University of Nottingham

School of Mathematical Sciences

Reinforcement Learning Approaches to Rapid Hippocampal Place Learning

Charline Tessereau

A thesis submitted to the University of Nottingham for the degree of

Doctor of Philosophy

Abstract

The ability to successfully navigate the physical environment is a vital skill for numerous species, including humans, to find food and shelter and remember how to return to important locations. As environments are inherently variable, brains have evolved amazing capabilities to adapt to various new situations. In particular, animals and humans have the ability to return to specific locations based on as few as a single experience. The mechanisms underlying behavioural flexibility in spatial navigation is the focus of ongoing research with repercussions in behavioural sciences, neurosciences, and artificial intelligence.

In particular, the field of Reinforcement Learning (RL), which investigates how an organism, virtual or living, learns to generate actions based on the reception of rewards, has been extremely active since the 1970s for the exploration of the mechanisms of flexibility underlying decision making. In parallel, neuroscience has also significantly advanced in uncovering the neural basis underlying spatial navigation mechanisms, for example with the discovery of neurons underlying the computation of cognitive maps [O'Keefe and Dostrovsky, 1971, Hafting et al., 2005], an internal representation of space. Past RL models design relies on representations that do not allow efficient flexibility in spatial navigation. However, models provide a theoretical framework that influences the interpretation of neural recordings. As recent recording technologies enable experimentalists to target an increasing number of neurons, there is a compelling need to develop new RL computational approaches for flexible spatial navigation, in particular to bridge the gap between neural population recordings and the production of behaviours.

In this thesis, I consider RL approaches in which the known coding properties of the cognitive map are used as a basis to perform spatial navigation. Specifically, I investigate computational ideas which enable agents to be more flexible in virtual spatial navigation scenarios. In particular, this thesis focuses on the Morris watermaze, an experimental apparatus in which rodents have to find a hidden platform within a pool of cloudy water. Rapid place learning in the Morris watermaze, demonstrated by rodents requiring only one exposure to a new platform location to subsequently be able to retrieve its position, is an example of flexibility in spatial navigation. I present different RL-based architectures which generate flexible behaviours in a virtual watermaze equivalent, and compare them to behavioural observations. I discuss both the similarity in behavioural performance (*i.e.*, how well they reproduce behavioural measures of rapid place learning) and neurobiological realism (*i.e.*, how well they map to neurobiological substrates involved in rapid place learning).

I propose distinct biologically realistic computational properties which enable an agent to be more flexible towards changes in goal locations. Behavioural flexibility requires hierarchical and generalisable representations for flexible transfer of knowledge. Hierarchical control is useful to generalise action chains, such as selecting a trajectory, to fulfil different purposes, such as reaching different goal locations. It also enables the adjustment of ongoing behaviour to unforeseen situations, for example, adapting to misprediction of the goal's location. Continuous encoding of space, action and time, permits smoother control and generalisation of experience, and removes the constraints caused by the choice of the representation's granularity. Neural networks in which connections between neurons reflect predictions about most likely future scenarios enable efficient planning of trajectories to adapt to novel situations. In a nutshell, flexibility requires efficient representations, and this thesis contributes to the investigation of their neural implementations.

Acknowledgements

I give thanks to my supervisors, Steve, Reuben, and Tobias, for their ongoing support. Thank you for guiding me in the first steps of my academic life, and for leaving me the space and plenty of freedom to explore and develop my own scientific interests.

I have had the chance to benefit from the energy of a new research group forming whilst doing my PhD. Thanks to Mark Humphries and his group for accepting me in their weekly meetings and for providing a space for interesting scientific discussions. Thanks to Mark van Rossum for your help, kindness, and social enthusiasm, and to his group for interesting scientific discussions around journal clubs.

Thank you to my partner, Georg, for helping keep the focus on what I have to do, and for being present and supportive. Thanks to my Comrade Ruth, the probably best roommate that I could live with during a period of thesis writing. I cherish and will never forget the unprecedented mutual support and organised collaboration that we have had in this house during this hard period of pandemic. Thank you to Vianney, for your unconditional love and support, and to have accompanied me in this period of life transition.

A special thanks to the amazing friends that have read chapters of this manuscript for typos and English advice. Namely, thanks to Josh, Jacob, Dan, Robin, Bill, Ruth, Georg, and Emily.

Thanks to the University of Nottingham for funding me for three years. Thanks to the Jacques Hadamard Mathematics Foundation, which funded my last master degree in Paris, opening the path of academia to me. A special thanks to Christophe Giraud for pointing me towards relevant persons in computational neurosciences.

Thanks to the persons that have inspired me and motivated my scientific curiosity at conferences and at work, some of whom became close friends. Thanks to the Imbizo team. A special thanks to Carlos, my Cosyne partner, for your understanding.

Thanks to my family, which I miss, for having shaped aspects of my personality that has led me to navigate overall interesting experiences from my point of view. Thanks to my mates from Nottingham for your good vibes and acceptance, to my friends from France for being always here with open arms when I visit and with whom connection remains, and thanks to the other beautiful connections that I have around the globe.

Publications

Parts of the work presented in chapters 4 and 6 of this thesis have been published in:

Charline Tessereau, Reuben O'Dea, Stephen Coombes, and Tobias Bast. "Reinforcement Learning approaches to hippocampus-dependent flexible spatial navigation". *Brain and Neuroscience Advances,* 5. 2021.

The work in chapter 7 of this thesis is being prepared for submission in:

Charline Tessereau, Tobias Bast, Reuben O'Dea, and Stephen Coombes. "Predictive representations for planning: relation between time, space, and efficiency". *In prep.*

Contents

Chapter 1

Introduction

Brains receive sensory information inputs and produce behavioural outputs. Information is transferred and transformed in order to guarantee a species' or individual's survival and prosperity. Although recent technological advances enable experimentalists to access *in-vivo* neural activity in different forms and from tens of thousands of neurons, interpreting this information to determine the underlying neural computations that support a particular behaviour is an ongoing challenge.

Computational neuroscience, which aims at elucidating how neural signalling is used in the brain to process information, has enabled major advances in inferring, reproducing and understanding the natural computations that underlie the transfer and transformation of information employed to produce a behaviour. In particular, Reinforcement Learning (RL) investigates how an animal or artificial agent learns from the reception of a reward. An iconic example of RL contribution to neurosciences is the demonstration that dopaminergic phasic activation in monkeys' brains resembles the reward prediction error in RL models [Schultz et al., 1997]. One exemplar success of RL development in artificial intelligence has enabled a computer program to beat a human grandmaster at GO [Silver et al., 2016].

Animals, including humans, show great behavioural flexibility. In particular, in a spatial navigation context, animals can return to specific locations based on as little as a single experience. In this thesis, I focus on the analysis and development of new RL approaches in computational neurosciences aiming at reproducing the flexibility shown by rodents in spatial navigation. The work presented here contributes to explaining the underlying neural mechanisms which support such behaviours.

Watermaze tasks, in which rats have to find a hidden platform in a pool of cloudy water surrounded by spatial cues, have long been used to study spatial navigation in the laboratory [Morris, 1981, Morris et al., 1982, Steele and Morris, 1999, Bast et al., 2009, Buckley and Bast, 2018]. Analogous tasks have been developed for human participants using virtual environments [Buckley and Bast, 2018]. Advances in experimental neurosciences [Bast et al., 2009] raise the need to revisit, combine, and synthesise existing spatial navigation models to achieve an enhanced multi-disciplinary understanding; and require new approaches to help us understand the computations underlying the flexibility shown by rodents and humans in a watermaze.

In this thesis, I investigate how flexibility can be generated using RL models in a virtual watermaze environment. I compare the behaviours of agents to those shown by rodents and humans in the watermaze. I discuss both the similarity in behavioural performance between agents, humans and rodents, *i.e.*, how well do they reproduce behavioural measures of rapid place learning, and the neurobiological realism of the models considered, *i.e.*, how well do they map to neurobiological substrates involved in rapid place learning.

Chapter 2 provides a review of experimental knowledge in the field of neuroscience of spatial navigation that will be relevant within this thesis. Spatial learning involves spatial memories organised within generalised representations, and the ability to make use of them to inform decision-making. In the watermaze, spatial learning is facilitated by the hippocampus [Morris et al., 1982], an area of the brain involved in memory and in which key spatial representations have been found [O'Keefe and Dostrovsky, 1971, Hafting et al., 2005]. Crucially, flexibility in the watermaze requires the hippocampus [Bast et al., 2009].

Chapter 3 presents general background notions to set the scene for RL [Sutton and Barto, 2018]. RL models have long been used to produce agents that can learn to perform certain actions or chains of actions based on the reception of a reward. The comparison of behaviours shown by agents with those of animals or humans enables the investigation of the mechanisms underlying decision-making. RL can be applied to spatial navigation, in which spatial representations can be linked to action selection mechanisms to produce trajectories to goal locations. Current artificial architectures have to balance the efficiency of computation with flexibility. A similar trade-off applies to the context of the watermaze.

In chapter 4, I discuss a previously published RL agent which uses hippocampal-inspired place representations to perform navigation in the watermaze [Foster et al., 2000]. The model describes the evolution of the link between places and actions, leading to direct trajectories towards a fixed goal location in a watermaze environment. However, the agent is not flexible towards changes in goal location. To address this, the model can be extended to learn coordinates over the space in order to efficiently compare locations for goal-directed navigation, thereby enabling flexibility [Foster et al., 2000]. For both the original approach and its extension, I discuss their biological realism in the light of recent experimental findings. Both architectures are in line with the involvement of striatal control and dopaminergic learning in spatial navigation. The coordinate-based extension is consistent with recent experimental results regarding goal and goal-directed encoding in the brain. To further facilitate the comparison of the model's behaviours to those of rodents, I implemented an additional behavioural metric commonly used in the watermaze literature.

Chapter 5 discusses a particular version of the model architecture examined in chapter 4 that considers continuous representations of actions and time. Continuous representations enable generalisation of knowledge and smoother control. This chapter discusses a mathematical framework consistent with continuous representations [Doya, 2000]. Frémaux et al. [2013] extended the continuous RL framework using spiking networks, and to spatial navigation tasks. Here, I adapt their approach in a new rate model, which considers temporal constraints of a biological neural code, for spatial navigation in the morris watermaze. I discuss the requirement for precision of the representation, by comparing spiking, discrete and continuous RL approaches in three

RL tasks, and in particular in the case of watermaze navigation. I show that precision of control and timescale of representation are linked, and that, in the example of the watermaze, high temporal precision does not seem necessary.

Given that flexibility relies on mechanisms that permit generalisation of experience, and that hierarchical reinforcement learning enables generalisation through abstraction, in chapter 6 I propose a new approach to flexibility in the watermaze using hierarchical representations. A hierarchical architecture, in which the selection of the goal is separated from the selection of the actions that lead to the goal, together with meta-computations, which adapt the behaviour of the agent from errors in goal selection, permit an agent to flexibly adjust to changes in goal locations. I show that the hierarchical model proposed is consistent with recent experimental results implicating prefrontal cortical areas in flexible spatial navigation.

Finally, flexibility in spatial navigation involves efficient planning procedures. Chapter 7 discusses how predictive properties of the tuning of hippocampal spatial representations can be used to plan trajectories. This chapter mixes results from graph theory, neural engineering and dynamical systems to derive a predictive representation of the 2D space for efficient memory storage and computations. Using this representation in a recurrent neural network enables the generation of trajectories to arbitrary goal locations [Corneil and Gerstner, 2015]. I investigate this approach further to examine the link between precision, timescale and computational cost within this framework. I show that the precision of the generated trajectory reduces with the distance of its spatial reach, and that precise trajectory plans are computationally costly.

To summarise, in this thesis, three computational concepts are discussed in their contribution to underlie flexibility in spatial navigation in the watermaze. First, behavioural flexibility seems to require generalisable and continuous representations for flexible transfer of knowledge, and to not restrict the resolution of the representations. Second, hierarchical control is useful to generalise action chains, such as selecting a trajectory, to any goal location. Third, predictive representations, which reflect the temporal statistics of experience, enable the generation of the most likely future scenarios to plan efficiently.

Chapter 2

Spatial navigation: the remarkable flexibility of animals and its neural substrates

2.1 Introduction

Successful spatial navigation is required in many every day tasks: humans and animals need to find food, shelter, and remember how to find these in diverse situations. Successful navigation requires knowledge of one's current location and where one wants to go, and requires choosing a favourable trajectory to get there. As our environments are constantly changing, animals have developed diverse strategies to succeed at such navigation. Spatial navigation requires reliable and flexible memory and decision mechanisms that enable animals to successfully adjust to changing situations.

In this chapter, I provide a general background on spatial navigation and its neural substrates. After introducing the basics of information transmission in the brain in section 2.2, I discuss the brain areas involved in memory and control of behaviours that are important for spatial navigation, and bridge the gap between behavioural studies and spatial navigation studies. In section 2.3, I introduce neurons specially tuned to represent spatial features. Behavioural procedures enable experimentalists to test how animals learn to associate a chain of actions to a reward, for example to reach a location in space, and section 2.4 provides a general background about the neural basis of action selection in the brain. Given the wide range of animal behaviours, and environments in which they take place, diverse control mechanisms have adapted so that brains save computational power when the situation is highly predictable and engage in more pre-processing when the situation demands prior careful considerations.

A commonly used apparatus to study spatial navigation is the Morris watermaze, which I describe in section 2.5. In particular, studies in this apparatus have revealed that the hippocampus, an area involved in memory and in which key neural correlates of spatial navigation have been found, is particularly crucial for flexibility in spatial navigation.

