the current time t. The Total Memory (TM) for the network without dynamic synapses is equal to 1. This result is consistent with the theoretical prediction for normal matrices. The TM for the network with dynamic synapses, using the parameters defined earlier, is equal to 6.6 compared to 1 which is the TM for normal networks (or this network before adding dynamic synapses). In the following section, we show how the Total Memory scales with the number of neurons in the network. This result indicates that there is an increase in memory performance when dynamic synapses are included. More importantly, the addition of dynamic synapses allows the network to break the normal matrix memory upper bound of 1. This result is important, because it shows that indeed other variables present in the network can play a role in short term memory capacity. Consequently, showing that the current picture of persistent neuronal activity as the neural substrate of working memory is incomplete, and that the memory performance of artificial circuits can be improved if dynamic synapses, as well as all the molecular machinery involved in synaptic communication, are included. In this analysis, we included only two synaptic variables, but it is known that there are many variables involved in synaptic communication with a cascade of time constants [Südhof, 1995].

In order to test how much of this memory performance is due to specific parameters, we show how the Total Memory of the network changes as the synaptic parameters τ_f and τ_d vary. We chose the parameter range for which network behaviour does not saturate - if we increased or decreased the networks' time constants any further, there was no further change in the dynamics. We show the results of how TM varies as each synaptic time constant (depression or facilitation) changes given the value of the other time constant is kept fixed.

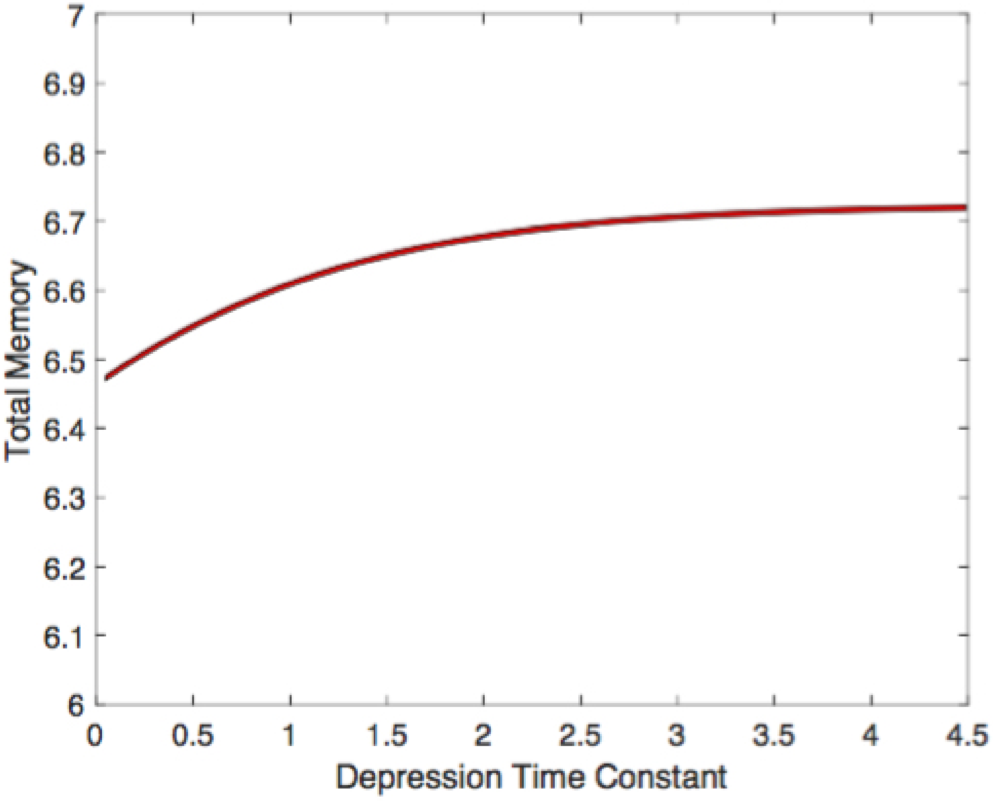

Figure 4.9: Total Memory with varying τ_f

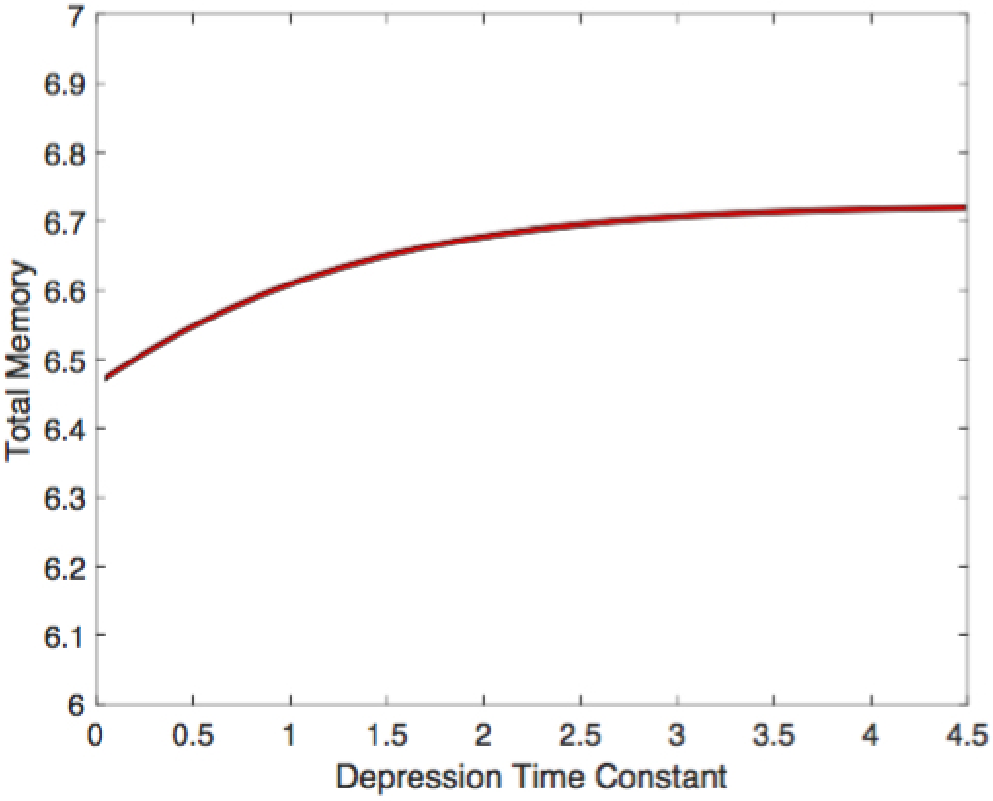

Figure 4.10: Total Memory with varying τ_d

Figure 4.9 shows how TM changes as τ_f increases and decreases from the value we used to for our analysis: $\tau_f = 1.5ms$. We can see that the memory increases as the memory time constant increases. This result makes sense, because the larger the memory time constant, the longer the effect of the input signal will last in the synapse. The same effect can be seen in Figure 4.10 . The main difference is that memory performance does not change as much as in the case of facilitation. The reason for this is that the time constant of facilitation is already large enough, and by using the dynamics defined by these two variables, the network cannot achieve a much larger memory capacity.

4.6.2 Orthogonal Network

In this section, we analyze the properties of an orthogonal network when dynamic synapses are added. The most salient result from this network connectivity is that the Schur Form is not very different from the one of the network without dynamic synapses. Below we show the figures that show how an orthogonal network does not change significantly with the addition of dynamic synapses.

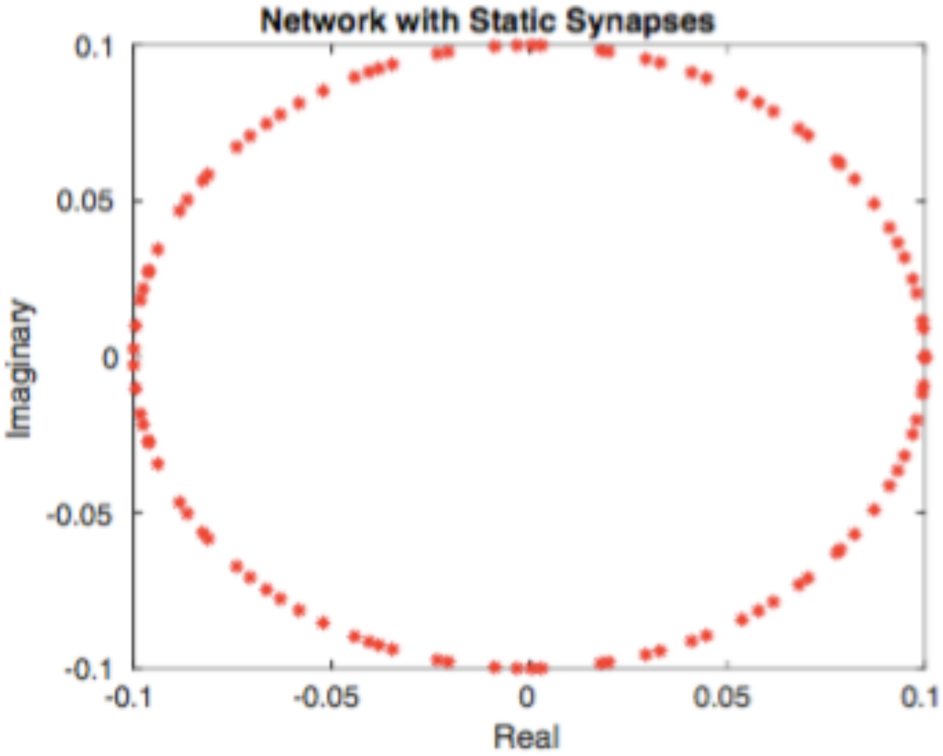

Figure 4.11: Eigenspectrum: Orthogonal Network with dynamic synapses

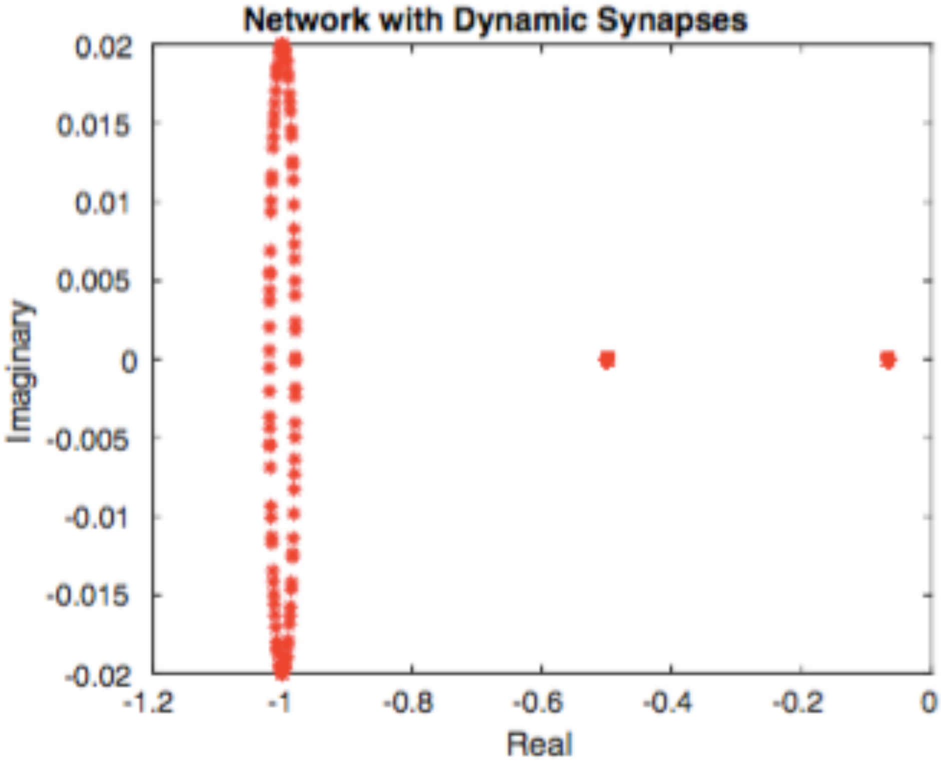

Figure 4.12: Eigenspectrum: Orthogonal Network without dynamic synapses

The orthogonal network preserves its diagonal hidden structure. This structure is closer to that of a normal matrix. For this reason, it is possible, and we will show that the network with orthogonal connectivity does not improve its memory capacity with the addition of dynamic synapses.

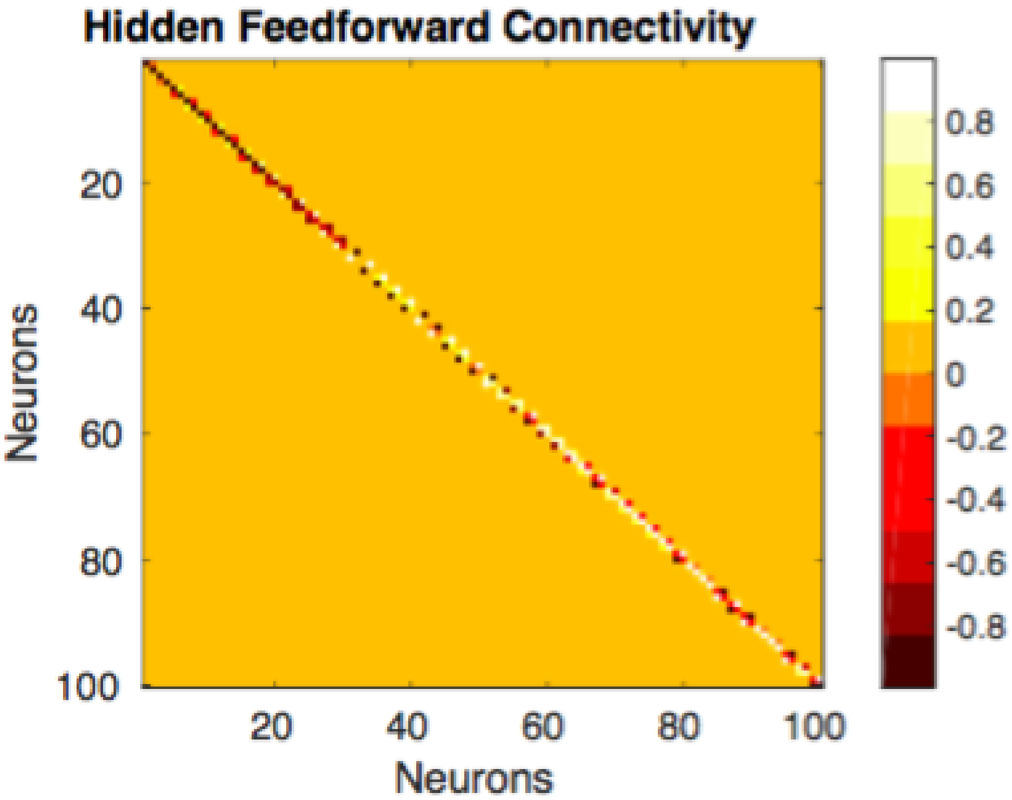

Figure 4.13: Schur Decomposition: Orthogonal Network with dynamic synapses

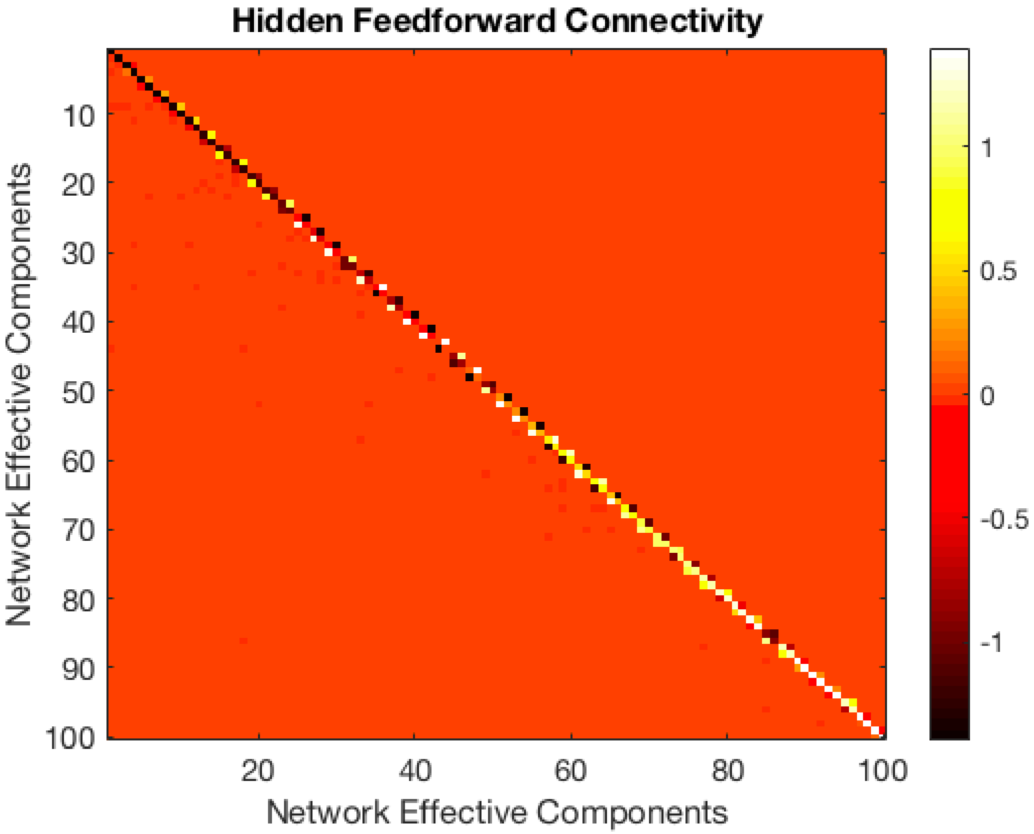

Figure 4.14: Schur Decomposition: Orthogonal Network without dynamic synapses

Following the results shown above,the network amplification of the orthogonal network with and without dynamic synapses decays with time. It does not transiently amplify, as was the case for the network with Symmetric matrix after the addition of dynamic synapses. This amplification profiles corresponds to that of normal matrices. Normal network amplification decays with time, and thus it has a poor memory capacity.

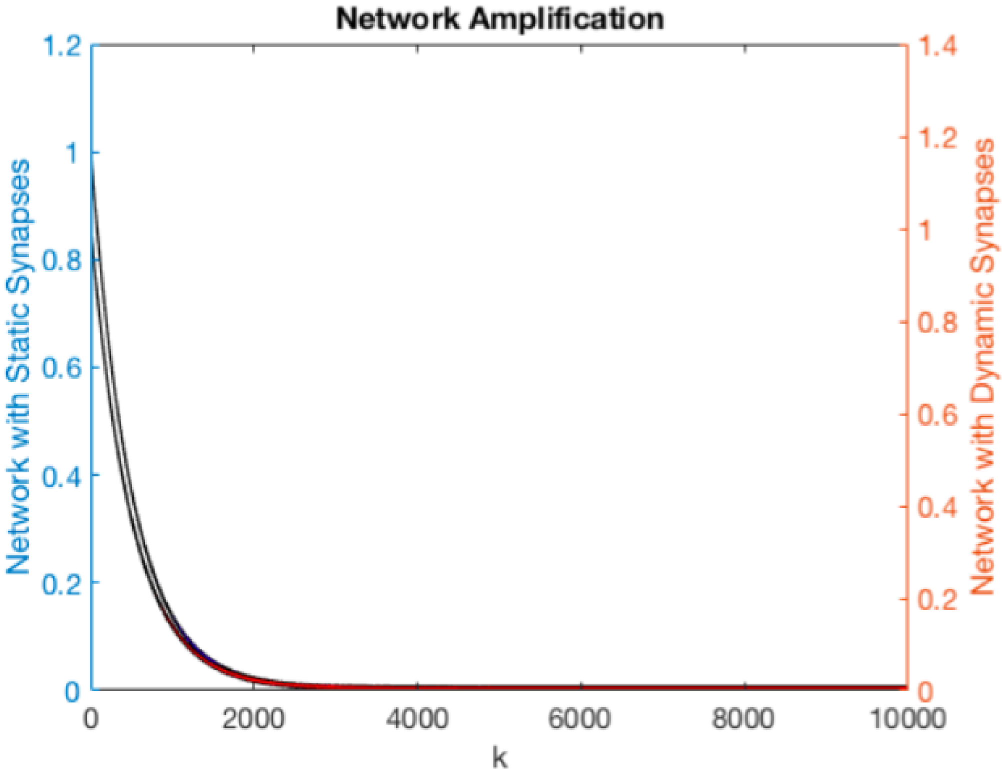

Figure 4.15: Network Amplification: Orthogonal Network with dynamic synapses (blue) and without dynamic synapses (blue)

Finally, we show that this network amplification profile does translate into a poor memory performance.

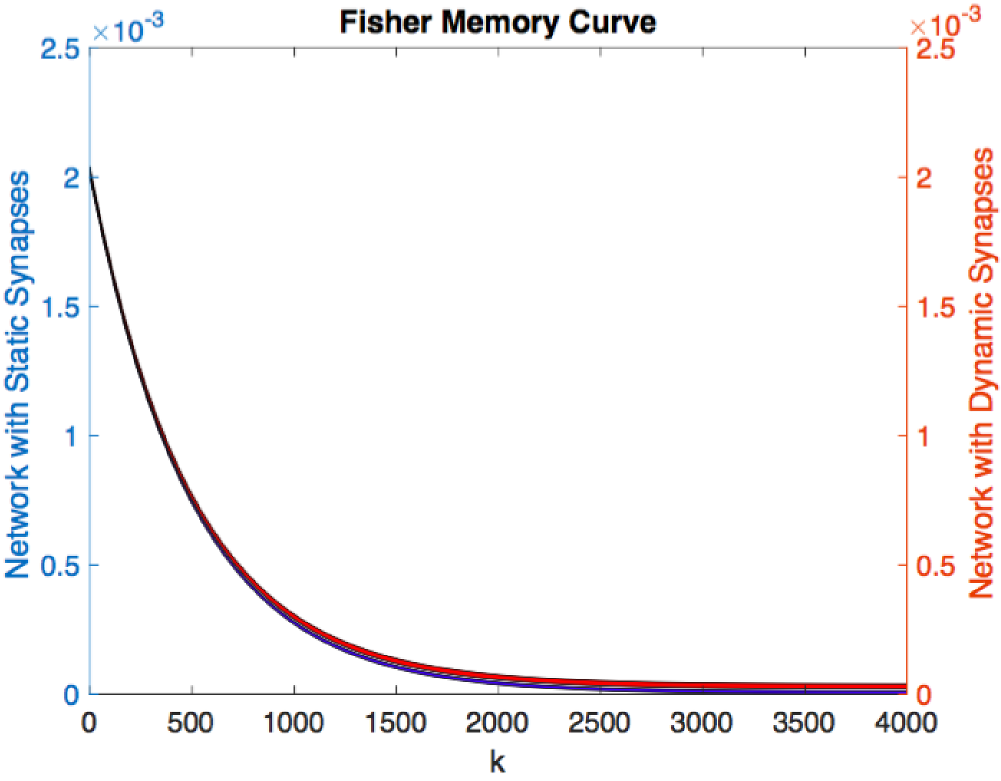

Figure 4.16: Fisher Memory Curve : Orthogonal Network with dynamic synapses (blue) and without dynamic synapses (blue)

The TM of the orthogonal network with or without dynamic synapses is equal to one, which is the bound for normal networks.