2.2 Information transmission in the brain: spikes and firing rates

In this section, I describe the basis of communication between neurons, following the expositions of Gerstner et al. [2014] and Galizia and Lledo [2013]. In this thesis, I focus on the level of networks of neurons. Figure 2.1 provide a schematic of a neuron and the important terminology that will be used in this thesis. Neurons are composed of dendrites, from which information is received, a cell body, called “soma”, that contains all the apparatus necessary for a normal cell to function (*e.g.*, a nucleus to store the DNA), and axons, through which the electrical activity propagates towards the following neurons. Neurons communicate with each other through chemical synapses. Synapses are small gaps between the cells through which information transmission occurs. A neuron that sends information to another neuron is called presynaptic and the neuron that receives the information is called postsynaptic. The information transmission between two neurons relies on neurotransmitters and ions: the difference in ion concentration inside and outside the cell generates an electrical voltage across the membrane of the cell which is known as “membrane potential”. Neurotransmitters are molecules that can bind to their allocated receptors in the synapses to modulate the opening of ion channels and trigger the release of ions in the synaptic cleft. This ion release modifies the membrane potential of both the pre and postsynaptic neurons.

When the membrane potential increases to a certain threshold, it can give rise to an electric discharge, which is characterised by a sudden sharp increase in potential, followed by a refractory period necessary for the membrane potential to return to baseline. This temporally localised sharp increase in the membrane potential is known as an action potential or spike. Figure 2.2a shows the temporal evolution of an action potential. A spike is temporally very localised (lasting only a few milliseconds). Figure 2.2b shows the temporal spiking patterns (also known as spike trains) of a neuron in response to the same stimulus. Spikes can be counted over a time window, and sometimes, when the experiment allows, averaged over similar subsequent situations to compute the firing rate which is indicative of a neuron's global activity. In figure 2.2c, the rate was computed from 50 responses to the same stimulus using a 10 ms window [van Drongelen, 2007]. In this thesis, I use mainly rate-based models, in which all units involved in the networks are described by a quantity referred to as "activity" which corresponds to a firing rate in biological tissues. Section 5.2.2 discusses the reason and the validity of the choice of the description of a neuron's activity by a rate.

Neurons can be excitatory, in which case their firing triggers an increase in potential in the postsynaptic neuron, which then becomes more likely to emit an action potential [Sayer et al., 1990]. The associated change in membrane potential is referred to as an excitatory postsynaptic potential (EPSP). Contrarily, certain neurons are inhibitory, in which case they reduce the probability of emitting a spike of the postsynaptic neuron. The postsynaptic neuron potential is then referred to as an inhibitory postsynaptic potential (IPSP).

The modification of the strength of a synapse is known as synaptic plasticity. It results in a stronger or weaker activation of the postsynaptic neuron from a presynaptic spike, depending on whether the synapse is strengthened or weakened. Synaptic plasticity is a phenomenon that seems to happen continuously, and that underlies learning [Citri and Malenka, 2008]. In the models considered in this thesis, the synaptic strength between

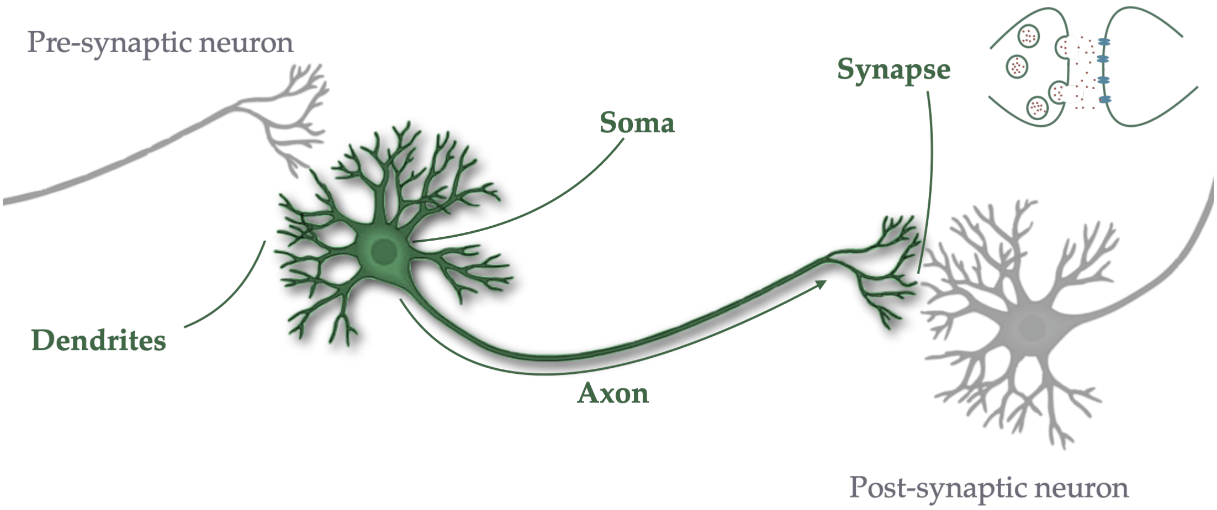


Figure 2.1: Schematic of a neuron (green cell) and a synapse (top right). Information from the presynaptic neuron (on the left) enters from the dendrites. This modifies the membrane potential of the neuron, which leads to the emission of a spike when it reaches a threshold. An electrical signal travels down the axon to release ions through the synapses, inducing a change of activity in the postsynaptic neuron. Top-right corner: the synapse is the locus of transmission of information of two neurons. Neurotransmitters bind to receptors to modulate the opening of ion (red dots) channels (blue "doors").

two cells is referred to as a "connection weight". Two units of the network will be linked by only one connection weight. In reality, the number of synapses between two neurons follows a bimodal distribution with a peak at 0 and a second peak at a small number (between 3 and 8) [Fauth et al., 2015].

The strength of the synapses between neurons defines pathways of information processing, which underlie cognition. In these pathways of information, neurons or groups of neurons can be specific to certain functions, and in particular for spatial navigation purposes. The following section focuses on the role of a particular brain area, the hippocampus, in spatial navigation.

2.3 The hippocampus and spatial navigation

The hippocampus is an area located in the medial temporal lobe (figure 2.3) traditionally associated with episodic memory, a type of memory where events occur within a certain spatio-temporal framework [Hasselmo, 2011]. Episodic memory is one of the two main kinds of long-term memory, along with semantic memory, which includes facts that are

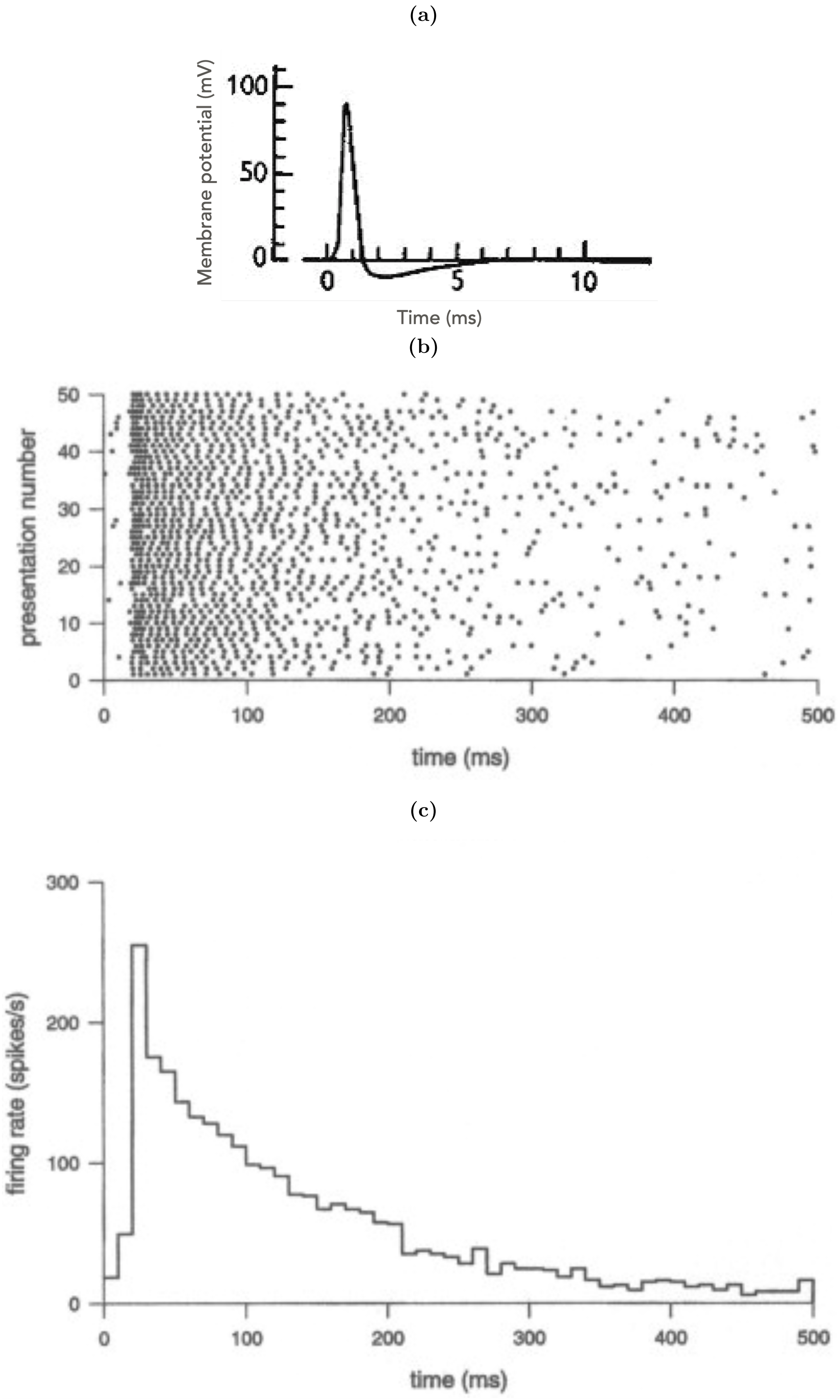


Figure 2.2: Spike, spike train and firing rate. (a) Action potential or spike, (b) spike trains of a neuron in responses to a visual stimulus: each row is a single response plotted against time, (c) and corresponding firing rate computed with a 10 ms window. (a) adapted from Losh and Llamocca [2019]. (b),(c) adapted from van Drongelen [2007].

common knowledge, for example names of colours or sounds of letters. The role of the hippocampus in episodic memory was identified with the study of patient HM, who, in an attempt to cure his epilepsy, had complete medial temporal lobe lesions, including very extensive hippocampal lesions. The observation that, after his surgery, he could not remember daily events as fast as they occur [Scoville and Milner, 1957], has opened investigations to unravel the crucial role of the hippocampus in memory formation and retrieval [Tulving and Markowitsch, 1998]. The hippocampus is thought to provide the link between time, space, events, contexts, and other details of scenes. As successful spatial navigation requires precise memory of past and current locations, of the current objective, and of landmarks allowing the choice of a good trajectory, it is consistent that the hippocampus is a key area for spatial navigation [O'Keefe and Dostrovsky, 1971, O'keefe and Nadel, 1978, Steele and Morris, 1999, Bast et al., 2009, Bast, 2011]. For example, one of the first symptoms observed in Alzheimer's disease patients, characterised by hippocampal loss [West et al., 1994], is spatial disorientation [Monacelli et al., 2003]. The following section exposes neural correlates of spatial navigation found in the Hippocampus. The important fact is that these neural spatial correlates are common to a wide variety of events, situations, and environments, and therefore hippocampal representations provide a general encoding of space, which enables an efficient representation of spatial memories. I describe the neural correlates most commonly referred to in the literature, highlight those that will be used in this thesis, and motivate the way they could be used for spatial navigation.

2.3.1 The cognitive map

Before the first single cell recordings in the hippocampus, the hippocampus was known to be involved in working memory from lesion studies [Douglas, 1967]. Previous studies revealed that rats with hippocampal damages showed poor performance in spatial maze tasks, in particular in tasks that require alternation of responses requiring an intact spatial working memory [Douglas, 1967]. In parallel, Tolman [1948] observed that animals that would naturally wander around in their environment prior to a task were much faster to

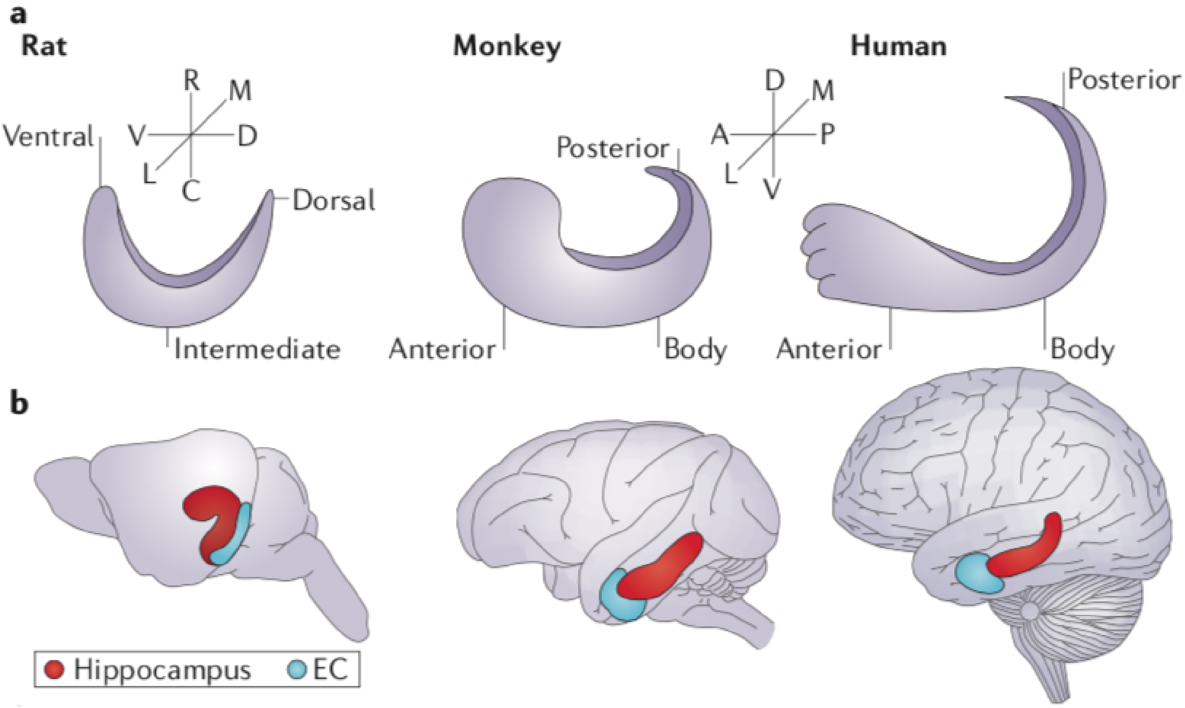


Figure 2.3: Location of the Hippocampus and Entorhinal cortex in the rat, monkey and human. The hippocampus can be divided along 3 axis: V: Ventral or D: Dorsal in rats (resp. A: Anterior and P: Posterior in primates); Rostral (r) or Caudal (C) in rats (resp. V: Ventral or D: Dorsal in primates); and Lateral (L) or Medial (M). Figure adapted from Strange et al. [2014] with permission.

learn the task than those that did not. He concluded that a form of latent knowledge must be formed during the first exploratory phase that can be used in the future if relevant. He referred to this transferable knowledge as “a cognitive map”, as it provides a general representation that can be used to acquire, store and recall information on a very wide range of situations [Tolman, 1948].

O’Keefe and Dostrovsky [1971] were the first to record isolated cells from the hippocampus while an animal was naturally wandering around an environment. They discovered place cells, neurons active mainly around a particular location of the environment, and whose firing rate reduces when the animal goes further from this location (see figure 2.4a). Based on these findings, the hippocampus has been identified as a cognitive map [O’Keefe and Dostrovsky, 1971, Stachenfeld et al., 2017, Behrens et al., 2018], and spatial representations within the hippocampal formation have been intensively studied; see the articles by Moser et al. [2017] and Jeffery [2018] for recent overviews. In this thesis, I use spatial coding by place cells to investigate how the properties of their firing profile can lead to

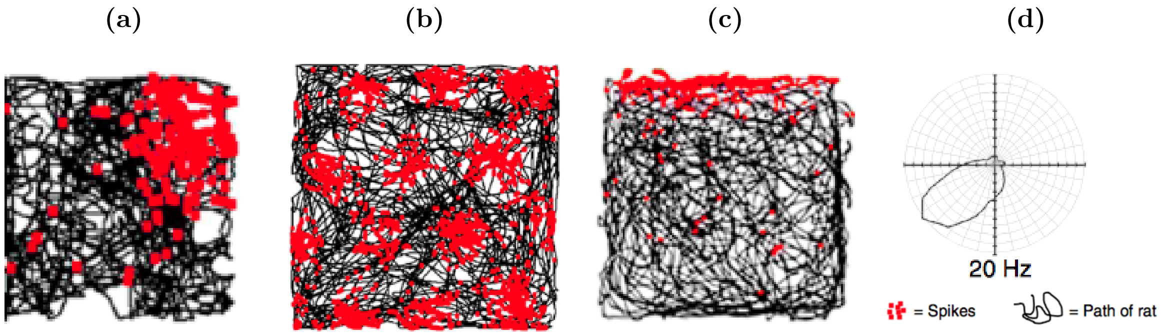


Figure 2.4: Examples of spatial coding in the hippocampal formation. (a) Place cells found in the hippocampus fire in an area centred around their respective preferred goal location (in this case, the top-right corner of the environment); (b) Head Direction cells in the subiculum (an area close to the hippocampus) fire more when the head of the animal points toward directions centred around their preferred direction (in this case, 225°); (c) Grid cells in the entorhinal cortex fire along a hexagonal grid; (d) Border cells in the parasubiculum encode proximity to a boundary. In (a), (c) and (d), the black line represents the trajectory of the rats and the red dots represent each time the selected cell emits a spike. Figure adapted from Marozzi and Jeffery [2012] with permission.

efficient learning in spatial navigation (in particular, in sections 4.3.1.2, 5.2.1 and chapter 7). Other spatial cells mentioned in the following paragraph are presented for general background, and to highlight that a cognitive map provides general and transferable representations for organised knowledge and behaviours.

Figure 2.4 presents typical firing profiles obtained by recording the activity of single cells of the most salient types of spatial coding in the hippocampal formation while an animal was wandering freely in an environment. Place cells [O'Keefe and Dostrovsky, 1971], already mentioned above, fire mainly around their preferred location (figure 2.4a). Grid cells [Moser et al., 2008] found in the entorhinal cortex, fire over a hexagonal grid (figure 2.4b). Head direction cells [Muller et al., 1996], which encode the head direction of the animal (figure 2.4d), are mainly found in the subiculum. Border cells [Marozzi and Jeffery, 2012], in the parasubiculum, encode the distance to a wall or boundary (figure 2.4c).

Various other types of spatial tuning contribute to the very rich literature of specialised spatial cells in the brain, for example, multi-modally tuned cells such as boundary vector cells [Barry et al., 2006], which encode both the distance and orientation to a boundary. Even within the place cells population, deeper investigations have led to the discovery of many other factors influencing their activity. For example, certain pyramidal cells respond to salient objects' locations [Rivard et al., 2004], to the location of peers [Danjo et al., 2018], and in rewarded tasks to reward locations [Hok et al., 2007, Gauthier and Tank, 2018] (see Poulter et al. [2018] for a review). Their activity can also be modulated by other sensory information, such as tactile [Gener et al., 2013], or gustatory [Herzog et al., 2019].

The aforementioned cells all encode information that is common to a wide variety of events and enable animals to be very flexible over a wide variety of environments and situations. Some spatially tuned cells also adapt to the current situation. For example, an effect known as remapping [Fyhn et al., 2007] is observed when place cells and/or grid cells change their firing profile or their preferred locations depending on the particular features of the current environment or task. With learning or repetitive exposure to a particular chain of events, place cell firing profiles can progressively change to represent more precisely one particular episode within a multi-episodic task [Sun et al., 2020]. The fact that spatial cell firing profiles can evolve dynamically, either through remapping, or to represent a new salient aspect of the current situation, such as a reward or a new route, is thought to be a key element of memory formation and flexibility [McKenzie et al., 2013, Stachenfeld et al., 2017]. Moreover, their activation can occur in other situations when the animal is not currently in the location or situation they preferentially code for, for example. This observation, which I discuss in more detail in the next section, suggests a more complex interpretation of their firing profile that goes from spatial correlates to memory engram.

2.3.2 Replay and preplay of spatial cells: between memory formation and improvement of performance

Rodent electrophysiology recordings show that while animals eat or rest there are brief high-frequency network oscillations in the hippocampus known as ripples [Buzsáki, 2015], accompanied by place cells activity that can "sweep" forwards and backwards of the animals' current location at compressed time-scales, *i.e.*, much faster than it would take the animal to move through the corresponding trajectory, a phenomenon that is linked with

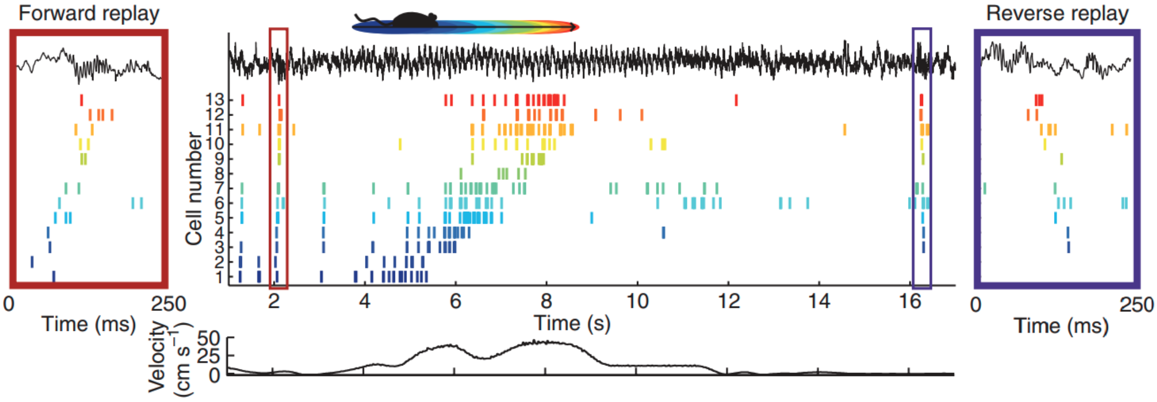


Figure 2.5: Reactivation of place cells before (left, red box), during (middle), and after (right, blue box) an animal's run along a straight line. Every line represents a place cell's spike train. The top layer represents the hippocampus CA1 local field potential, and the bottom layer represents the speed of the mouse. In this example, forward replay occurs mostly before the run, and reverse replay after the traversal of the environment. Adapted from Carr et al. [2011].

planning and memory formation [Foster, 2017].

Forward preplay is observed when animals pause at decision points, for example at the intersection of two branching corridors (behavioural observation interpreted as an indication of "deliberation" about future options, known as vicarious trial-and-error [Redish, 2016]), or before engaging in a task. At these moments, hippocampal place cell activity can be observed to sweep forward from the actual location of the subject [Pfeiffer and Foster, 2013], covering sequentially possible future scenarios [Johnson and Redish, 2007]. This mechanism plays a role in planning processes [Doll et al., 2015]: for example, preplayed patterns can be selective of goal location and predict the future behaviour of the animal [Foster, 2017], or correlate with the subject's future choice [Pfeiffer and Foster, 2013]. Backward replay events are sequential reactivations of place cells that occur in an order that corresponds to a reverse trajectory from the animals' current location to its starting location [Foster and Wilson, 2006]. These act like direct reactivations of the sequence just travelled and supposedly reinforce the neural pathways underlying the trajectory just selected [Foster and Wilson, 2006]. Figure 2.5 shows recordings of place cells while an animal runs along a track. Different place cell spike times are marked with different colours depending on their preferred location. During forward replay the order of spikes of these cells is preserved but occurs on a compressed timescale. After the run, when the animal stops, reverse replay can be observed as the order of activation of those cells occurs in reverse order on a compressed timescale.

Additional experimental evidence suggests that replay patterns are directly linked to behavioural performance in spatial navigation. First, replay is modulated by the discovery of salient rewarded locations [Igata et al., 2020] and by reward fluctuations [Ambrose et al., 2016]: backward replay rates are higher for higher rewards than for lower rewards. As replay activities are sometimes correlated with areas of reinforcement [Van Der Meer and Redish, 2011], they can facilitate or help consolidate place-reward associations [Lansink et al., 2009]. Similarly, hippocampal preplay serves to generate possible scenarios whose value would be computed by the striatum (an area involved in reward learning, discussed in section 2.4.2) [Chersi and Burgess, 2015, Stoianov et al., 2018], a phenomenon which is thought to be at the heart of goal-oriented search [Dolan and Dayan, 2013]. Moreover, the interruption of replay periods while learning a spatial memory task impairs performance [Jadhav et al., 2012]. Learning a task progressively modifies observed replay patterns [Shin et al., 2019]. Very early on during learning, forward and backward reactivations in the hippocampus seem linked to both the computation of value of possible scenarios [Ruediger et al., 2012], and to the consolidation of past place-reward associations [Lansink et al., 2009]. With learning, replay activities seem to shift from hippocampal reverse replay, retrospective evaluation, to forward replay, prospective planning [Shin et al., 2019].

To summarise, the various patterns of hippocampal replay are linked with a wide range of functions such as memory consolidation [Jadhav et al., 2012], forming and consolidating existing maps [Gupta et al., 2010], or planning [Johnson and Redish, 2007]. A recent theoretical work [Mattar and Daw, 2018] has proposed to unify the interpretations within an RL framework. Mattar and Daw [2018] suggest that the selection of replayed events could depend on a trade-off between the time and cost allocated to replaying a particular event and the advantages earned from replaying it. The argument is that an experience is worth being replayed during a task only if the replay influences positively the performance in this task, and if a similar situation is likely to occur in the future. Mattar and Daw [2018] designed a probability of replaying events that depends on both the benefit of the replay and the probability of encountering this situation again, and their simulations reproduce various experimental results about replay and preplay from the literature.

The replay of spatial cells is linked with learning and behavioural performance. In particular, preplay activities span possible future trajectories for the animals to plan ahead. In the watermaze, accessing replay is difficult as replay measurements necessitate an electrode left continuously in the brain of the animal, which is hardly compatible with navigation in a watermaze. Given the experimental results discussed in this section, which link replay and behavioural performances in other spatial navigation tasks, it is reasonable to assume that replay phenomena could be involved in watermaze tasks too. In chapter 7, I present a model to explain the generation of preplay trajectories in hippocampal place cell networks [Corneil and Gerstner, 2015]. Preplaying potential trajectories can help decision making by planning ahead. The next section provides more details about the neural substrates underlying decision making in the brain.

2.4 Behavioural control: from spatial memory to movement

The identification of a cognitive map for spatial navigation opened a door to much research on spatial memory. General spatial coding properties, such as those enabled by the cognitive map, are compelling as they represent a very efficient coding option for all-purpose spatial situations. Therefore, parts of our ability to successfully navigate to goal locations could be linked with the presence of these comprehensive representations. However, navigating requires selecting a trajectory, sometimes made of complex chains of decisions, in order to reach an intended destination. It is still largely unclear how spatial representations are linked with action selection mechanisms to successfully navigate. Spatial goals can be in direct reach, or out of any direct sensory perception. They can be located on an easy and safe path, or involve a complicated journey in which routes change quite often. Therefore, the hippocampal cognitive map should interact with the action control systems in the brain to successfully navigate. As most spatial destinations are rewarding locations, spatial navigation experiments, which substantially consist of training animals to reach a reward in a space, can be seen as a part of a physiological and behavioural process and area of study called conditioning. The next sections present general background knowledge on reward motivated systems, including how they could also play out in spatial navigation. This bridges the gap between general action selection mechanisms and the mechanisms that enable an animal to choose the next movement direction to reach a desired goal.