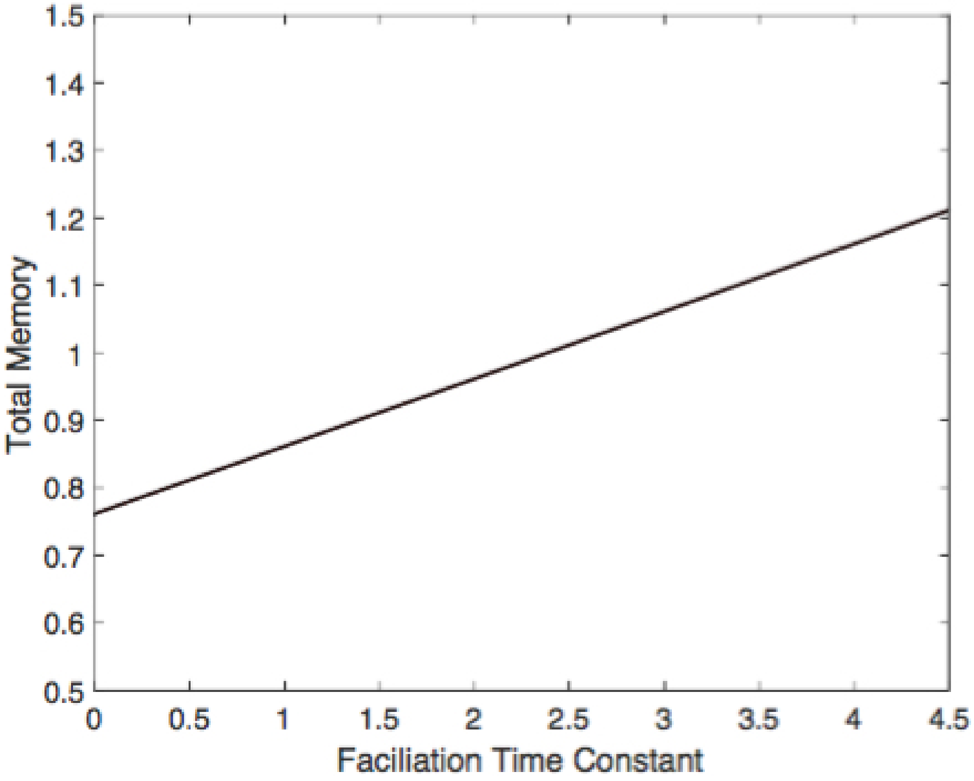

Figure 4.17: Total Memory with varying τ_f : 100 Orthogonal Networks

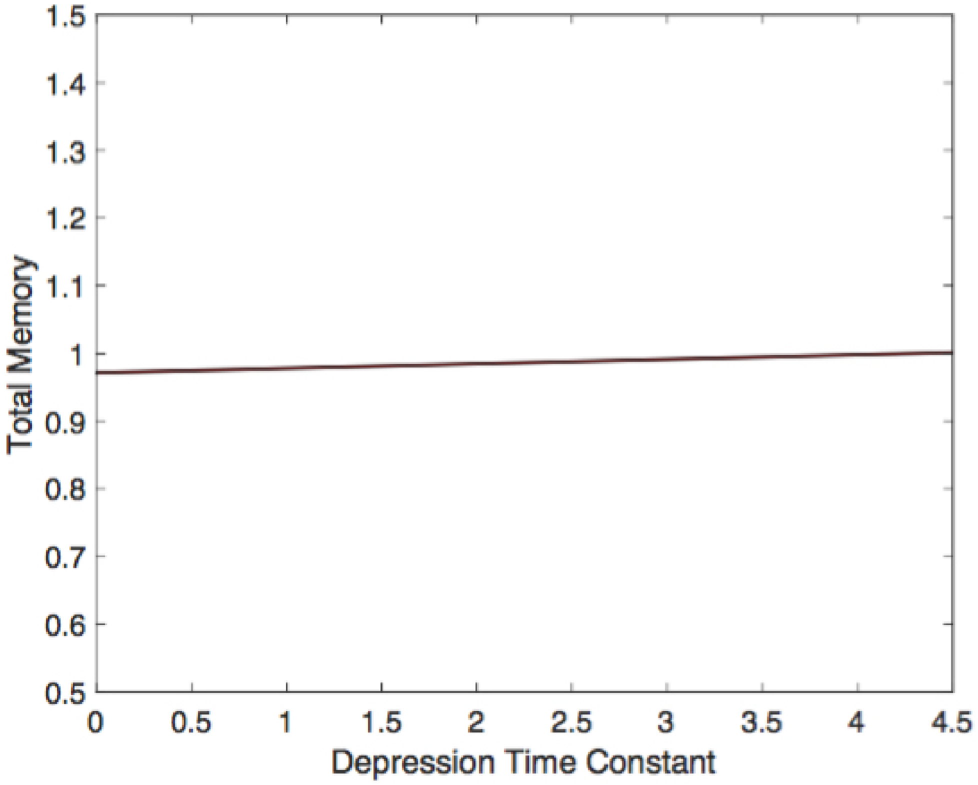

Figure 4.18: Total Memory with varying τ_d : 100 Orthogonal Networks

The results of the analysis of this connectivity matrix are very interesting. They show that dynamic synapses cause an improvement in memory performance only in certain cases. This result opens the door for further questions, which will be discussed in the conclusions section.

4.6.3 Total Memory

Now, we show a graph with the Total Memory achieved by all networks when dynamic synapses are added. Error bars correspond to the standard deviation from the mean from 100 networks. We showed how dynamic synapses change the connectivity of normal matrices in a way that it changes its hidden feed-forward structure, and allows them to have a higher memory capacity. Now, we show the Total Memory achieved by each of the two connectivities discussed (Symmetric and Orthogonal) with dynamic synapses. We showed the average results and standard deviation (error bars) in Figure 4.19.

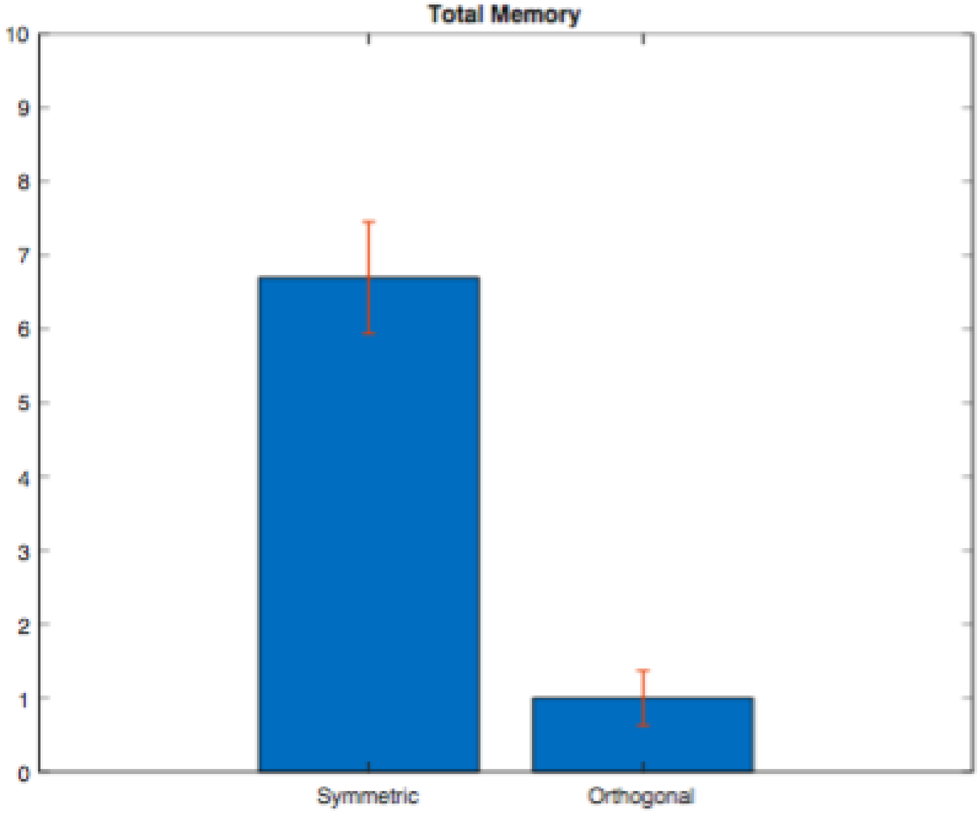

Figure 4.19: Total Memory achieved by Symmetric and Orthogonal connectivity matrices with dynamic synapses

As we can see, memory performance is better for the Symmetric network, even if it is not extensive. Extensive memory as was defined earlier is memory capacity proportional to the number of neurons in the network. We also saw in this chapter that the Orthogonal network was not affected by the addition of dynamic synapses, and consequently its memory capacity is that of a normal matrix and equal to one.

In Figures 4.20, and 4.21 we show how Total Memory scales as the number of neurons in the network increases.

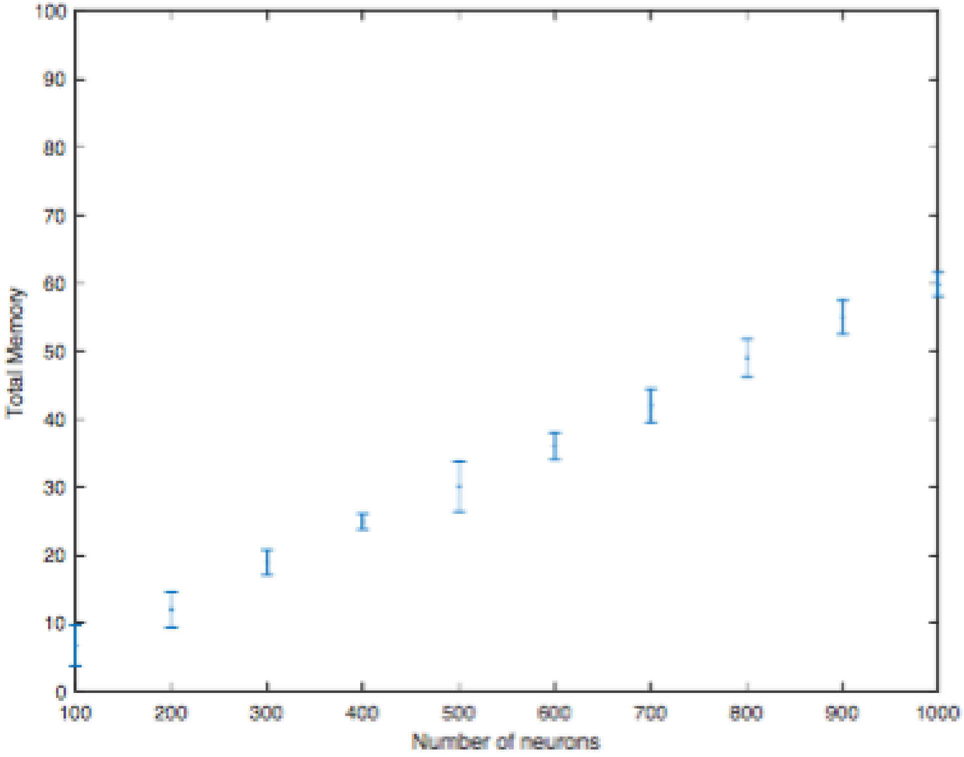

Figure 4.20: Total Memory as a function of the number of neurons in the network (Symmetric Network)