2.4.1 Spatial navigation and conditioning

Conditioning emerged in psychology with Pavlov [Pavlov, 1902] who repeatedly rang a bell before feeding dogs and progressively observed that the dogs started salivating from the observation of the bell, indicating that they could predict the reception of the food from hearing the bell. Classical conditioning experiments consist of associating a stimulus to a reinforcer, which can be a reward or a punishment, over repeated exposure to the contingency of presentation of the two. After conditioning, subjects' behaviour indicates that the sensory cue is triggering the expectation of receiving a reward and tends to trigger approach behaviour [Bierley et al., 1985]. Studies of the neural mechanisms underlying associations through reinforcement have exposed particular neurons in the ventral tegmental area (VTA) and substantia nigra, that produce the neurotransmitter dopamine, whose activity switches during learning of the association between a stimulus and the reception of a reward [Schultz et al., 1997].

Schultz et al. [1997] repeatedly presented a stimulus followed by a reward at a fixed time interval. They observed that early on during the phase of learning, when the reward has not been yet associated with the stimulus, neurons in the VTA fire more than baseline when receiving the reward. After repeated exposures to the same stimulus-reward contingency, they fire above baseline from the presentation of the stimulus itself, before the reception of the reward. If the reward is then missing at the time where it should occur, dopaminergic neurons activity is inhibited. Therefore, their activity correlates with a reward prediction error signal [Schultz et al., 1997]. Figure 2.6 shows recordings of a single neuron in VTA during repeated exposure to the same stimulus-reward contingency. Each line represents the spikes emitted by this neuron across repeated situations. The top panel shows the neurons' response to a simple presentation of a reward, and one can see that the neuron emits more spikes just after the reception of the reward. In the middle panel, after the stimulus has been associated with the reward, the neuron emits more spikes just after the presentation of the stimulus. In the lower panel, the stimulus has been associated with a reward, but in these trials, the reward is missing. As a result, the neuron emits more spikes at the presentation of the stimulus and is inhibited at the time where the reward should occur.

Classical conditioning consists of associating a stimulus to a reward in a passive way: the association is created by simple observation and reception of a reward. A similar procedure can be extended to generate an association between an action or a chain of actions and the reception of reward, which is known as operant conditioning [Skinner, 1971]. Dopaminergic neurons and dopamine are at the core of reinforcement mechanisms involved in operant conditioning [Schultz, 2016, Schultz et al., 1997, Glimcher, 2011]. In spatial navigation, learning to reach a goal location requires being able to reinforce the chain of decisions that were previously made until the reception of a reward, so that this can be successfully reproduced in a similar situation. Therefore, it seems important to investigate how reinforcement mechanisms might be involved in spatial navigation. Section 4.3.2 explains how the coding properties of dopaminergic neurons come into play for spatial learning in the Morris watermaze.

One of the main areas to which dopaminergic neurons project is the striatum, extensively studied for its involvement in action selection and behavioural control [Robbins and Everitt, 1992]. The striatum can be divided into functionally different areas, perhaps the prominent division present in the literature is between the ventral and dorsal striatum [Voorn et al., 2004], where experimental evidence suggests that the ventral striatum is involved in the behavioural expression of place memories, and the dorsal part more implicated in the control of low level kinematics of the displacement [Atallah et al., 2007]. The striatum receives inputs from a very wide range of areas, including cortex and hippocam-

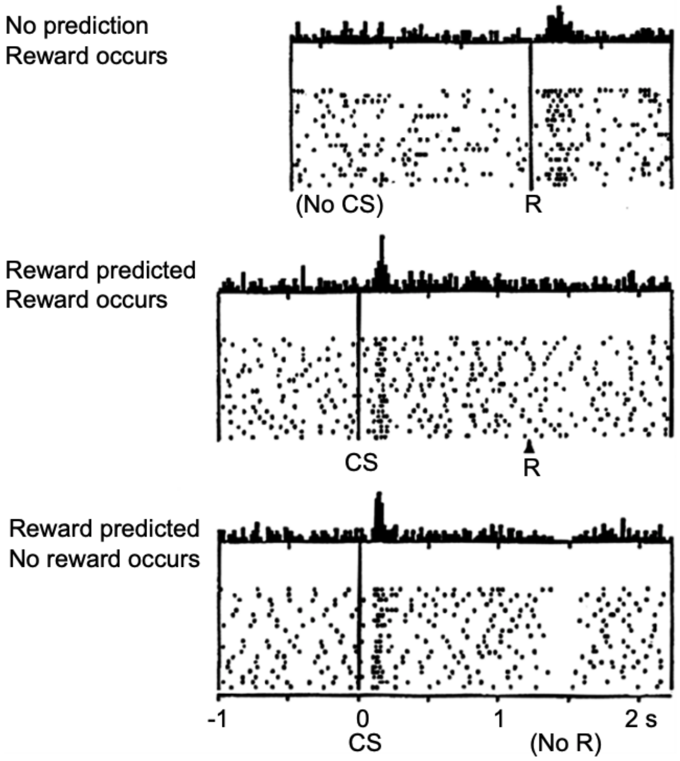


Figure 2.6: Ventral Tegmental Area dopaminergic neuron responses shift while learning. Each row shows the spiking times of a single neuron, or raster plot, on repeated situations of each kind. At the top of every raster plot, the firing rate obtained by counting the number of spikes over a very short temporal window is shown. Early on learning, dopaminergic neurons fire at the onset of the reward reception (up). After learning, they fire when the stimulus is presented (middle) and are inhibited if the reward does not follow the presentation of the stimulus (bottom). Their behaviour corresponds to a reward prediction error signal. Figure adapted from Schultz et al. [1997] with permission.

pus, which enable different parts of striatum to process information in parallel [Pennartz et al., 2011], and may underlie the capacity to adopt successful response strategies in diverse situations [Lau et al., 2017]. In section 4.3.2, I discuss in more detail its potential role for navigation in the Morris watermaze.

Another main locus of projection of dopaminergic neurons is the prefrontal cortex, an area also important for trajectory selection, spatial learning, planning and flexibility. In a nutshell, prefrontal areas are involved in the selection of chunks of trajectories, such as the selection of an arm leading to a reward at the end of it for example [Ito et al., 2015], and in planning mechanisms on longer timescales, such as planning long action agendas to pursue goals [Spiers, 2008]. It is an important area for spatial memory and goal representation [Hok et al., 2005, Poucet and Hok, 2017], in particular from its connectivity with the hippocampus [Binder et al., 2019]. Section 6.3.2 provides more detail about the type of computations that prefrontal areas could perform for spatial navigation, in particular about their role in flexibility in spatial navigation.

Dopamine influences areas that are crucial for decision making in spatial navigation. Dopamine release in the hippocampus is also crucial for forming and accessing goal memories [McNamara et al., 2014] and modulates hippocampo-frontal interactions in spatial navigation [Goto and Grace, 2008]. Reinforcement mechanisms in the brain are at the core of the development of reward-based behaviours. Being able to maintain a goal in memory, and to select the chain of actions necessary to pursue it is crucial for successful spatial navigation. This is a behavioural control problem. As decisions occur in various situations, sometimes stable and sometimes highly unpredictable, and to pursue various goals, sometimes more instantaneous than others, brains have developed many control solutions to this problem. In particular, the following section introduces two types of behaviour often contrasted in behavioural sciences, namely habitual and goal-directed.

2.4.2 Habitual and goal-directed behaviours

Operant conditioning involves studying how actions are associated with reward. In the lab, this can involve reinforcing one option and investigating how long it takes for this option to be selected by the animal. Complex behaviours can be taught to animals using this experimental technique [Skinner, 1971]. In behavioural sciences, two types of behaviours are often contrasted: habitual or goal-directed. Operant conditioning can be used to investigate both habitual and goal-directed behaviours and how situations often require to go from one to the other, and the dichotomy can also apply in the context of spatial navigation, as I explain below.

When a situation has been repeatedly successful in the past, the decision tends to become automatic. This is known as habitual behaviour. From their automatic nature, habitual behaviours are usually less flexible, and their development can lead to a insensitivity to changes in the action–outcome contingency, which is characterised by the repetition of the same behaviour even in situations in which it is not favourable anymore [Dezfouli and Balleine, 2012]. Addictions are an extreme example of habitual behaviours, as they can involve repeating the consumption of a rewarding substance, sometimes at the expense of sanity [Koob and Volkow, 2010].

By contrast, goal-directed behaviours involve careful consideration of the expected outcomes of decisions. When the task demand involves changing goal locations and/or reward rules quite often, or requires planning, the behaviour is thought of as goal-directed [Ito et al., 2015]. With repeated exposure to the same situation, for example when overtraining to the same goal location, the control of actions switches from goal-directed to habitual [Dezfouli and Balleine, 2012]. Both types of behaviour are useful in different situations (see a full review in Corbit [2018]). A habitual response enables animals to react faster and to save computational resource, but is successful only in stable and predictable conditions. A goal-directed approach requires more planning, but is more flexible in changing conditions, and therefore will be ultimately more successful in situations which are not fully predictable and in which time is not a limited resource [Corbit, 2018]. Most behaviours lie between the two extremes [Schreiner et al., 2020, Dayan, 2009].

A major area of research investigates the neural pathways underlying action selection and behavioural control, and the condition of development of habits and of maintenance of flexibility. Figure 2.7 presents an illustrative spatial navigation experimental design that

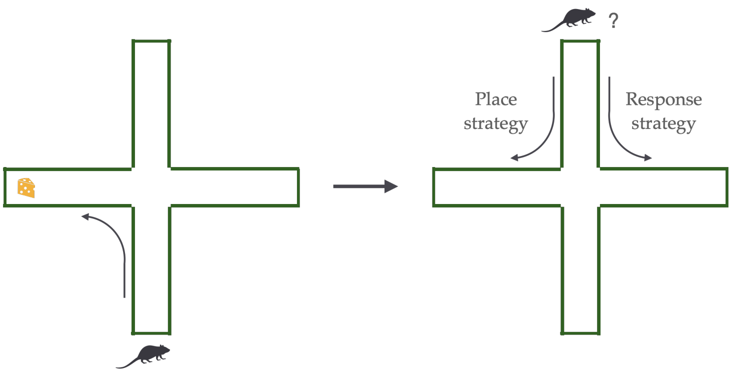


Figure 2.7: An example of experimental paradigm used to test the establishment of habitual behaviours. Rats start at a particular location (here the end of the south arm of the plus maze) and learn to find a food reward. When animals successfully reach the reward, it could be because they have learned that choosing a particular chain of motor commands, in that case turning left at the end of the corridor, is the correct response (response strategy), or it could be that they use a map of the space in which they can represent goal locations, in that case in the west arm of the maze, that they use to navigate (place strategy). Probe trials (left) enable the experimenter to distinguish between these alternatives: animals are placed at the end of another arm, and will choose to turn left at the centre if they are implementing a response strategy. However, if they are using a place strategy, they will go to the right arm using the available cues. Early on during learning, probe trials reveal that rats develop place strategy. However, with sufficient training, rats come to use a response strategy [Packard and McGaugh, 1996]. Figure reproduced from Corbit [2018].

enables one to disentangle habitual and goal-directed control. Rodents are first trained to find a reward at the end of a consistent arm in a multi-arm maze, always starting from the same position. If rats learn to reach the goal, a possible conclusion is that they have formed a cognitive map of the environment using the cues around them, and can use this map to represent the goal location and access it to move towards it using a goal-directed approach. A second possibility is that they associated the exact pattern of displacements that they went through to the reception of the reward, which would correspond to a habitual strategy. Placing the rats in a different starting location enables one to distinguish the two: rats which use a goal-directed approach will be able to find the reward, whereas rats with a habitual approach will repeat the same patterns of motions and go to the end of another arm. Early on during learning, rodents tend to use a goal-directed approach and show behavioural flexibility, whereas with sufficient repeated exposure to the same trial, they develop a habitual strategy [Packard and McGaugh, 1996].

Experimental evidence suggests that goal-directed strategies are supported by the ventral striatum [Liljeholm et al., 2015, Pennartz et al., 2011, van Der Meer and Redish, 2011], prefrontal areas [Spiers, 2008], the hippocampus [Pfeiffer and Foster, 2013] and their interaction [Ito et al., 2015, Pennartz et al., 2011]. Dopamine is crucial for the development of habitual behaviour from repetitive learning [Wang et al., 2011, Yin and Knowlton, 2006], as impairing NMDA receptors in dopaminergic neurons prevents habit formation, cue-reward association is still formed, but does not translate into behaviour [Wang et al., 2011]. Dopaminergic modulation of striatum is also required for habit formation [Faure et al., 2005]. Experimental evidence suggests that during the development of habitual behaviours, striatal control of the behaviour switches from ventral to dorsal [Vollstädt-Klein et al., 2010, Atallah et al., 2007].

To summarise, the control of behaviour adapts to the environmental demands, where more predictability leads to the development of faster but less flexible solutions than when the environment is more variable. The areas responsible for action selection and memory, and together with reinforcement mechanisms, interact in spatial navigation to enable the development of chains of actions to new goal locations and/or reinforce existing ones. In spatial navigation, regardless of whether the control is goal-directed or habitual, the neural spatial representations all require the use of sensory inputs that enable animals to form a representation of the current situation and to compare their evolution with predictions to make sure the direction followed is the one intended. These sensory inputs can be external, such as when using visual cues, or internal, for example using sensory-motor information to assess the speed and direction. In the next section, I distinguish navigation strategies commonly referred to in the spatial navigation literature.

2.4.3 Navigation strategies

Navigation strategies are often cue based, whereby animals use cues around or within an environment to determine their location and to choose a direction to reach a goal. The representation of landmarks and cues in relation to the locations of other objects, landmarks and cues in the environment is referred to as allocentric spatial coding (*e.g.*, “at the right of the red car”, “between the church and the post office”). Navigation strategies which include choices in direction that are made in reference to cues in the environment

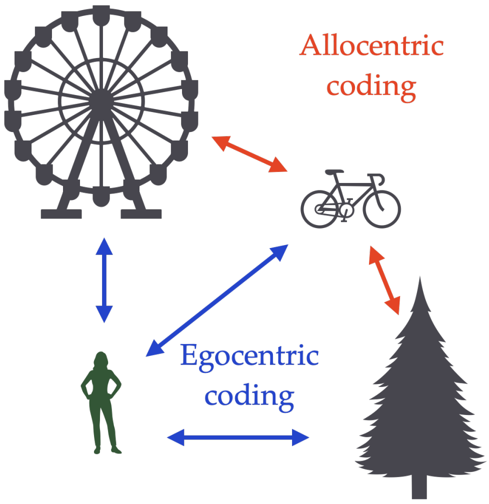


Figure 2.8: Allocentric (red) versus egocentric (blue) spatial coding. Allocentirc representation encodes the position of one object with respect to other objects (*e.g.*, between the tree and the wheel). Egocentric representation encodes the position of objects in space relative to the body axes of the self (*e.g.*, left, right).

are mentioned as allocentric navigation strategies (*e.g.*, “I move towards the church”).

Conversely, the term egocentric spatial processing is employed when cues, objects and landmarks are located relative to the body axes of the self (*e.g.*, left-right, front-back). Egocentric navigation appoints to direction choices made independent of environmental cues (*e.g.*, “after ten steps, I turn left”). Figure 2.8 illustrates the distinction between allocentric and egocentric navigation. Animals are known to employ egocentric strategies, a very common example of which is known as path integration, in which the current position is deduced from the sum of vectors of directions that have been travelled to from a starting point. This phenomenon is very well studied in desert ants, which can hold a very accurate home vector formed through navigating that enables them to reach their nest at any time during foraging [Müller and Wehner, 1988]. Interestingly, this vector is reset in ants only when they perceive cues that are associated with their nest: if transported somewhere else in the environment, even at the entrance of their nest, they will run in the direction of their previous home vector [Knaden and Wehner, 2006]. A recent computational model also provided a hypothesis on the neural network underlying this computation in the ant’s brains, and neurons encoding aspects of the computations suggested in the model have later been found [Haferlach et al., 2007].

Various experiments aim at elucidating strategies of spatial navigation and their neural substrates. They cover a wide spectrum from behavioural studies of methods employed by taxi drivers to perform route selection [Ziebart et al., 2008] and their neural peculiarities [Ekstrom et al., 2003], to *in vitro* electrophysiology recordings of cells hypothesised to play a role in spatial navigation [Isaac et al., 2009]. In this thesis, I focus on behavioural research, which uses humans or rodents as participants to study their behaviours and/or investigate their neural correlate.

The mazes used in spatial navigation studies can be of different shapes, from open fields such as a sand arena or wide pools of water, to complex 3D worlds containing many walls, corridors, and obstacles. Goals can be places where food pellets, escape opportunities, or numerical or financial rewards in the case of humans, are given to participants. One of the most frequently used laboratory tools in behavioural neurosciences is the Morris watermaze, which investigates allocentric cue-based navigation in rodents. The next section describes in more detail the apparatus and the main tasks that are performed in it that are relevant to this thesis. I discuss the results of lesion or drug studies in rodents on navigation tasks in the watermaze, aimed at exploring the neural substrates of this particular situation.

2.5 The Morris watermaze

2.5.1 Description of the apparatus

The Morris watermaze (figure 2.9) is used in rodent experiments to study the psychological processes and neural mechanisms of spatial learning and memory [Morris, 1981]. Rodents are placed in a circular pool of cloudy water, in which they have to swim to an escape platform. As they do not like swimming, reaching the platform constitutes a reward. The platform is submerged a few centimetres below the surface of water, such that it is hidden from direct view. Rodents have to rely on cues around the maze, such as visual cues, but also other sensory cues, in order to determine their location and remember where the platform is. A virtual version of this apparatus has been created for human participants, reviewed in Buckley and Bast [2018].

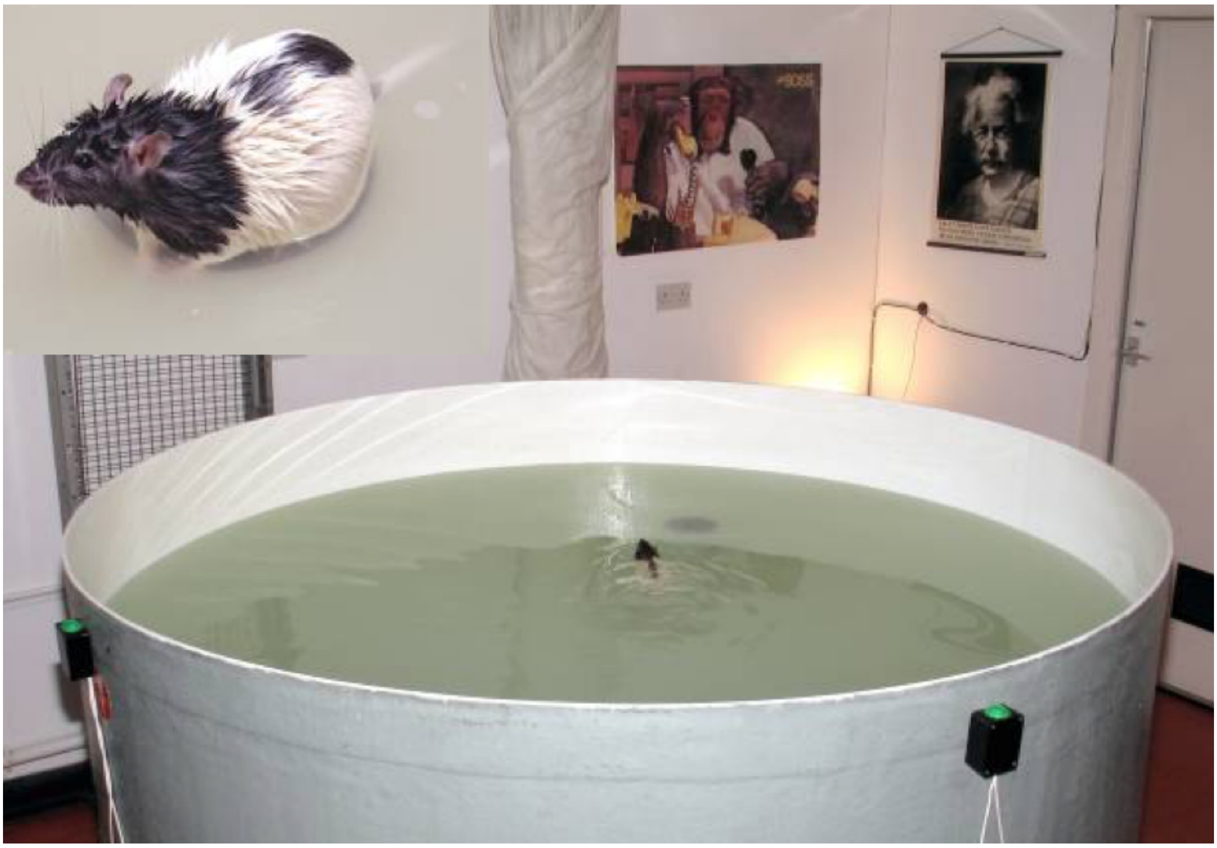


Figure 2.9: Picture of the Morris watermaze. Rats have to find a hidden platform in the pool of cloudy water. They rely on cues around the maze (for example, visual cues, as can be seen on the image) to determine their location and the correct direction to the platform. Picture showing the watermaze used by the team of Dr. Tobias Bast in Nottingham.

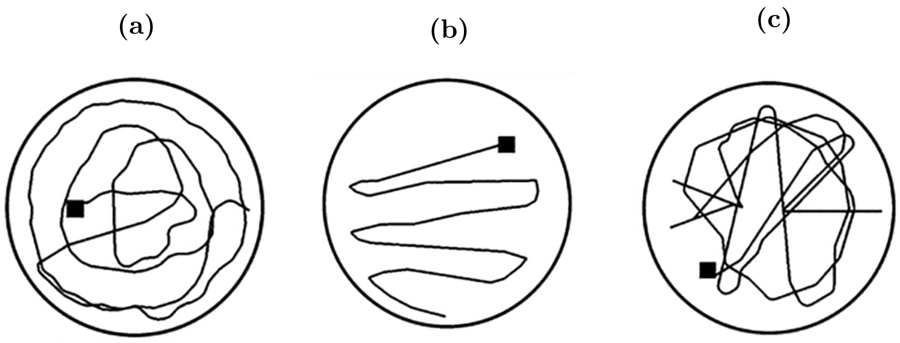


Figure 2.10: Typical search trajectories observed in humans during the first trial are quite systematic: humans adopt clear scanning strategies, such as circles (a), zig-zag (b), or a mix between the two (c). Figures adapted from Buckley and Bast [2018]. Rats tend to search the maze in circles, similar to (a) [Steele and Morris, 1999].

Many different tasks have been performed using this apparatus: studying learning of a fixed goal location [Morris, 1981], changing the platform location every few trials [Steele and Morris, 1999], comparing tasks in which the platform is cued or not [Pearce et al., 1998], or changing the relative position of the platform relative to cue [Oswald and Good, 2000]. For each task, a variety of behavioural measures can be used to assess learning: the latency, or time that rats take to reach the platform, is a common choice. In certain trials, the platform is removed for some time, which enables one to measure the persistence of rodents around the location of the platform last encountered. In such trials, "search preference" for the correct area/zone can be used to measure place memory. In standard watermaze learning tasks, the time in the correct quadrant compared to the time in other quadrants expressed as a percentage is often analysed. The trials in which search preference is assessed are referred to as probe trials, as the search preference can be a measure of the certainty that rodents have in their estimation of the platform location. Typical search trajectories during the first trial are quite systematic: rats tend to search the maze in circles, humans tend to search either in circles or using other systematic scanning patterns, such as zig-zags for example (figure 2.10 describes different stereotypical trajectories).

In the next section, I describe two common tasks performed in the watermaze. I show that they are supported by different neural substrates, and I outline the behaviours of healthy and hippocampally lesioned animals in those tasks.

2.5.2 Incremental learning in the Morris watermaze

In the original task performed in the watermaze, referred to as spatial reference memory test, the platform location remains the same over many trials and days of training. The animals incrementally learn the place of the hidden platform using distal cues surrounding the watermaze, and then navigate to it from different start positions [Morris, 1981]. The start positions are usually randomly chosen between the north, south, east and west extremities of the watermaze [Morris, 1981, Steele and Morris, 1999, Bast et al., 2009]. Learning is reflected by a reduction in the time taken to reach the platform location ("escape latencies") across trials and a search preference for the vicinity of the goal location when the platform is removed in probe trials (figure 2.11). Figure 2.11d shows that for healthy animals, approximately ten trials are necessary to attain almost optimal, stable, performances.

In behavioural studies, reversal learning refers to and measures the ability of an individual or animal to actively suppress the ongoing reward related responses and to adopt new responses when a change occurs in the task. In the watermaze, reversal learning can be assessed by changing the goal location and measuring the number of trials that rats require to adjust to the new location. In general, the change in goal location is made after many trials, to make sure the first location has been learned. In figure 2.11d, which presents the latencies for a fixed platform location for the first seven days and a change of platform location on the first trial of the eighth day [Foster et al., 2000], one can see that a single trial is sufficient for the rats to take direct paths to the new goal location on subsequent trials. This observation suggests that general spatial knowledge of this environment aggregated during the first trials to a particular goal location can be flexibly adapted to reach a platform in another location. This phenomenon is known as latent learning [Tolman, 1948] and will be discussed further in section 2.3.1.