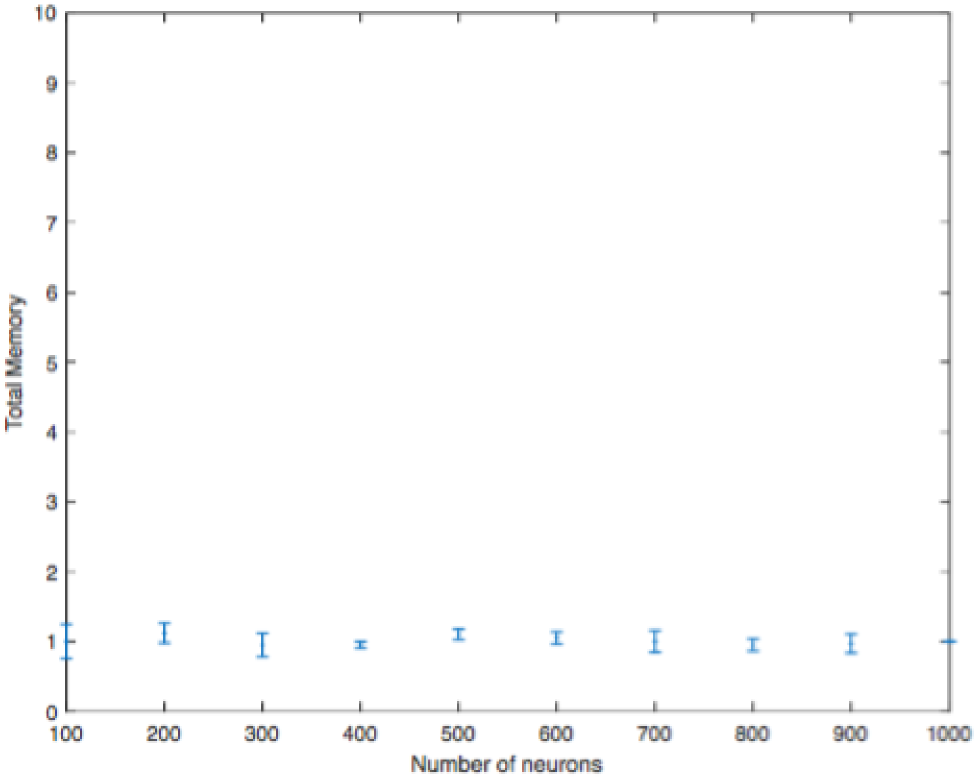

Figure 4.21: Total Memory Scaling as a function of the number of neurons in the network (Orthonormal Network)

4.7 Conclusions

One of the most remarkable characteristics of the human brain is its ability to hold information in short-term memory for use in prospective decision-making and action-selection. The current and most widely accepted hypothesis regarding the mechanistic implementation of working memory is persistent activity of neurons in cortical networks. However, the notion that this hypothesis is the sole neural substrate of working memory has been challenged [Hempel et al., 2000] [Mongillo et al., 2008] [Stokes, 2015] [Maass and Markram, 2002]. Persistent activity models have focused on storing discrete stimulus in specific working memory tasks; such as, the delayed matching task [White et al., 2004] [Toyoizumi, 2012]. In these models, each input is stored as a unique pattern of activity, and the presentation of a new input disrupts this memory. Stokes [Stokes, 2015] showed that neural circuits can overcome this disruption by extending the neural substrate of working memory from persistent activity in neurons to activity in hidden states, which he hypothesized were dynamic synapses. Furthermore, working memory tasks often require that items are stored and recalled sequentially. These sequential working memory tasks require the reconstruction of a whole dynamic sequence of inputs after a delay; for which standard persistent activity models are insufficient due to the interference of each input in the sequence [Toyoizumi, 2012]. In this chapter, we leveraged the fact that generic recurrent networks have been suggested as memory buffers [Maass et al., 2002] [Jaeger, 2001], and used a generic linear recurrent network endowed with dynamic synapses as a model of working memory.

Synapses between neurons are not static, rather they are composed of many membrane proteins with diverse temporal dynamics in terms of membrane trafficking, molecular structure and excitability. It has been proposed that this dynamic behaviour plays an important role not only in long term memory [Fusi et al., 2005], but also in working memory. In this chapter, we used a linear recurrent neural network, where synaptic strength is not static, but rather has its own intrinsic dynamics: short term facilitation and short term depression. Inspired by the proposals of [Stokes, 2015], [Maass and Markram, 2002] and [Mongillo et al., 2008], which regard synapses as active neural substrates of working memory, we tested whether dynamic synapses could change the effective connectivity of networks and increase their memory capacity.

We used previous work regarding the limits of memory capacity in linear recurrent neural circuits, and focused our analysis on networks with poor memory capacity (normal networks). In our analysis, we tested whether the temporal dynamics of synapses could change the effective connectivity of these network to have a hidden feed-forward structure. This structure is a pre-requisite for an effective network to have supra-linear transient amplification, which is necessary for a network to have a memory capacity of order higher than 1. We showed that dynamic synapses can change the structure of symmetric networks to that of anti-symmetric networks, which have previously been shown to have a significantly greater amplification profile compared to that of symmetric networks [Li and Dayan, 1999]. This result shows that neurons and synapses (endowed with their own dynamics) are capable of positively modifying neural networks. It is for future work to understand whether including more dynamic synapses, with a wider range of time constants, could further improve memory capacity. Particularly, it would be interesting to study the properties of real neural connectivities, and explore how they change with the addition of more dynamic variables. From a theoretical point of view, the connection between eigenspectrum modifications and changes in the networks' dynamics is poorly understood - outside the cannonical cases. It would be useful to have this understanding, so we can have a better intuition of how changes in network structure lead to higher memory performance. Then, we can guide our research towards understanding how dynamic synapses can improve network structure.

Our results also suggest that we need to understand the conditions under which extensive memory, memory capacity proportional to the number of neurons, can be achieved. The symmetric network achieved a higher memory capacity when dynamic synapses were added; however, it did not achieve extensive memory. To evaluate this result, it is important to remember that the theoretical prediction for non-normal matrices is that they are capable of achieving extensive memory (memory capacity proportional to the number of neurons). There is no guarantee that they will achieve it. In order to achieve this bound, the network needs to exhibit transient supra-linear amplification, and this amplification of the signal needs to be higher than the noise amplification. It was shown that the delay line, and matrices unitarily equivalent to it, are the ones that can achieve extensive memory or memory of order proportional to the number of neurons in the network [Ganguli et al., 2008]. In our analysis, we saw that the amplification profile changed for the symmetric network when dynamic synapses were added. However, this amplification profile was not sufficient for the network to achieve extensive memory. The two main reasons for this are that it is unclear whether this particular amplification profile was sufficient to counteract the amplification of the noise. Our network has high values along the delay line (see its Schur Form plot), but it also has other elements connected in the network. These interactions cause noise through interference, and this noise is amplified every time step. Future work should focus on understanding the conditions that cause non-normal matrices to have a favorable transient amplification profile, and how a network can be driven closer to a delay line profile. This question is not straight forward. It requires a deep mathematical analysis of network theory beyond our current understanding. The second reason why this amplification profile was insufficient for the Symmetric network is because in our analysis, we only read the information stored in the neurons. The components of the effective network are both neurons and synapses; however, we only used the information available in the neurons. In the brain, only neurons can transmit information, so we wanted the comparison to be fair. We only measured the available information in the neurons, and not all the information that was amplified and available in the network. It is unclear how the brain uses information available in hidden variables, whether it has better decoding mechanisms to exploit information stored in synapses, or whether simply having this information readily available is sufficient to increase memory capacity. This point touches upon the topic of 'salient' working memory, which was discussed by [Stokes, 2015] and which implied that information hidden to a decoder is still present in other variables inside the network and it is useful to perform a task. Our results suggest that the neural substrates of working memory should be re-analyzed, and that we need to find a method for quantifying information present in all variables of a neural circuit.

Additionally, the results of this chapter give rise to new hypothesis regarding how the memory of discrete items in working memory and sequential memory can be understood by the same neural substrates. By including both neurons and synapses, we propose a more complete picture of working memory, where information can be stored and retrieved at different time scales. In this way, we not only reconcile the two opposing views described in this chapter, but we also suggest a mechanism for working memory of discrete items and sequential items. When an item is presented, and retained, the first mechanism to store this item is persistent activity. However, as time evolves, changes in synapses occur, and information is stored in these variables as well. The information stored in synapses can be used to mediate sequential memory tasks, as well as interruptions by other items like in Stoke's experiment without eliminating persistent activity as a neural substrate. Our proposal is to rather build on these models and include dynamic synapses in the picture. We have shown that dynamic synapses can change the effective network connectivity and improve memory capacity. However, we only used two dynamic synaptic variable, and the brain has multiple dynamic variables inside each neuron. By excluding all these variables in standard analysis, we are limiting our understanding of working memory. We believe that to fully understand the neural substrates and capacity limits of working memory, we need to introduce the rich temporal dynamics and biochemical cascades present inside each neuron in real neural circuits.

We hope that our results may be of particularly use in the field of artificial neural networks. So far, these networks make use of other structures, such as external memory buffers or gates, to learn temporal sequences. Otherwise, information is quickly lost due to the fast-forgetting of neurons and the chaotic dynamics of the network. Furthermore, even training these networks is difficult due to problems with back-propagation such as exploding and vanishing gradients. Different solutions have been proposed to these learning problems, partly based on the same analysis methods that we have used. For instance, it has been proposed that orthogonal or unitary connectivity matrices can be used to avoid these problems with training and can guarantee stable dynamics because they preserve Euclidean norms and ensure unit length eigenvalues [Kerg et al., 2019][Orhan and Pitkow, 2019]. Therefore, they prevent exponential growth or decay in long products of Jacobians used during gradient decent computations. However, these connectivity matrices have limited temporal expressivity, do not deal well with tasks that require online computations, and their norm preserving characteristic breaks down in the presence of noise or non-linearities [Orhan and Pitkow, 2019]. Orhan et. al suggests that for non-linear networks or for networks with noisy inputs, one should focus on the signal-to-noise ratio of the propagated input (how signal is amplified through the circuit compared to noise, and in these scenarios non-normal matrices outperform unitary matrices, as predicted by [Ganguli et al., 2008]. Furthermore, these authors showed that networks that were initialized with an orthogonal or unitary connectivity matrix, through training, changed their structure to non-normality and hidden chain-like feedforward [Orhan and Pitkow, 2019]. This result is aligned with our observations. They suggest that both biological and artificial learning mechanisms are capable of adapting their structure input and task requirements. We showed that this is possible with dynamic synapses and Orhan et. al showed that this is possible with training. Dynamic synapses (short-term plasticity) and neural network training have the same objective: modify the structure of the network to adapt to different tasks. It has been shown that this adaptation improves memory capacity, and therefore the mechanisms that allow for it (dynamic synapses) should be included as part of the neural substrate of working memory. The field connecting structure adaptation with computational performance is just emerging, but we believe that the way forward consists of incorporating more diversity and components from real neural circuits.