On the incremental place learning task in the watermaze, hippocampal lesions are known to disrupt rats' performance [Morris et al., 1982], slowing down learning [Morris et al., 1990] and severely limiting rats' ability to navigate to the goal from variable start positions [Eichenbaum, 1990]. Figure 2.11a compares the latencies between hippocampal,

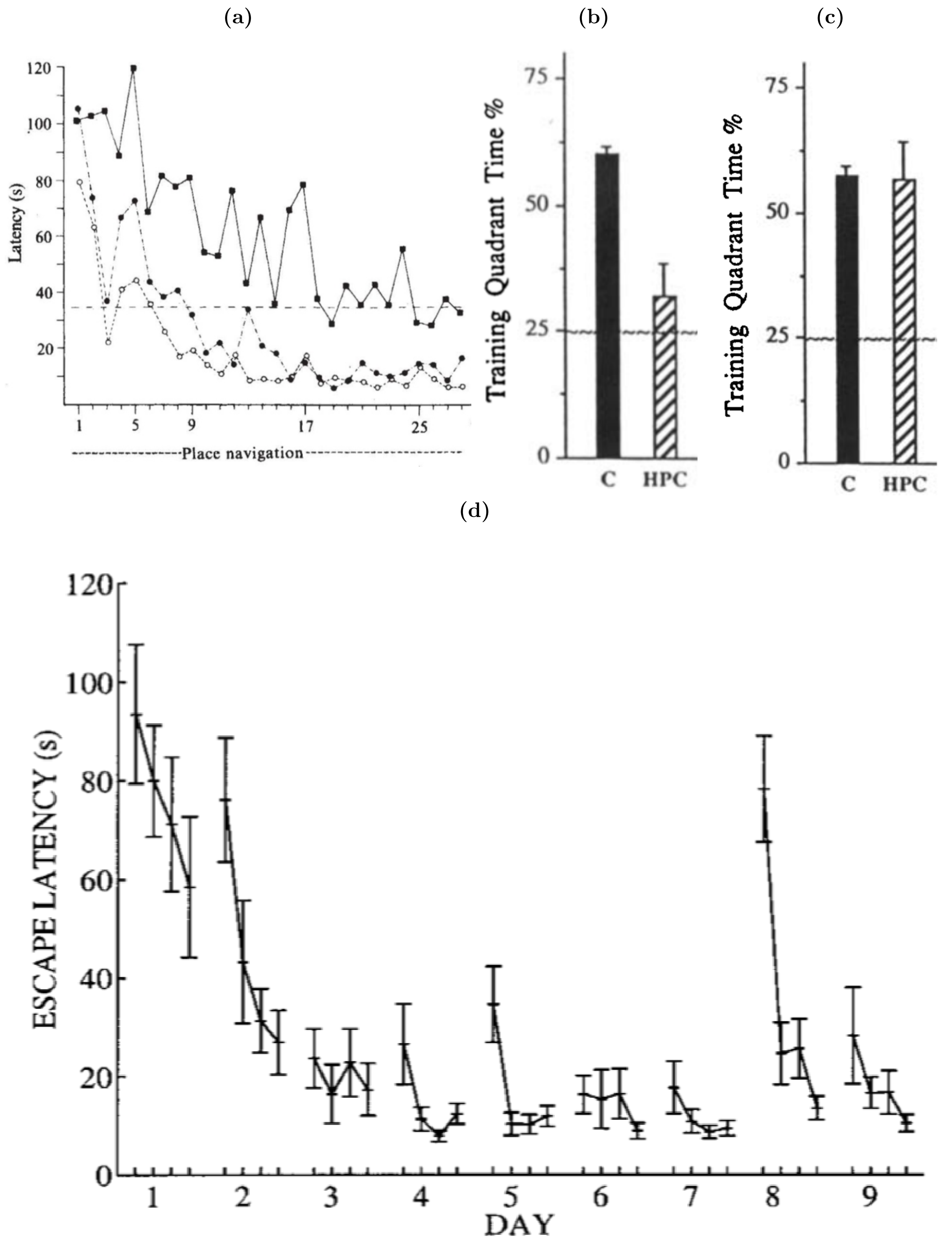


Figure 2.11: Performances of rats in the incremental learning task of the Morris watermaze. (a) Latencies for hippocampal lesioned animals (full square), cortical lesioned animals (full circles) and control animals (empty circles), adapted from Morris et al. [1982]; (b,c) search preference equivalent, computed on four quadrants instead of eight, for control animals (C) and hippocampal lesions (HPC), after few trials (b) and after extensive training (c), adapted from Morris et al. [1990]; and (d) latencies of control rats with reversal learning (the platform location changes on day 8), adapted from Foster et al. [2000] with permission.

cortical lesioned, and control animals, showing that hippocampal lesioned animals are slower to reach lower latencies than other lesioned animals. However, rats with partial hippocampal lesions sparing less than half of the hippocampus can show relatively intact performance on the incremental place learning task [de Hoz et al., 2003, Moser et al., 1995], and even rats with complete hippocampal lesions can show intact place memory following extended incremental training [Bast et al., 2009, Morris et al., 1990]. Figures 2.11b and 2.11c compare the search preferences, computed using four quadrants of equal surface around the maze, between control and hippocampal lesioned animals, after respectively limited exposure to the goal location (figure 2.11b) and after overtraining (figure 2.11c). One can see that early on in learning (*i.e.*, within the first few trials), complete hippocampal lesioned animals are almost at chance level, but that performance is intact after extensive training. Hippocampal N-Methyl-D-aspartate (NMDA) receptor blockade, which consists of using drugs to inhibit the action of the NMDA receptor involved in synaptic plasticity, does disrupt spatial learning on the standard watermaze task in a new environment [Bye and McDonald, 2019]. However, when rats have received pretraining, *i.e.*, when they already have been exposed to the platform position and have learned to navigate towards it, they still show intact performances in the task with blockade of hippocampal synaptic plasticity [Bannerman et al., 1995, Inglis et al., 2013]. These findings suggest that incremental place learning, although normally facilitated by hippocampal mechanisms, can partly be sustained by extra-hippocampal mechanisms. When the platform position changes regularly, however, the hippocampus becomes necessary for rapid adaptation, as the next section describes.

2.5.3 Rapid hippocampal place learning in the Morris watermaze

In reversal learning, the change in the task happens after many trials necessary to have learned an optimal response (figure 2.11d). Rapid spatial learning can be appraised in the watermaze by measuring the behavioural responses shown by rats when a change in

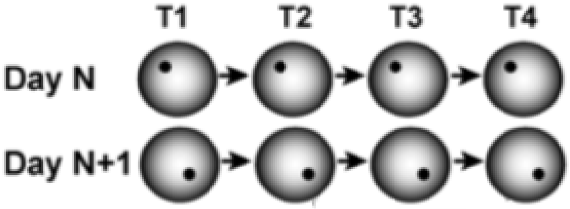


Figure 2.12: Delayed-Matching to Place (DMP) task in the Morris watermaze. Rats have to learn a new goal location (location of escape platform) every day, and complete four navigation trials to the new location on each day. Figure adapted from figure 2 in Bast et al. [2009].

platform location occurs quite often. In the delayed-matching-to-place (DMP) task, the location of the platform remains constant during trials within a day (typically four trials per day), but is changed every day (figure 2.12, Steele and Morris [1999], Bast et al. [2009]). As the changes in goal location occur quite often, the behaviour in this task is a measure of the flexibility shown by rodents towards changes in goal location.

A key observation from the behaviour of rats on the DMP task is that a single trial to a new goal location is sufficient for the animal to learn this location and subsequently to navigate to it efficiently [Steele and Morris, 1999]. This phenomenon is therefore commonly referred to as “one-shot” or “one-trial” place learning. Such one-trial place learning is reflected by a marked latency reduction between the first and second trials to a new goal location (figure 2.13a), with little further improvement on subsequent trials, and by a marked search preference for the vicinity of the correct location when trial 2 is run as probe (figure 2.13b) with the platform removed [Bast et al., 2009]. Buckley and Bast [2018] have shown that human participants display similar one-trial place learning to rats in the virtual reality DMP task developed for human participants.

In contrast to incremental place learning, rapid place learning, based on one or a few experiences, may absolutely require the hippocampus, with extra-hippocampal mechanisms unable to sustain such learning [Bast, 2007]. Studies in rodents have shown that spatial navigation based on one-trial place learning on the DMP watermaze task is highly sensitive to hippocampal dysfunction that may leave incremental place learning performance in the watermaze relatively intact. Specifically, one-trial place learning performance on the watermaze DMP test is severely impaired, and often virtually abolished, by complete and partial hippocampal lesions [Bast et al., 2009, De Hoz et al., 2005, Mor-

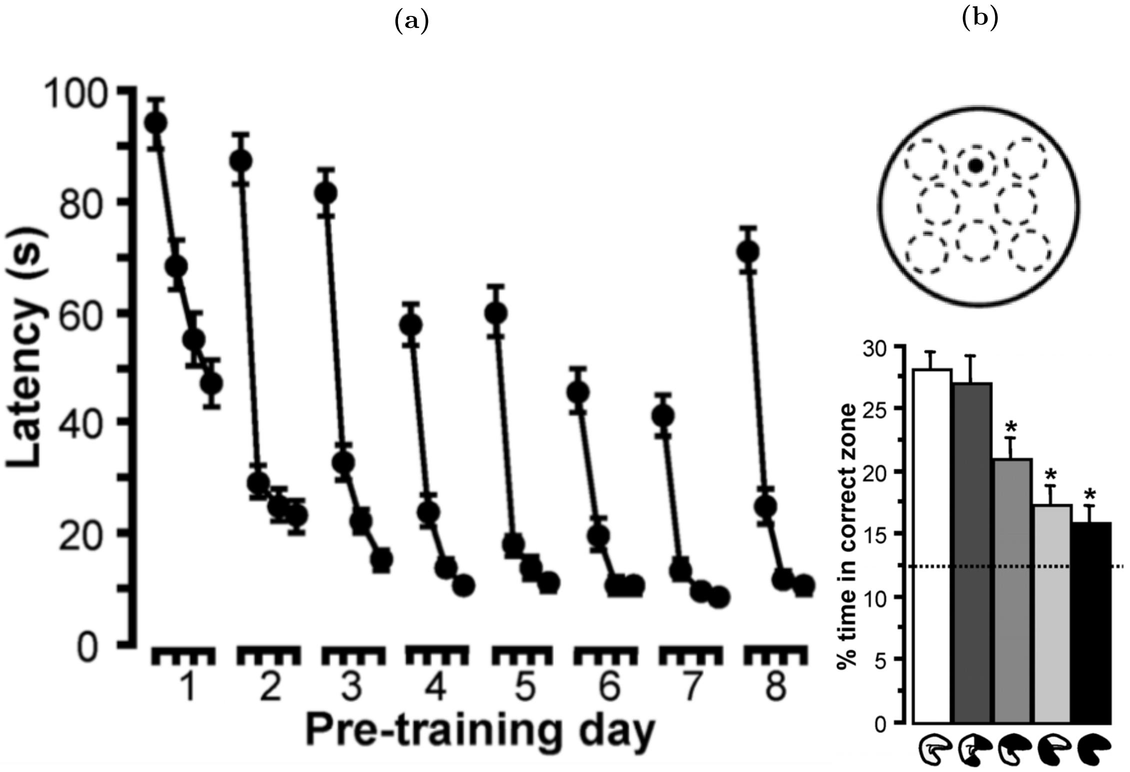


Figure 2.13: One-shot place learning by rats in the DMP watermaze task. (a) The time taken to find the new location reduces markedly from trial 1 to 2, with little further improvements on trials 3 and 4, and minimal interference between days; (b) Search preference for different hippocampal lesions. When trial 2 is run as a probe trial, during which the platform is unavailable, control rats show marked search preference for the vicinity of the goal location (left). To measure search preference, the watermaze surface is divided in eight equivalent symmetrically arranged zones (stippled lines in sketch), including the 'correct zone' centred on the goal location (black dot). The search preference corresponds to time spent searching in the 'correct zone', expressed as a percentage of time spent in all eight zones together. The chance level corresponds to 12.5%, corresponding to the rat spending the same time in each of the eight zones depicted in the sketch. These behavioural measures highlight successful one-shot place learning for control animals. Hippocampal lesions sparing less of the hippocampus significantly impair performances (third and fourth column), and full hippocampal lesioned animals (right) are the most impaired. Figure adapted from figure 2 in Bast et al. [2009].

ris et al., 1990], as well as by disruption of hippocampal plasticity mechanisms (Inglis et al. [2013], Nakazawa et al. [2003], O'Carroll et al. [2006], Pezze and Bast [2012], Steele and Morris [1999], also compare similar findings by Bast et al. [2005] in a dry-land food-reinforced DMP task) or by aberrant hippocampal firing patterns [McGarrity et al., 2017]. Rats with hippocampal lesions and NMDA receptor blockade show similar swim paths on trial 1 and trial 2 to the same goal location, swimming in circles over large areas of the watermaze surface [Steele and Morris, 1999, Redish and Touretzky, 1998], suggesting that they do not have or cannot access information about the recent goal location and/or to the history of their positions. Consistent with findings in rats that watermaze DMP performance is highly hippocampus-dependent, human participants' 1-trial place learning performance on the virtual DMP task is strongly associated with theta oscillations in the medial temporal lobe (including the hippocampus) [Bauer et al., 2020]. Overall, the findings reviewed above suggest that the DMP paradigm is a more sensitive assay of hippocampus-dependent navigation than incremental place learning paradigms, as the hippocampus may be necessary for good performance on the DMP task that cannot be sustained by other extra-hippocampal areas [Bast, 2007].

2.6 Summary

In this chapter, I provided a general background to information transmission in the brain and in particular in spatial navigation. I showed that spatial navigation involves precise memory mechanisms combined with action selection mechanisms.

I identified a crucial area for spatial navigation, the hippocampus, which contains neural correlates of spatial navigation, in particular place cells, which fire preferentially when the animal is at a certain location of the environment. These cells are involved in replay mechanisms that hint towards a link between performance and spatial memory.

As spatial navigating situations can involve successive decisions about the direction to follow to move to a rewarded location, spatial navigation studies can be integrated into operant conditioning studies which investigate the establishment of behaviours from the reception of a reward. Therefore, the areas implicated in action selection, in particular, the striatum and prefrontal areas, and reinforcement learning, with dopamine at its core, are also important for spatial navigation purposes.

Finally, the Morris watermaze allows one to assess allocentric spatial navigation, as animals have to rely on distal cues around the maze to form a representation of their location and choose a direction to reach the escape platform. In this apparatus, one can test place learning on different behavioural procedures, in particular rapid place learning, which can be assessed when the platform position often changes. One crucial result is that the hippocampus facilitates incremental learning to a fixed platform location, but is absolutely necessary for flexibility in the watermaze.

How can the hippocampal spatial cells be linked to the selection of a trajectory? In the next chapter, I present a computational framework that can be used to link spatial representations to actions. I explain how this framework can serve to perform spatial navigation.

How can one bridge the gap between spatial correlates featured by hippocampal place cells and the flexibility in the watermaze which fully demands the hippocampus? In this thesis, in particular in chapter 4, 6 and 7, I investigate different solutions about the computations underlying flexibility in the Morris watermaze using place cell properties as a basis of representation of positions.

Chapter 3

Reinforcement learning agents in spatial navigation

3.1 Introduction

Sutton and Barto [2018] defined Reinforcement Learning (RL) as a computational approach to learning, based on the idea that brains learn from interacting with the environment. In this chapter, I describe how robots and agents can learn by using RL algorithms. I provide the reader with the basic vocabulary used in the RL literature together with that which will be used throughout this thesis. Section 3.2 describes different approaches to learning within the RL framework, including model-free, model-based, and successor representation methods. I present the general objective of an RL agent and describe how these different approaches address it.