For temporal sequential memory, both biological learning and artificial learning rules might change the effective connectivity of the network to a feed-forward structure. Kerg et. al take this argument one step further and suggest a more flexible approach that uses both normal and non-normal structures and harness the advantages of both. They proposed a novel connectivity structure, called non-normal RNN (nnRNN), based on splitting of the Schur form into normal (diagonal) and non-normal (non-diagonal, feed-forward) parts. By adding a non-normal component, these recurrent networks can use the computational advantages of non-normality, such as transient expansion and compression and better expressivity to encode and transmit. In this way, on tasks well suited for unitary RNNs, nnRNNs learn a normal connectivity; and on tasks requiring temporal expressivity, they learn a non-normal connectivity. nnRNNs retain the stability advantages and training speed of unitary RNNs, while enhancing expressivity, especially on tasks that require computations over ongoing input sequences [Kerg et al., 2019]. This new proposal shows how training can adjust the connectivity structure of the network to increase performance, which is aligned with the proposal of this chapter that networks can flexibly adjust their connectivity matrices and that non-normal connectivity can arise when learning is activated in the synapses. Previous work [Hennequin et al., 2012], has also shown that recurrent networks can amplify signals by the presence of near-critical eigenvalues in the network connectivity matrix, or through the non-normality of this network. In particular, we showed that there exists a trade-off between non-normal amplification and the dynamical slowing caused by the near-critical eigenvalues of the network. Furthermore, it was shown that in order for transient amplification to occur with little slowing of dynamics, matrix connectivity must be structured. We showed that an anti-symmetric network can achieve this and argued that synaptic plasticity could be accounted for the shaping of network connectivity. The characteristics of network connectivity on biological neural circuits and whether they learn temporal inputs using a combination of normal and non-normal components remains an open question.

We have argue that understanding the neural substrates of working memory is closely linked to understanding its capacity limits and constraints. The physical structure of working memory is what ultimately limits its capacity. This understanding is essential to determine the computational constraints of almost all cognitive computations including reward-based decision-making [Collins and Frank, 2012]. Earlier in this thesis, we explored how episodic memory can be used in reinforcement learning. In our model, episodes are retrieved and integrated at the time of a decision. This implies that the retrieval and action-value computations associated with episodes rely on working memory. The capacity to retrieve more than one episode, and integrate this information adequately is highly dependent on working memory capacity. The retrieval of episodes based on contextual cues is meditated by working memory [Bornstein and Norman, 2017]. Furthermore, our model is based on a temporal context model of episodic memory TCM [Howes et al., 2009]. This model relies heavily on working memory because context is updated as a low pass filter of previous contexts held in working memory. The motivation of this chapter was to further our understanding of working memory capacity, so in the future we can develop more accurate algorithms of cognition. Future work on the mechanistic implementations of working memory will be key for the development and understanding of algorithmic and normative models of cognition.

Chapter 5

General Conclusions

In the previous sections, we described individual conclusions for each chapter. The goal of this section is to try to bring together the two separate analysis done in this thesis. For that, we put the analysis performed in this thesis into context. One of the most important goals in theoretical neuroscience is to understand how humans make rational computations to guide their actions and decisions in the face of uncertainty and biological constraints. To achieve this goal, it is hypothesized that humans make assumptions about their environment that allows them to guide their decisions accordingly. Specifically, humans infer the casual relationships of events, and use this information to make predictions regarding the consequences of actions. Then, if they are rational, they choose the one that gives them the highest reward. It is important to note that even though it would be optimal to perform exact inference of these causal relationships, human's cognitive capacity is limited and the real world is multivariate and complex; for that reason, it is assumed that the inference performed by humans is only approximate [Franklin et al., 2019] [Tenenbaum et al., 2011].

This work explored properties of two memory systems relevant for neural and cognitive computation: episodic memory and episodic memory. In the first section, we proposed a new framework for reward based learning based on episodic memory called Contextual Episodic. This framework recalls and weights episodes from the past based on their contextual similarity, and uses them to guide decisions. Furthermore, this framework can be combined with model-free cached values (Hybrid model), capturing the human ability to integrate multiple sources of information to make decisions. We showed that this model fits human data better than previous models, demonstrating the importance of contextually-sensitive episodic memory for decision-making. We also derived a generative model for which a particular form of episodic recall is required for optimal posterior inference (also known as recognition), and we proposed our episodic-based frameworks (Contextual Episodic and Hybrid models) as approximate inference mechanisms.In order to understand the computations performed by the brain, it is important to understand the structure of the natural world [Brunswik, 1955] [Simon, 1955]. Using our proposed generative model, we showed that there is a range of parameters for which episodes are more or less important, and demonstrated that the standard Rescorla-Wagner rule, the Contextual Episodic or the Hybrid model each works best in certain situations in terms of making more accurate inferences and predictions of future rewards. In particular, the Hybrid model collected the greatest reward when the environment consisted of continuous events interspersed with random events or repetitions of similar events from the past. The Hybrid model is the model that fits human data the best; for this reason, it might be said that human inference operates as if it expects to be in an environment with this property. That is, human inference assumes that events are generally temporal contiguous, but it is also ready for new situations or situations similar to the past to occur. It is for these types of situations that an episodic memory would be necessary, and where an episodic-based reinforcement learning algorithm can play an important role.

In the second section, we discussed another very important memory system for cognition: working memory. This memory system is a core component of many cognitive computations, as well as many of their theoretical proposals. Such is the case of our Contextual Episodic and Hybrid models. For this reason, understanding the mechanisms of working memory and its limits is necessary for developing constrained and accurate models of cognition. Because the definition of working memory and its neural substrates are poorly understood, in this section we explored the hypothesis that dynamic synapses are hidden, but active components of working memory [Stokes, 2015]. Using a simplified linear recurrent neural network model with dynamic synapses, we showed that they can play an important role at expanding the temporal capacity of working memory capacity. It is an interesting question for future work to understand whether this result is applicable to realistic neural connectivities, as well as understand the conditions under which dynamic synapses (or other dynamic elements inside the neuron) are active components during short term storage in the brain. Furthermore, this finding is important for the machine learning community. Artificial neural networks, which are based on recurrent networks with static synapses, suffer from poor temporal memory. LSTMs [Hochreiter and Schmidhuber, 1997], GRUs [Cho et al., 2014], among other architectures have been proposed as a way to train and buffer temporal information in artificial networks. Our work suggests that including other variables, with a diverse set of time constants, could be one way to harness the temporal power of recurrent networks.

As a final thought, this thesis has explored new ways to better understand working memory and episodic memory. Although we studied them rather separately, in fact they interact in two critical ways. First, working memory is likely the store for the episodes that are recalled through the process of decision-making. Thus, the capacity of working memory plays a central role in determining the quality of the contribution of episodes to choice. Second, the time constants of the evolution of context in the Temporal Context Model [Howard and Kahana, 2002] depend on the nature of working memory. It is therefore a pressing task for the future to study episodic and working memory conjointly, looking at how they realize good- and bad-quality decision-making.

Appendix A

Confusion Matrices (Artificial Data Analysis)

	RW	**TC**	**S**	**H**
RW	115.24	118.80	116.02	117.54
TC	116.80	114.56	116.72	118.07
S	115.98	117.20	112.89	113.56
H	117.14	119.82	114.01	112.02

Table A.1: Confusion Matrix Experiment 1

	RW	**TC**	**S**	**H**
RW	72.32	76.80	75.61	78.80
TC	77.20	74.93	76.45	79.57
S	78.90	81.05	71.05	72.49
H	88.79	92.20	76.79	69.72

Table A.2: Confusion Matrix Experiment 2

Appendix B

Model Selection for Contextual Episodic

In Chapter 2, we introduced the Contextual Episodic model, in which episodes stored are equal to rewards received. However, episodes stored could also be equal to the Q values of actions at each time step. In order to select between these two ways of storing episodes, we performed model selection between these two models. We tested two Contextual Episodic models, one where the stored values are Q values and another one where the stored values are rewards. We call the first model Contextual Episodic$_Q$ and the second one just Contextual Episodic. The table below shows the results of the BIC scores computed with human data from Experiment 2. The Contextual Episodic model with rewards as episodes out-performs the model with Q values as episodes. For this reason, we decided to use the value of rewards received as the episodes stored.

Models	**BIC Scores**
Contextual Episodic$_Q$	106.56
Contextual Episodic	103.85

This result suggests humans use a combination of techniques and sources of information to make decisions. Intuitively, it makes sense that humans would have a running average value of actions, and complement it with episodic recall if needed. At the time of decision, it would make the most sense to keep both sources of information, and make use of the one that appears more accurate or relevant at the time.

Appendix C

Weighting Functions

In this appendix, we show the plots of the recency-based and the TCM-based weighting functions. These functions were taken model fitting experiment 2 data. The plots show the weights for previous time step during the 162 time steps of the experiment.It can be seen that recency-based weighting function has an exponential decay, while TCM-based weighting function weights different items with higher weights.