Section 3.3 describes how comparing agents and participants in different tasks enables one to formulate hypotheses about brain computations and information processing involved in certain tasks. In section 3.3.1, I highlight parallels between animals and synthetic agents that enable one to investigate which type of RL approaches produce behaviours that most closely resemble observed experimental data. In particular, I discuss this in the context of the Morris watermaze 2.9.

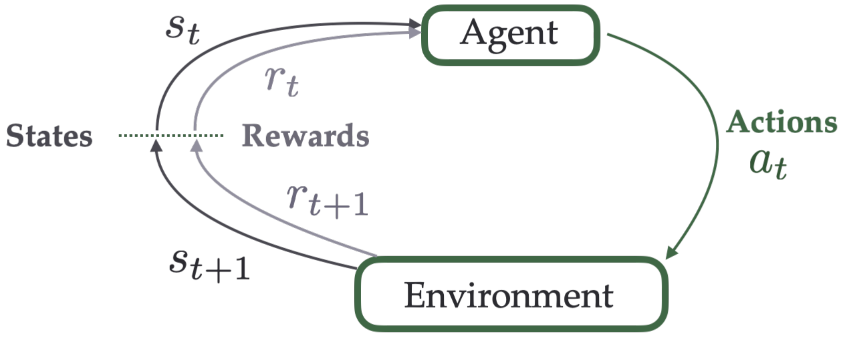


Figure 3.1: Key components of RL models. An agent in the state s_t (which, in a spatial context, often corresponds to a specific location in the environment) associated with the reward r_t takes the action a_t to move from one state to another. Depending on the available routes and rewards in the environment, this action leads to the reception of a potential reward r_{t+dt} in the subsequent state (or location) s_{t+dt}.

3.2 Reinforcement Learning

3.2.1 Definitions

In RL, the agent, modelling the animal or human, is not told what to do, but instead discovers which actions give most reward by trying them. This goal-seeking agent should exploit its knowledge and explore new actions. The agent-environment interface can be formalised within a mathematical framework. The introduction of the framework presented here is based on the exposition of Sutton and Barto [2018]. Typical RL problems involve the following components: states, actions, rewards, values, and policy [Sutton and Barto, 2018], as shown in figure 3.1 and explained below.

- A set of states $S = \{s_t, t \geqslant 0\}$ that contains all the information about the environment that is available to the agent at time t. In spatial navigation, states usually represent the agent location, but can be extended to describe more abstract concepts such as contexts or stimuli [Sutton and Barto, 2018].

- A set of actions $A = \{a_t, t \geqslant 0\}$. Actions are decisions to transition between states (*i.e.* a decision to move from one location to another in a spatial navigation context).

- A set of rewards $R = \{r_t, t \geqslant 0\}$. Rewards are scalar values usually given at certain spatial locations (*e.g.* $r_t = 1$ in the goal position, $r_t = 0$ elsewhere), mimicking goal locations in navigation tasks.

- A policy function $\pi_t(a|s) = Prob(a_t = a|s_t = s)$ represents the probability of taking the action a if the agent is in state s at time t. A policy is a probability distribution over the set of possible actions. In a spatial navigation context, this distribution defines which directions are more likely to be chosen when the agent is in a certain location.

- Values quantify the mean expected reward to be obtained under a given state or action. The value function $V^{P,\pi}$ can be a function of (*i.e.* dependent upon) the state alone, in which case it refers to the discounted total amount of reward that an agent can expect to receive in the future from a current state s at time t, further defined in equation (3.1). It depends on both the transition probabilities of the environment P, and on the current policy π (see equations (3.1) and (3.9) in section 3.2.2 and equation (3.19) in section 3.2.4). Alternatively, the value can refer to the state-action pair, in which case it refers to the value of taking a particular action at a certain state.

The problem that an agent faces is an optimisation problem. The goal is to find a policy π in order to maximise the value function $V^{P,\pi}(s)$ (*i.e.* to maximise the expected amount of reward). The policy that maximises the value function is known as optimal policy [Sutton and Barto, 2018].

With these definitions, the problem can then be represented as a Markov decision process, equipped with transition probabilities P between states that shape the way actions change states, and a reward function that maps states to reward. P refers to the transition probabilities between states naturally present in the environment. For example, the availability of a road to go from a place to another may depend on the weather or whether work is being undertaken. P reflect the statistics of these fluctuations. P is a matrix when the number of states is discrete. When the state space is continuous, P defines a probability transition density function. In a spatial navigation context, the transition probabilities typically depend on the spatial structure of the environment, and the latter can also vary temporally in certain contexts, as routes might open or close on particular occasions, for example. The probabilities of

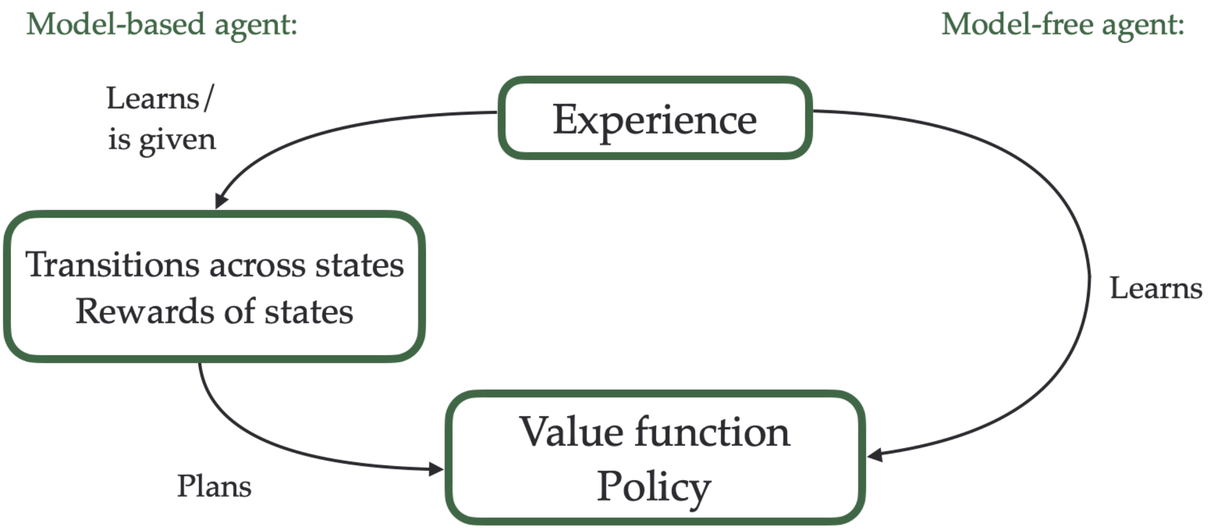


Figure 3.2: Model-free versus model-based approaches in RL. In model-free RL (right), an agent learns the values of the states on the fly, *i.e.* by trial and error, and adjusts its behaviour accordingly (to maximise its expected rewards). In model-based RL (left), the agent learns or is given the transition probabilities between states within an environment and the rewards associated with the states (a "model" of the environment), and uses this information to plan ahead and select the most successful trajectory.

transition between states/locations in an environment and the rewards available at every location form a *model* of the environment. In the models presented in this thesis, the transition matrix P is constant through time. This corresponds to a very stable environment, with no new route or addition of barriers during learning. The problem of finding the optimal policy depends on whether or not the agent knows the model beforehand or not. When the agent knows the model, it can then search through the next possible states using the transition matrix to select the best option. This in turn defines the best policy. These methods are known as *model-based* methods and I describe them in more detail in section 3.2.3. When the agent does not have access to the model, it has to discover the states it can visit and their associated rewards through direct exploration of its environment. This approach is called *model-free*, and is described in more detail in the next section. Figure 3.2 illustrates these two approaches.

3.2.2 Model-free RL using Temporal-Difference learning

In model-free RL (figure 3.2, right), the *model* is unknown and the agent must discover its environment, associated rewards, and learn how to optimise behaviour through trial and error. The agent needs to form an estimate of the value function and of the optimal policy. Both are expected to improve with experience, as the agent observes the consequences of its actions.

One possible way to achieve this is to repeat the same procedure many times and average the results, in a Monte-Carlo fashion [Sutton and Barto, 2018]. However, this demands storing every experience before being able to improve behaviour, and thus may require extensive storage capacity. Another way to improve behaviour from experience, which demands very little storage capacity, consists of updating estimates at every timestep, on the fly, from the observation of the current estimates. This process is known as bootstrapping and its most famous and widely used application is known as Temporal Difference (TD) learning. This thesis considers two versions of TD learning to perform spatial navigation in the watermaze, one in which time is discrete, and another in which time is continuous. The next sections present these approaches and the derivation of update rules for the value function and policy depending on the TD error.

3.2.2.1 TD learning in discrete time

When considering time as a discrete variable, as in Foster et al. [2000], for example, the value function is given by:

$$V^{P,\pi}(s) = E_{P,\pi}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \cdots | s = s(t)], \tag{3.1}$$

where $E_{P,\pi}[\cdot\,]$ refers to the expectation according to the transition probability matrix of the environment P and to the policy of the agent π. The value function quantifies the expected sum of all future rewards r_j that will be received at time j, $j \in \mathbb{N}$, discounted by a factor γ, which appraises the extent to which immediate rewards are favoured compared to delayed ones of the same magnitude. In this formulation, is it assumed that the reward discounting is uniform across time. The degree to which a reward is devalued between one time and the following is constant across time. In equation (3.1), this manifests as a geometrical discounting by powers of γ.

Here, P is assumed constant, therefore for simplicity in what follows, I will write $E_\pi[\cdot]$ and drop the dependence on P. As the chain of states s_j, $j \geqslant 0$, follows the Markov property, the value function satisfies the Bellman equation [Bellman, 1957]:

$$\begin{aligned} V^\pi(s) &= E_\pi[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \cdots | s = s(t)], \\ &= E_\pi[r_t + \gamma(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots) | s = s(t)], \\ &= \sum_a \pi(a|s) \sum_{s'} P(s', r|s, a) \left[r + \gamma E_\pi[r_{t+1} + \gamma r_{t+2} + \cdots | s_{t+1} = s']\right], \\ &= \sum_a \pi(a|s) \sum_{s'} P(s', r|s, a) \left[r + \gamma V^\pi(s')\right], \end{aligned} \tag{3.2}$$

where $P(s', r|s, a)$ refers to the probability of receiving reward r and reaching state s' knowing that the agent has chosen action a in state s. Note that the correspondence between the states s and the reward r is unique, as r is a function over the state space. The Bellman equation (3.2) expresses the relationship between the values of two successive states. Its development is possible thanks to the Markov property, the first line of the equation is conditioned on s_t and afterwards transmitted to conditioning on s_{t+1} [Bellman, 1957]. This equation is valid for every value associated to a certain policy, including for the optimal policy which maximises the value function:

$$\begin{aligned} V^*(s) &= E_{\pi^*}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \cdots | s = s(t)], \\ &= \sum_a \pi^*(a|s) \sum_{s'} P(s', r|s, a) \left[r + \gamma V^*(s')\right], \end{aligned} \tag{3.3}$$

where the optimal policy, π^*, is defined by:

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, V^\pi(s). \tag{3.4}$$

Equation (3.3) defines Bellman's principle of optimality [Bellman, 1957], which can be understood as taking an optimal decision now and following the optimal policy in the subsequent future being equivalent to following the optimal policy from the current state. If v_θ is an estimator for the optimal value function depending on a parameter or set of parameters θ, the Bellman equation (3.2) and the principle of optimality give a consistency equation that the estimator should minimise:

$$\delta_t = r_t + \gamma v_\theta(s_{t+1}) - v_\theta(s_t). \tag{3.5}$$

δ_t is known as Temporal-Difference error, and is used to update the parameter(s) θ to improve the estimator. Different ways to formulate an update for the parameters from this TD error are possible [Sutton and Barto, 2018, Doya, 2000]. The model presented in chapter 4 uses the following error:

$$E(t) = \frac{1}{2}|\delta_t|^2. \tag{3.6}$$

The formulation of the objective function enables it to be convex and a simple formulation of the gradient. This objective function can be used to derive a learning rule to update the parameter θ with the following rule:

$$\theta \leftarrow \theta - \alpha \frac{\partial E(t)}{\partial \theta} = \theta - \alpha \delta_t \frac{\partial \delta_t}{\partial \theta}, \tag{3.7}$$

which corresponds to a gradient descent of the objective function defined by the squared TD error. Most approaches use the first order approximation to the TD gradient instead of the full TD gradient $\partial \delta_t / \partial \theta$, an approach named semi-gradient descent, which can also be obtained via minimisation of the value error $v_\pi(s) - v_\theta(s)$ (see Sutton and Barto [2018], chapter 9, section 9.3 “Stochastic Semi-gradient Methods”):

$$\theta \leftarrow \theta + \alpha \delta_t \frac{\partial v_\theta(s_t)}{\partial \theta}. \tag{3.8}$$

Chapter 4 develops this approach for the case where the value is a linear function over the state representation. The following section presents a special case of the TD error algorithm presented here, in which the time variable t is continuous.