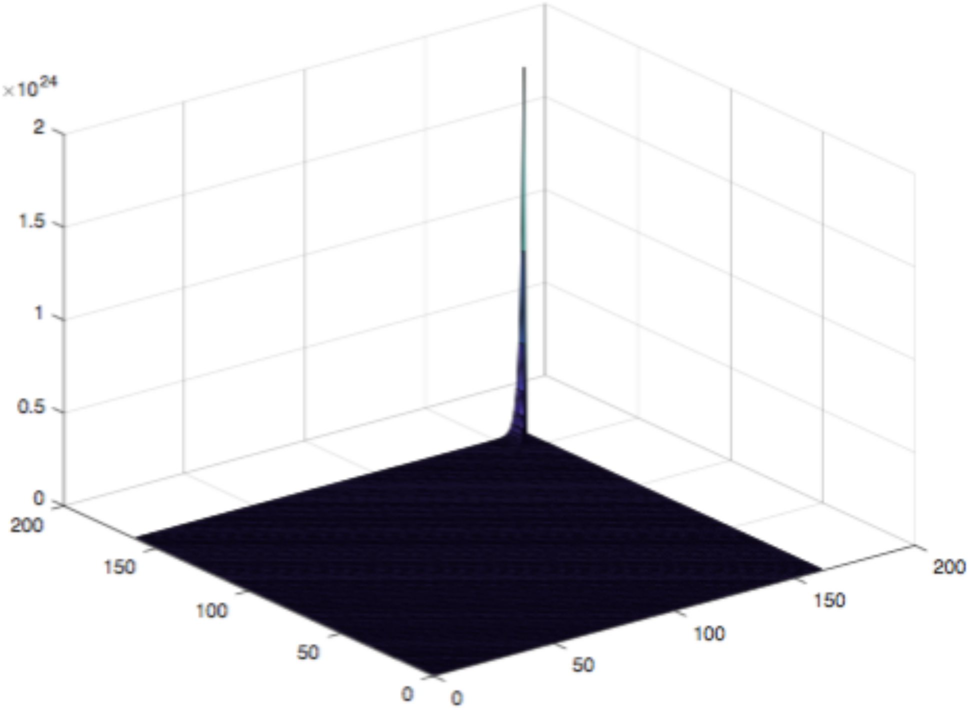

Figure C.1: Recency-based Weight

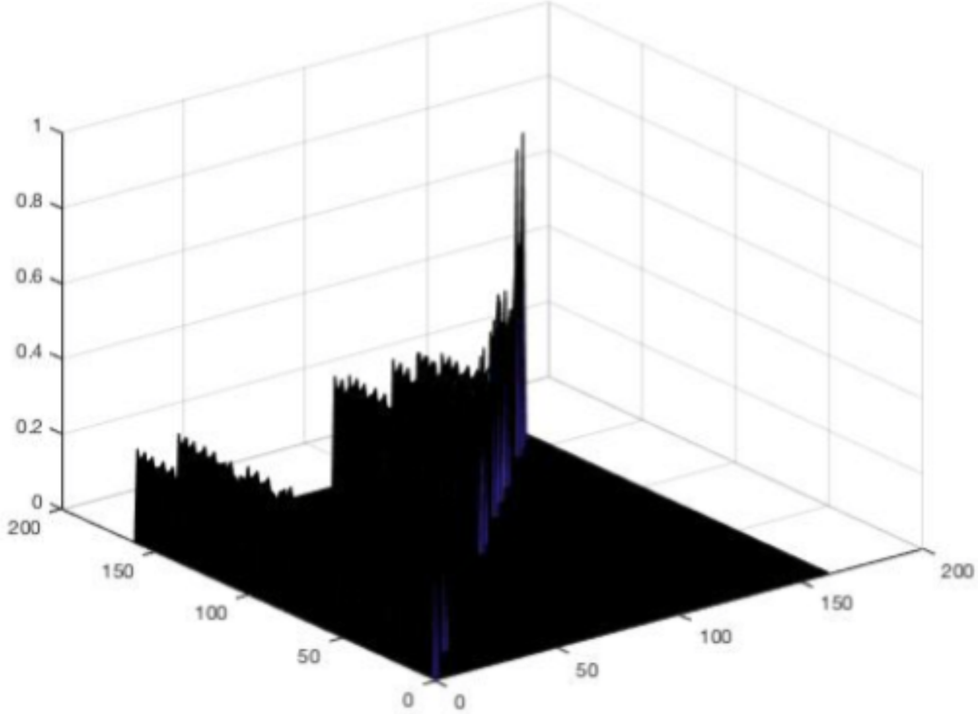

Figure C.2: TCM-based Weight

Appendix D

Fisher Information

Fisher information, which measures the amount of information an observed random variable carries about an unknown parameter of the distribution from which the observed variable is sampled. Formally, it is defined as the variance of the score (partial derivative with respect to parameter of the logarithm of the likelihood function). Since the expected value of the score is equal to zero [Ly et al., 2017]. To define Fisher information, we define a random variable y drawn from a distribution $P(y|)$. The Fisher information is equal to:

$$I(\theta) = E\left[\left(\frac{d}{d} logP(y|\theta)\right)^2\right] \tag{D.1}$$

The Fisher information can also be written as [Ly et al., 2017]:

$$I(\theta) = E\left[\left(-\frac{d^2}{d^2} logP(y|\theta)\right)\right] \tag{D.2}$$

The Fisher information is also used to give a lower bound on the variance of any unbiased estimator of a model's parameters. This bound is called the Cramér–Rao bound. Intuitively, the Fisher information tells us how much we can infer about the parameters of a distribution when we make an observation. The Fisher information tells us how steep or flat the distribution is. When the distribution is sharply peaked, the Fisher information is high and the variance of the estimator is low. Each observation carries a lot of information. On the other hand, when the distribution is flat, one observation is not very informative and the Fisher information is low [Ly et al., 2017]. Figure D.1 provides a visual explanation of this description, where $L()$ refers to the Likelihood function $L() = P(y|)$

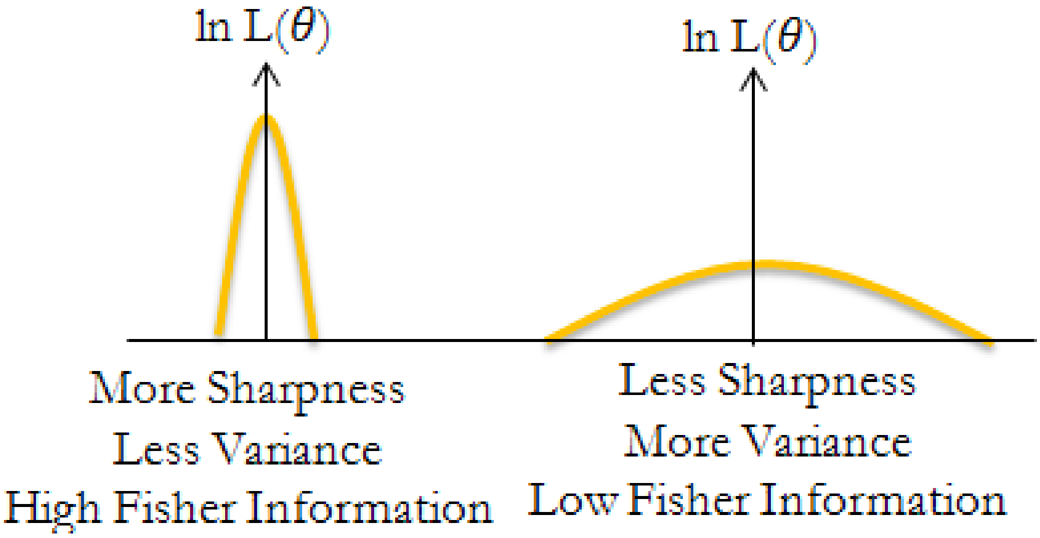

Figure D.1: Fisher Information

Appendix E

Linear and Linearized Networks Response to input pulse

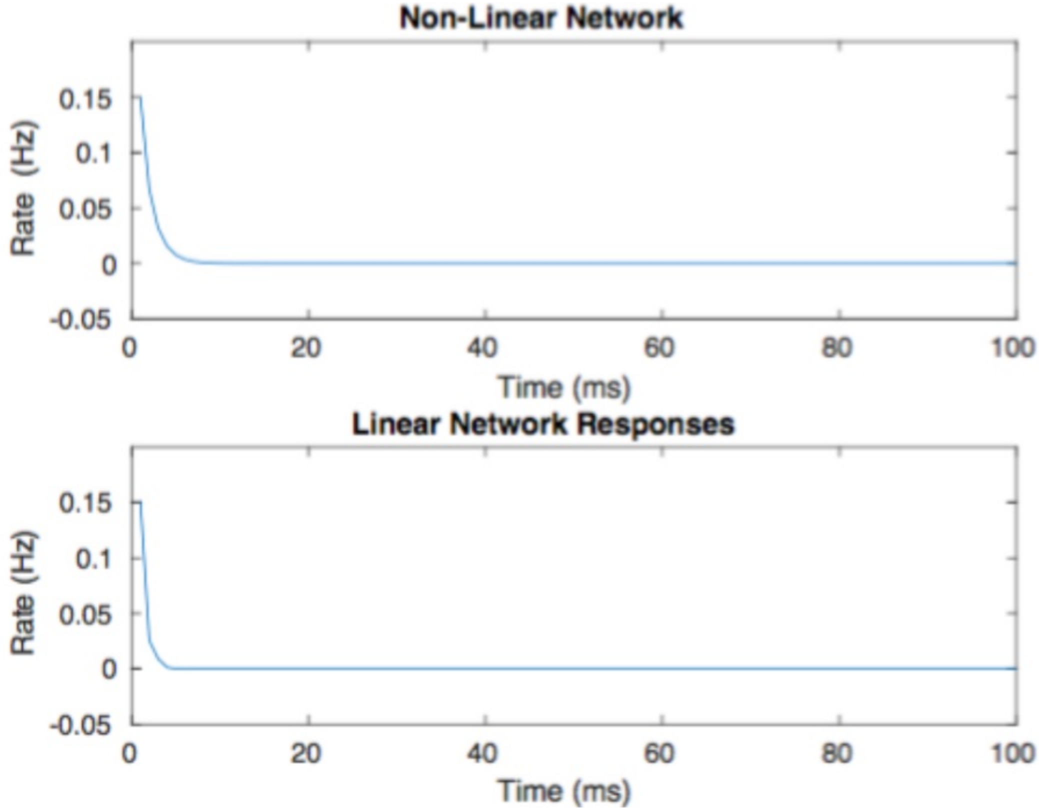

Figure E.1: Networks Dynamics: Linear and Non-linear Networks' response to a pulse input

Appendix F

Schur Form Symmetric Network with dynamic synapses

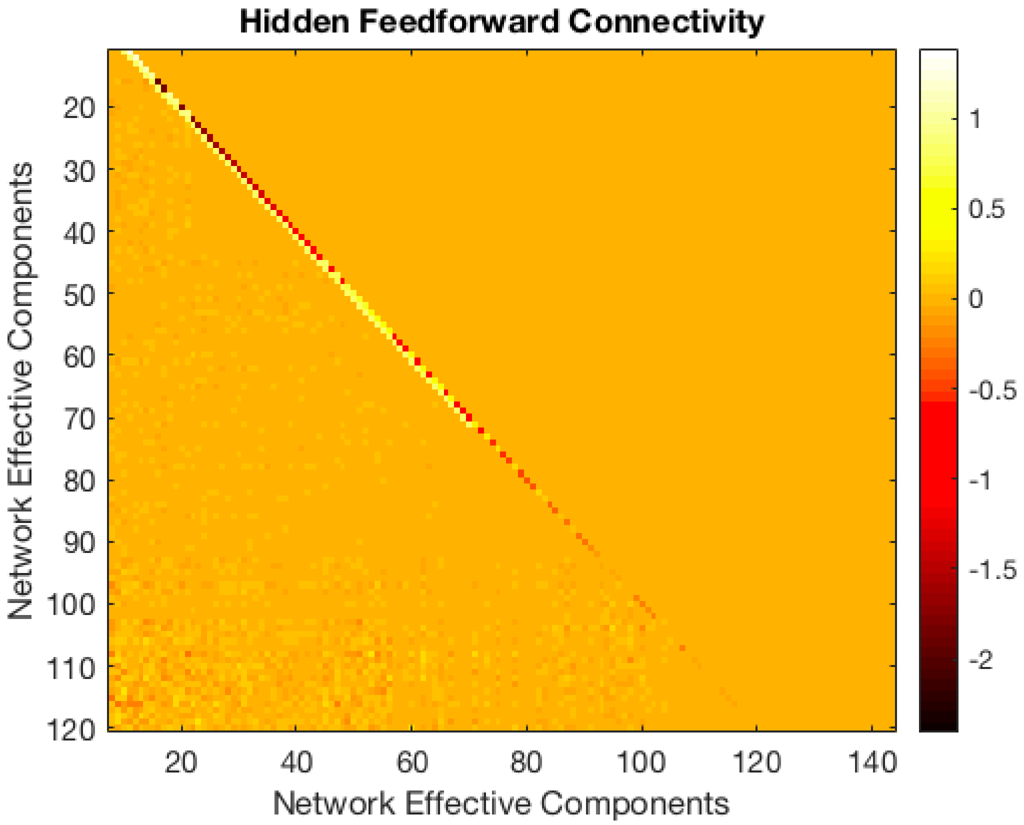

Figure F.1: Schur Decomposition: Symmetric Network with dynamic synapses: zoom version. Entries along the delay line can be seen

Bibliography

[Aben et al., 2012] Aben, B., Stapert, S., and Blokland, A. (2012). About the distinction between working memory and short-term memory. *Frontiers in psychology*, 3:301.

[Alexander and Fuster, 1973] Alexander, G. E. and Fuster, J. M. (1973). Effects of cooling prefrontal cortex on cell firing in the nucleus medialis dorsalis. *Brain research*, 61:93–105.

[Amit and Fusi, 1994] Amit, D. J. and Fusi, S. (1994). Learning in neural networks with material synapses. *Neural Computation*, 6(5):957–982.

[Angela and Dayan, 2005] Angela, J. Y. and Dayan, P. (2005). Uncertainty, neuromodulation, and attention. *Neuron*, 46(4):681–692.

[Asllani et al., 2018] Asllani, M., Lambiotte, R., and Carletti, T. (2018). Structure and dynamical behavior of non-normal networks. *Science advances*, 4(12):eaau9403.

[Baddeley, 2000] Baddeley, A. (2000). The episodic buffer: a new component of working memory? *Trends in cognitive sciences*, 4(11):417–423.

[Baddeley, 2012] Baddeley, A. (2012). Working memory: theories, models, and controversies. *Annual review of psychology*, 63:1–29.

[Baddeley and Hitch, 1974] Baddeley, A. D. and Hitch, G. (1974). Working memory. In *Psychology of learning and motivation*, volume 8, pages 47–89. Elsevier.

[Baddley et al., 2003] Baddley, J. W., Pappas, P. G., Smith, A. C., and Moser, S. A. (2003). Epidemiology of aspergillus terreus at a university hospital. *Journal of clinical Microbiology*, 41(12):5525–5529.

[Balleine, 2005] Balleine, B. W. (2005). Neural bases of food-seeking: affect, arousal and reward in corticostriatolimbic circuits. *Physiology & behavior*, 86(5):717–730.

[Beal et al., 2002] Beal, M. J., Ghahramani, Z., and Rasmussen, C. E. (2002). The infinite hidden markov model. In *Advances in neural information processing systems*, pages 577–584.

[Bernardo et al., 2007] Bernardo, J., Bayarri, M., Berger, J., Dawid, A., Heckerman, D., Smith, A., and West, M. (2007). Generative or discriminative? getting the best of both worlds. *Bayesian statistics*, 8(3):3–24.

[Blei et al., 2017] Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

[Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

[Bornstein et al., 2017] Bornstein, A. M., Khaw, M. W., Shohamy, D., and Daw, N. D. (2017). Reminders of past choices bias decisions for reward in humans. *Nature Communications*, 8:15958.

[Bornstein and Norman, 2017] Bornstein, A. M. and Norman, K. A. (2017). Reinstated episodic context guides sampling-based decisions for reward. *Nature Neuroscience*, 20(7):997.

[Brunswik, 1955] Brunswik, E. (1955). Representative design and probabilistic theory in a functional psychology. *Psychological review*, 62(3):193.

[Buonomano and Maass, 2009] Buonomano, D. V. and Maass, W. (2009). State-dependent computations: spatiotemporal processing in cortical networks. *Nature Reviews Neuroscience*, 10(2):113.

[Cho et al., 2014] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

[Collins and Frank, 2012] Collins, A. G. and Frank, M. J. (2012). How much of reinforcement learning is working memory, not reinforcement learning? a behavioral, computational, and neurogenetic analysis. *European Journal of Neuroscience*, 35(7):1024–1035.

[Corkin, 2002] Corkin, S. (2002). What's new with the amnesic patient hm? *Nature reviews neuroscience*, 3(2):153.

[Cowan, 1999] Cowan, N. (1999). An embedded-processes model of working memory. *Models of working memory: Mechanisms of active maintenance and executive control*, 20:506.

[Cowan, 2001] Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and brain sciences*, 24(1):87–114.

[Cowan, 2016] Cowan, N. (2016). *Working Memory Capacity: Classic Edition*. Routledge.

[Cowan et al., 2004] Cowan, N., Chen, Z., and Rouder, J. N. (2004). Constant capacity in an immediate serial-recall task: A logical sequel to miller (1956). *Psychological science*, 15(9):634–640.

[Cowan et al., 2000] Cowan, N., Nugent, L. D., Elliott, E. M., and Geer, T. (2000). Is there a temporal basis of the word length effect? a response to service (1998). *The Quarterly Journal of Experimental Psychology Section A*, 53(3):647–660.

[Daneman and Carpenter, 1980] Daneman, M. and Carpenter, P. A. (1980). Individual differences in working memory and reading. *Journal of verbal learning and verbal behavior*, 19(4):450–466.

[Dasgupta et al., 2017] Dasgupta, I., Schulz, E., and Gershman, S. J. (2017). Where do hypotheses come from? *Cognitive psychology*, 96:1–25.

[Daw et al., 2008] Daw, N. D., Courville, A. C., and Dayan, P. (2008). Semi-rational models of conditioning: The case of trial order. *The probabilistic mind*, pages 431–452.

[Daw et al., 2005] 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.

[Dayan et al., 2003] Dayan, P., Abbott, L., et al. (2003). Theoretical neuroscience: computational and mathematical modeling of neural systems. *Journal of Cognitive Neuroscience*, 15(1):154–155.

[Dayan and Kakade, 2001] Dayan, P. and Kakade, S. (2001). Explaining away in weight space. In *Advances in neural information processing systems*, pages 451–457.

[Dayan et al., 2000] Dayan, P., Kakade, S., and Montague, P. R. (2000). Learning and selective attention. *Nature neuroscience*, 3(11s):1218.

[Dayan and Niv, 2008] Dayan, P. and Niv, Y. (2008). Reinforcement learning: the good, the bad and the ugly. *Current opinion in neurobiology*, 18(2):185–196.

[D'Esposito, 2007] D'Esposito, M. (2007). From cognitive to neural models of working memory. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1481):761–772.

[Dickinson, 1980] Dickinson, A. (1980). *Contemporary animal learning theory*, volume 1. CUP Archive.

[Dickinson and Balleine, 2002] Dickinson, A. and Balleine, B. (2002). The role of learning in the operation of motivational systems. *Steven's handbook of experimental psychology: Learning, motivation and emotion*, 3:497–534.

[Dolan, 2007] Dolan, R. J. (2007). The human amygdala and orbital prefrontal cortex in behavioural regulation. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1481):787–799.

[Duncan and Shohamy, 2016] Duncan, K. D. and Shohamy, D. (2016). Memory states influence value-based decisions. *Journal of Experimental Psychology: General*, 145(11):1420.

[Durstewitz and Seamans, 2006] Durstewitz, D. and Seamans, J. (2006). Beyond bistability: biophysics and temporal dynamics of working memory. *Neuroscience*, 139(1):119–133.

[Durstewitz and Seamans, 2002] Durstewitz, D. and Seamans, J. K. (2002). The computational role of dopamine d1 receptors in working memory. *Neural Networks*, 15(4-6):561–572.

[Durstewitz et al., 2000] Durstewitz, D., Seamans, J. K., and Sejnowski, T. J. (2000). Neurocomputational models of working memory. *Nature neuroscience*, 3(11s):1184.

[D'Esposito et al., 2000] D'Esposito, M., Postle, B. R., and Rypma, B. (2000). Prefrontal cortical contributions to working memory: evidence from event-related fmri studies. In *Executive control and the frontal lobe: Current issues*, pages 3–11. Springer.

[Eichenbaum and Cohen, 2004] Eichenbaum, H. and Cohen, N. J. (2004). *From conditioning to conscious recollection: Memory systems of the brain*. Number 35. Oxford University Press on Demand.

[El-Arini, 2008] El-Arini, K. (2008). Dirichlet processes: a gentle tutorial. In *Select Lab Meeting*, volume 10.

[Ferguson, 1973] Ferguson, T. S. (1973). A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230.

[Finch, 2004] Finch, S. (2004). Ornstein-uhlenbeck process.

[Fox and Roberts, 2012] Fox, C. W. and Roberts, S. J. (2012). A tutorial on variational bayesian inference. *Artificial intelligence review*, 38(2):85–95.

[Franklin et al., 2019] Franklin, N., Norman, K. A., Ranganath, C., Zacks, J. M., and Gershman, S. J. (2019). Structured event memory: a neuro-symbolic model of event cognition. *BioRxiv*, page 541607.

[Funahashi et al., 2004] Funahashi, S., Takeda, K., and Watanabe, Y. (2004). Neural mechanisms of spatial working memory: contributions of the dorsolateral prefrontal cortex and the thalamic mediodorsal nucleus. *Cognitive, Affective, & Behavioral Neuroscience*, 4(4):409–420.

[Fusi et al., 2005] Fusi, S., Drew, P. J., and Abbott, L. F. (2005). Cascade models of synaptically stored memories. *Neuron*, 45(4):599–611.

[Fuster and Alexander, 1971] Fuster, J. M. and Alexander, G. E. (1971). Neuron activity related to short-term memory. *Science*, 173(3997):652–654.

[Ganguli et al., 2008] Ganguli, S., Huh, D., and Sompolinsky, H. (2008). Memory traces in dynamical systems. *Proceedings of the National Academy of Sciences*, 105(48):18970–18975.

[Ganguli and Latham, 2009] Ganguli, S. and Latham, P. (2009). Feedforward to the past: The relation between neuronal connectivity, amplification, and short-term memory. *Neuron*, 61(4):499–501.

[Geisler and Diehl, 2003] Geisler, W. S. and Diehl, R. L. (2003). A bayesian approach to the evolution of perceptual and cognitive systems. *Cognitive Science*, 27(3):379–402.

[Gershman and Daw, 2017] 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 et al., 2015] Gershman, S. J., Horvitz, E. J., and Tenenbaum, J. B. (2015). Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science*, 349(6245):273–278.

[Ghahramani, 2010] Ghahramani, Z. (2010). Bayesian hidden markov models and extensions. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 56–56. Association for Computational Linguistics.

[Gilboa and Schmeidler, 2001] Gilboa, I. and Schmeidler, D. (2001). *A theory of case-based decisions*. Cambridge University Press.

[Gobet and Simon, 2000] Gobet, F. and Simon, H. A. (2000). Five seconds or sixty? presentation time in expert memory. *Cognitive Science*, 24(4):651–682.

[Goedeke and Diesmann, 2008] Goedeke, S. and Diesmann, M. (2008). The mechanism of synchronization in feed-forward neuronal networks. *New Journal of Physics*, 10(1):015007.

[Hempel et al., 2000] Hempel, C. M., Hartman, K. H., Wang, X.-J., Turrigiano, G. G., and Nelson, S. B. (2000). Multiple forms of short-term plasticity at excitatory synapses in rat medial prefrontal cortex. *Journal of neurophysiology*, 83(5):3031–3041.

[Hennequin et al., 2012] Hennequin, G., Vogels, T. P., and Gerstner, W. (2012). Non-normal amplification in random balanced neuronal networks. *Physical Review E*, 86(1):011909.

[Hertz and Prügel-Bennett, 1996] Hertz, J. and Prügel-Bennett, A. (1996). Learning short synfire chains by self-organization. *Network: Computation in Neural Systems*, 7(2):357–363.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Hopfield, 1982] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558.

[Howard and Kahana, 2002] Howard, M. W. and Kahana, M. J. (2002). A distributed representation of temporal context. *Journal of Mathematical Psychology*, 46(3):269–299.

[Howes et al., 2009] Howes, A., Lewis, R. L., and Vera, A. (2009). Rational adaptation under task and processing constraints: Implications for testing theories of cognition and action. *Psychological review*, 116(4):717.

[Jaeger, 2001] Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13.

[Jäkel et al., 2009] Jäkel, F., Schölkopf, B., and Wichmann, F. A. (2009). Does cognitive science need kernels? *Trends in cognitive sciences*, 13(9):381–388.

[Jordán et al., 1999] Jordán, F., Takács-Sánta, A., and Molnár, I. (1999). A reliability theoretical quest for keystones. *Oikos*, pages 453–462.

[Kalman, 1960] Kalman, R. E. (1960). On the general theory of control systems. In *Proceedings First International Conference on Automatic Control, Moscow, USSR*.

[Keramati et al., 2011] Keramati, M., Dezfouli, A., and Piray, P. (2011). Speed/accuracy trade-off between the habitual and the goal-directed processes. *PLoS computational biology*, 7(5):e1002055.

[Kerg et al., 2019] Kerg, G., Goyette, K., Touzel, M. P., Gidel, G., Vorontsov, E., Bengio, Y., and Lajoie, G. (2019). Non-normal recurrent neural network (nnrnn):

learning long time dependencies while improving expressivity with transient dynamics. *arXiv preprint arXiv:1905.12080.*

[Killcross and Coutureau, 2003] Killcross, S. and Coutureau, E. (2003). Coordination of actions and habits in the medial prefrontal cortex of rats. *Cerebral cortex*, 13(4):400–408.

[Lahiri and Ganguli, 2013] Lahiri, S. and Ganguli, S. (2013). A memory frontier for complex synapses. In *Advances in neural information processing systems*, pages 1034–1042.

[Lengyel and Dayan, 2008] Lengyel, M. and Dayan, P. (2008). Hippocampal contributions to control: the third way. In *Advances in neural information processing systems*, pages 889–896.

[Li and Dayan, 1999] Li, Z. and Dayan, P. (1999). Computational differences between asymmetrical and symmetrical networks. In *Advances in Neural Information Processing Systems*, pages 274–280.

[Lieder and Griffiths, 2019] Lieder, F. and Griffiths, T. L. (2019). Resource-rational analysis: understanding human cognition as the optimal use of limited computational resources. *Behavioral and Brain Sciences*, pages 1–85.

[Lundqvist et al., 2016] Lundqvist, M., Rose, J., Herman, P., Brincat, S. L., Buschman, T. J., and Miller, E. K. (2016). Gamma and beta bursts underlie working memory. *Neuron*, 90(1):152–164.

[Ly et al., 2017] Ly, A., Marsman, M., Verhagen, J., Grasman, R. P., and Wagenmakers, E.-J. (2017). A tutorial on fisher information. *Journal of Mathematical Psychology*, 80:40–55.

[Maass and Markram, 2002] Maass, W. and Markram, H. (2002). Synapses as dynamic memory buffers. *Neural Networks*, 15(2):155–161.

[Maass et al., 2002] Maass, W., Natschläger, T., and Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–2560.

[Mackintosh, 1983] Mackintosh, N. J. (1983). *Conditioning and associative learning*. Clarendon Press Oxford.

[Marr and Poggio, 1976] Marr, D. and Poggio, T. (1976). From understanding computation to understanding neural circuitry.

[Marr et al., 1991] Marr, D., Willshaw, D., and McNaughton, B. (1991). Simple memory: a theory for archicortex. In *From the Retina to the Neocortex*, pages 59–128. Springer.

[Matsumoto and Tanaka, 2004] Matsumoto, K. and Tanaka, K. (2004). The role of the medial prefrontal cortex in achieving goals. *Current opinion in neurobiology*, 14(2):178–185.

[Mayo and Crockett, 1964] Mayo, C. W. and Crockett, W. H. (1964). Cognitive complexity and primacy-recency effects in impression formation. *The Journal of Abnormal and Social Psychology*, 68(3):335.

[Metcalfe et al., 2013] Metcalfe, A. W., Ashkenazi, S., Rosenberg-Lee, M., and Menon, V. (2013). Fractionating the neural correlates of individual working memory components underlying arithmetic problem solving skills in children. *Developmental Cognitive Neuroscience*, 6:162–175.

[Miller, 1956] Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81.

[Miyake and Shah, 1999] Miyake, A. and Shah, P. (1999). *Models of working memory: Mechanisms of active maintenance and executive control*. Cambridge University Press.

[Mongillo et al., 2008] Mongillo, G., Barak, O., and Tsodyks, M. (2008). Synaptic theory of working memory. *Science*, 319(5869):1543–1546.

[Ng and Jordan, 2002] Ng, A. Y. and Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*, pages 841–848.

[O'Doherty et al., 2003] O'Doherty, J. P., 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.

[O'Reilly and Frank, 2006] O'Reilly, R. C. and Frank, M. J. (2006). Making working memory work: a computational model of learning in the prefrontal cortex and basal ganglia. *Neural computation*, 18(2):283–328.

[Orhan and Pitkow, 2019] Orhan, A. E. and Pitkow, X. (2019). Improved memory in recurrent neural networks with sequential non-normal dynamics. *arXiv preprint arXiv:1905.13715*.

[Ormerod and Wand, 2010] Ormerod, J. T. and Wand, M. P. (2010). Explaining variational approximations. *The American Statistician*, 64(2):140–153.

[Pearce and Hall, 1980] 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.

[Plonsky et al., 2015] Plonsky, O., Teodorescu, K., and Erev, I. (2015). Reliance on small samples, the wavy recency effect, and similarity-based learning. *Psychological review*, 122(4):621.

[Poldrack et al., 2001] Poldrack, R. A., Clark, J., Paré-Blagoev, E. a., Shohamy, D., Moyano, J. C., Myers, C., and Gluck, M. A. (2001). Interactive memory systems in the human brain. *Nature*, 414(6863):546.

[Polyn et al., 2009] Polyn, S. M., Norman, K. A., and Kahana, M. J. (2009). A context maintenance and retrieval model of organizational processes in free recall. *Psychological review*, 116(1):129.

[Poole et al., 2017] Poole, B., Zenke, F., and Ganguli, S. (2017). Intelligent synapses for multi-task and transfer learning.

[Rescorla et al., 1972] Rescorla, R. A., Wagner, A. R., et al. (1972). A theory of pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. *Classical conditioning II: Current research and theory*, 2:64–99.

[Rushworth and Behrens, 2008] Rushworth, M. F. and Behrens, T. E. (2008). Choice, uncertainty and value in prefrontal and cingulate cortex. *Nature neuroscience*, 11(4):389.

[Schacter and Addis, 2009] Schacter, D. L. and Addis, D. R. (2009). On the nature of medial temporal lobe contributions to the constructive simulation of future events. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1521):1245–1253.

[Schwarz et al., 1978] Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

[Seamans et al., 2001] Seamans, J. K., Durstewitz, D., Christie, B. R., Stevens, C. F., and Sejnowski, T. J. (2001). Dopamine d1/d5 receptor modulation of excitatory synaptic inputs to layer v prefrontal cortex neurons. *Proceedings of the National Academy of Sciences*, 98(1):301–306.

[Service, 1998] Service, E. (1998). The effect of word length on immediate serial recall depends on phonological complexity, not articulatory duration. *The Quarterly Journal of Experimental Psychology: Section A*, 51(2):283–304.

[Shahbazi et al., 2016] Shahbazi, R., Raizada, R., and Edelman, S. (2016). Similarity, kernels, and the fundamental constraints on cognition. *Journal of Mathematical Psychology*, 70:21–34.

[Sherry and Schacter, 1987] Sherry, D. F. and Schacter, D. L. (1987). The evolution of multiple memory systems. *Psychological review*, 94(4):439.

[Simon, 1955] Simon, H. A. (1955). A behavioral model of rational choice. *The quarterly journal of economics*, 69(1):99–118.

[Socher et al., 2009] Socher, R., Gershman, S., Sederberg, P., Norman, K., Perotte, A. J., and Blei, D. M. (2009). A bayesian analysis of dynamics in free recall. In *Advances in neural information processing systems*, pages 1714–1722.

[Squire et al., 1984] Squire, L. R., Cohen, N. J., and Nadel, L. (1984). The medial temporal region and memory consolidation: A new hypothesis. *Memory consolidation: Psychobiology of cognition*, pages 185–210.

[Stokes, 2015] Stokes, M. G. (2015). 'activity-silent'working memory in prefrontal cortex: a dynamic coding framework. *Trends in cognitive sciences*, 19(7):394–405.

[Strogatz, 1994] Strogatz, S. H. (1994). Nonlinear dynamics and chaos: with applications to physics. *Biology, Chemistry, and Engineering (Studies in Nonlinearity), Perseus, Cambridge, UK.*

[Südhof, 1995] Südhof, T. C. (1995). The synaptic vesicle cycle: a cascade of protein–protein interactions. *Nature*, 375(6533):645.

[Sutton, 1988] Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44.

[Sutton, 1992] Sutton, R. S. (1992). Gain adaptation beats least squares. In *Proceedings of the 7th Yale workshop on adaptive and learning systems*, volume 161168.

[Sutton and Barto, 1998] Sutton, R. S. and Barto, A. G. (1998). *Introduction to reinforcement learning*, volume 135. MIT press Cambridge.

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

[Talmi et al., 2019] Talmi, D., Lohnas, L. J., and Daw, N. D. (2019). A retrieved context model of the emotional modulation of memory. *Psychological review*, 126(4):455.

[Teh, 2010] Teh, Y. W. (2010). Dirichlet process. *Encyclopedia of machine learning*, pages 280–287.

[Teh et al., 2005] Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2005). Sharing clusters among related groups: Hierarchical dirichlet processes. In *Advances in neural information processing systems*, pages 1385–1392.

[Tenenbaum et al., 2011] Tenenbaum, J. B., Kemp, C., Griffiths, T. L., and Goodman, N. D. (2011). How to grow a mind: Statistics, structure, and abstraction. *science*, 331(6022):1279–1285.

[Tetzlaff et al., 2012] Tetzlaff, C., Kolodziejski, C., Markelic, I., and Wörgötter, F. (2012). Time scales of memory, learning, and plasticity. *Biological cybernetics*, 106(11-12):715–726.

[Toyoizumi, 2012] Toyoizumi, T. (2012). Nearly extensive sequential memory lifetime achieved by coupled nonlinear neurons. *Neural computation*, 24(10):2678–2699.

[Tulving, 1985] Tulving, E. (1985). How many memory systems are there? *American psychologist*, 40(4):385.

[Tulving and Schacter, 1990] Tulving, E. and Schacter, D. L. (1990). Priming and human memory systems. *Science*, 247(4940):301–306.

[Van Gael et al., 2008] Van Gael, J., Saatci, Y., Teh, Y. W., and Ghahramani, Z. (2008). Beam sampling for the infinite hidden markov model. In *Proceedings of the 25th international conference on Machine learning*, pages 1088–1095. ACM.

[Watkins, 1989] Watkins, C. J. C. H. (1989). Learning from delayed rewards.

[Welch et al., 1995] Welch, G., Bishop, G., et al. (1995). An introduction to the kalman filter.

[Wenger and Shing, 2016] Wenger, E. and Shing, Y. L. (2016). Episodic memory. In *Cognitive Training*, pages 69–80. Springer.

[White and McDonald, 2002] White, N. M. and McDonald, R. J. (2002). Multiple parallel memory systems in the brain of the rat. *Neurobiology of learning and memory*, 77(2):125–184.

[White et al., 2004] White, O. L., Lee, D. D., and Sompolinsky, H. (2004). Short-term memory in orthogonal neural networks. *Physical review letters*, 92(14):148102.

[Widrow and Hoff, 1960] Widrow, B. and Hoff, M. E. (1960). Adaptive switching circuits. Technical report, Stanford Univ Ca Stanford Electronics Labs.

[Wilhelm et al., 2013] Wilhelm, O., Hildebrandt, A. H., and Oberauer, K. (2013). What is working memory capacity, and how can we measure it? *Frontiers in psychology*, 4:433.

[Zenke et al., 2017] Zenke, F., Poole, B., and Ganguli, S. (2017). Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3987–3995. JMLR. org.