3.2.2.2 TD learning in continuous time

In the case where time is continuous, as in Frémaux et al. [2013], for example, the value function can be written:

$$V^{P,\pi}(s) = E_{P,\pi}\left[\int_t^\infty \mathrm{e}^{-\frac{t'-t}{\tau}} r(t')\mathrm{d}t' | s = s(t)\right]. \tag{3.9}$$

The role of temporal discounting in this case is held by the exponential term $\mathrm{e}^{-\frac{t'-t}{\tau}}$ in which τ is equivalent to an inverse discount factor. Here, the geometric discount proposed in equation (3.1) is replaced by an exponential kernel with a discount rate defined by τ. The intuition is similar to the discrete case discussed in (3.1), the discounting is assumed stationary with time. In (3.9), the reward function $r(t)$ is a general function of time, but in the models presented in this thesis, $r(t)$ is only a function of states, *i.e.* $r(t) = r_t(s_t)$. In other situations $r(t)$ can also depend on the action so that $r(t) = r_t(s(t), a(t))$. The optimal value function V^* corresponding to the optimal policy π^* can be written:

$$V^*(s) = \max_{a_{u\, u\in[t,\infty)}\in A} \left(E_{P,\pi}\left[\int_t^\infty \mathrm{e}^{-\frac{t'-t}{\tau}} r(t')\mathrm{d}t' | s = s(t)\right]\right), \tag{3.10}$$

where A refers to the action set introduced in section 3.2.1. Equation (3.9) can be differentiated with respect to time:

$$X(t) = \int_t^\infty \mathrm{e}^{-\frac{t'-t}{\tau}} r(t')\mathrm{d}t' = \mathrm{e}^{\frac{t}{\tau}} \int_t^\infty \mathrm{e}^{-\frac{t'}{\tau}} r(t')\mathrm{d}t'.$$

The Leibniz fundamental theorem of calculus for differentiating an integral gives:

$$\frac{\mathrm{d}}{\mathrm{d}t}X(t) = \mathrm{e}^{\frac{t}{\tau}}\mathrm{e}^{-\frac{t}{\tau}} r(t)(-1) + \frac{1}{\tau}\mathrm{e}^{\frac{t}{\tau}} \int_t^\infty \mathrm{e}^{-\frac{t'}{\tau}} r(t')\mathrm{d}t' = \frac{X(t)}{\tau} - r(t). \tag{3.11}$$

Taking the expectation under the current policy and transition probabilities π, P, and using the law of iterated expectations gives:

$$\frac{V^{P,\pi}(s)}{\tau} = E_{P,\pi}[r(t) + \frac{\mathrm{d}}{\mathrm{d}t}V^{P,\pi}(s)] = E_{P,\pi}\left[r(t) + \frac{\partial V^{P,\pi}(s)}{\partial s}\dot{s}(t)\right], \tag{3.12}$$

where $\dot{s}(t) = \mathrm{d}s(t)/\mathrm{d}t$. Applying (3.12) to the optimal value function leads to:

$$\frac{1}{\tau}V^*(s(t)) = \max_{a \in A}\left(r(t) + \frac{\partial V^*(s)}{\partial s}\dot{s}(t)\right). \tag{3.13}$$

Equation (3.13) is known as Bellman's principle of optimality [Bellman, 1957], and forms a condition for the optimal value function at time t which links the instantaneous value function to its value in the close future. Equation (3.12) allows one to form a consistency condition that value function should respect:

$$\dot{V}^{\pi}(s(t)) = \frac{1}{\tau}V^{\pi}(s(t)) - r(t). \tag{3.14}$$

This consistency equation enables the formulation of a TD error in continuous time:

$$\delta_t = r(t) - \frac{1}{\tau}V^{\pi}(s(t)) + \dot{V}^{\pi}(s_t). \tag{3.15}$$

If the value function here is also estimated depending on parameters, similarly update rules as in (3.7) can be used. Chapter 5 presents the application of this method to the case where the value function and the policy are linear estimators of the state representation.

3.2.2.3 TD methods: summary

The algorithms presented in sections 3.2.2.1 and 3.2.2.2 update an estimator of the value function from the error observed during a single experience. In actor-critic architectures, as will be described in chapter 4, both the value function and the policy are being learned simultaneously. Hence, the same TD error can be used to update both parameters. Chapters 4 and 5 consider examples of actor-critic architectures for performing spatial navigation.

TD learning is an example of a one step update used to learn from direct experience. For example, robots can be trained to walk towards a particular goal using a reward signal that is proportional to the distance travelled in the direction of the goal, a technique widely used in Brain-Computer Interfaces (for example, see Rosca and Leba [2017], Aljalal et al. [2020]). Agents trained using TD methods are in general very efficient in the task for which they are trained. However, such approaches usually are not flexible [Gershman, 2018]. Performing a new task requires first to discover that the former goal is not the current one. Then, it involves retraining the value and policy estimates for this new objective, which demands repeated exposures. This issue does not arise when agents have access to both the reward function and the transition probabilities of the environment, which is the case in model-based methods.

3.2.3 Model-based methods

In this section, I consider that the agent has access to the reward function r and to the transition probability P given by the environment. I assume that r and P are static and deterministic functions. Equation (3.3) of the optimal value function in discrete time leads to the following estimation:

$$V^*(s) = \max_a \sum_{s'} P(s', r|s, a) \left[r + \gamma V^*(s') \right]. \tag{3.16}$$

Equation (3.16) links the optimal value of a state to the optimal values at the subsequent states. The value of a state can be computed using a tree search. In a tree search, the value of a state s, $V(s)$, is updated at every step of the computation by $\max_a \sum_{s'} P(s', r|s, a) \left[r + \gamma V^*(s') \right]$, therefore the algorithm progressively searches through the possible branches of the tree starting at state s. Once the value of every state has been computed, one can thus select a deterministic policy:

$$\pi(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s', r|s, a) \left[r + \gamma V^\pi(s') \right], \tag{3.17}$$

which, in theory, is the optimal policy. This is referred to as Model-Based (MB) planning.

In reality, MB methods are often very computationally costly as the number of states can be large and they can be densely connected. In spatial navigation, for example, environments are sometimes finely discretised and all neighbouring states are connected if they are not separated by an environment boundary, leading to exponential trees, in which every node can lead to many new branches. When this is the case, an exhaustive visit of every possible branch of the tree is computationally infeasible. If so, the tree-search algorithm terminates when the difference in successive values reaches below a certain threshold, which should be chosen low enough for the resulting policy to be very close to the optimal policy [Sutton and Barto, 2018].

Also, and more importantly, in natural situations, it is very unlikely that any agent will have exhaustive information about both the reward and transition probabilities. However, having such information leads to better predictions, and so ideally one wants to be able to form a model from interacting with the environment and use it to plan ahead. An algorithm known as Dyna was originally introduced by Richard Sutton [Sutton and Barto, 2018] and uses update mechanisms introduced in section 3.2.2 to learn the model of the environment and use it to form improved predictions.

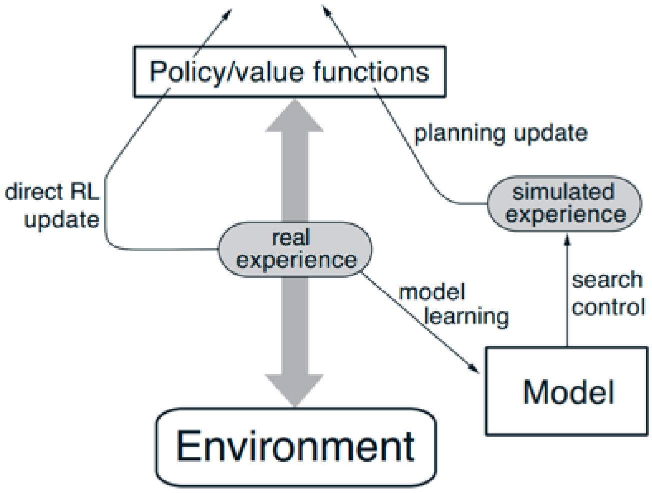


Figure 3.3: Dyna general architecture, adapted from Sutton [1991]. In Dyna, real experiences are used to improve the behaviour online (left part of the diagram), *i.e* the policy and value function estimates. Each pair of (decision, outcome) experiences is stored to create a model of the event (right part). The model is then used offline to replay past events in order to update the policy and value function.

In Dyna, the agent stores the transition experienced at every time step, along with the outcome of this transition (figure 3.3). For example, the set $(s_t, a_t, s_{t+1}, r_{t+1})$ corresponds to receiving r_{t+1} in the state s_{t+1} reached after having chosen the action a_t at time t from state s_t. From these experiences, a model of the environment can be formed, which links the pair (s_{t+1}, r_{t+1}) to the decision pair (s_t, a_t). The Dyna algorithm includes replay periods, in which particular events are sampled from the model in order to produce simulated experiences that will improve the policy and the estimate of the value function offline [Sutton, 1991]. In practice, this means selecting a stored set $(s_t, a_t, s_{t+1}, r_{t+1})$ according to a chosen probability distribution, and using equation (3.7) to update the estimates.

As they can be replayed, fewer real encounters with the environment are required to achieve good performance, allowing faster learning [Sutton and Barto, 2018]. Events to be replayed can be selected uniformly at random or following a particular distribution. The choice of a probability distribution over replayed events can shape the behaviour of the agent. Additionally, since the number of events to replay can be very high, replaying them all is computationally infeasible, so one wants to select the events carefully to maximise the benefit of the chosen replay. For example, in Cazin et al. [2019], replay choice probability is reward-modulated, such that events that are closer to reward reception are replayed more frequently. This leads to the discovery of shortest paths more quickly, in fewer real trials, than a normal agent.

Another extension includes the replay of *important* events, called prioritised replay, in which the events that have led to extremely positive values of reward prediction errors are more replayed [Sutton and Barto, 2018]. In that case, the events are chosen only according to how influential the replay will be. Building on this idea, Mattar and Daw [2018] proposed that the choice of replayed events should be done according to a trade-off between the cost of replaying the event and the benefit of replaying this particular event. The probability distribution over events that they use is computed according to the likelihood of the event occurring in the future and how much replaying the event would improve the behaviour. It is a slight improvement to the prioritised replay algorithm in that it also takes into account the relevance of replaying the event as well as its effect. In their paper, Mattar and Daw [2018] are able to match the behaviour observed from diverse recordings of replay from hippocampal place cells of rodents to their models' behaviour.

Replaying events can be useful to accelerate learning. It enables the agent to use fewer real encounters with the environment to reach good performance. This offline learning can be used to update the value function and policy as presented here, but can also be used to learn statistics over the transition of the environment from the observations of pairs of states (s_t, s_{t+1}), for example. The next section presents an alternative approach, the successor representation, which represents a trade-off between model-free and model-based approaches and consists of using the long-term statistics of the environment.

3.2.4 Successor Representation

In this section I consider a situation in which the agent has access to the reward function r and to the transition probability matrix P_π of the Markov chain given by the environment and the policy of the agent. The state space is of finite size N.

The value function, given in equation (3.1) can be written:

$$V_P(s) = \sum_{t=0}^{\infty} \gamma^t P_\pi(s|s_t) r(s_t), \tag{3.18}$$

where $r(s_t)$ refers to the reward that the agent will receive when reaching state s_t, and $P_\pi(s|s_t) = \sum_a \pi(a|s_t) P(s|s_t, a)$ is the probability of reaching state s knowing s_t. If the reward function over the state space is constant in time and only a function of the state, *i.e.* for all t and s_t, $r(t) = r(s_t)$, it can be written:

$$r = [r(s_1), r(s_2), \ldots, r(s_N)] \in \mathbb{R}^N.$$

Similarly to the value function $V_P \in \mathbb{R}^N$, the vector of values over the state space can be written:

$$V_P = [V_P(s_1), V_P(s_2), \ldots, V_P(s_N)],$$

so the value function expression in equation (3.18) can be developed as:

$$V_P = \sum_{t=0}^{\infty} \gamma^t P_\pi^t r = (\mathbb{1}_N - \gamma P_\pi)^{-1} r. \tag{3.19}$$

The matrix

$$M = (\mathbb{1}_N - \gamma P_\pi)^{-1} \tag{3.20}$$

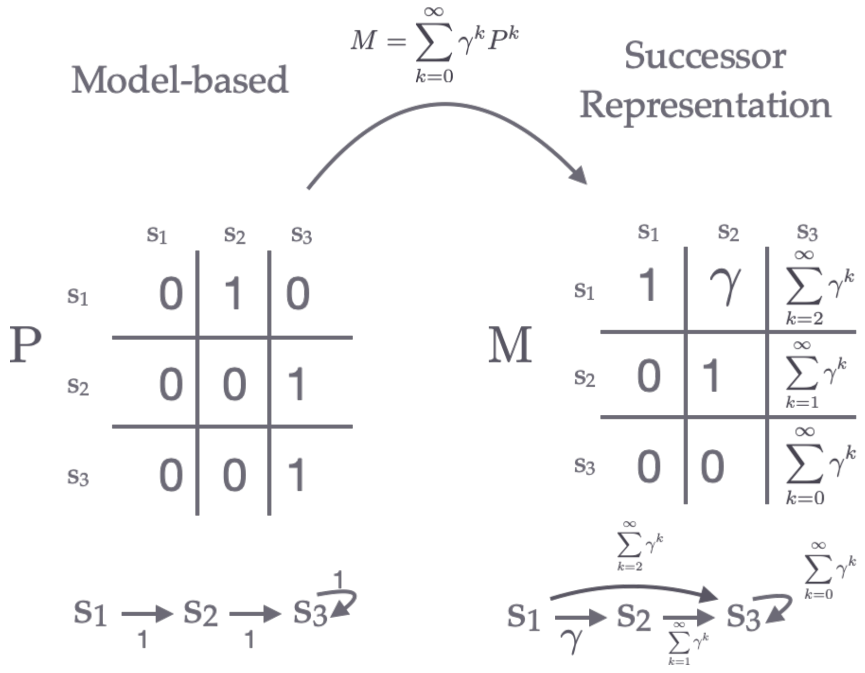

$M = \sum_{k=0}^{\infty} \gamma^k P^k$
Model-based
Successor Representation
P
M
s1 s2 s3
$\begin{array}{c|c|c|c} & s_1 & s_2 & s_3 \\ \hline s_1 & 0 & 1 & 0 \\ s_2 & 0 & 0 & 1 \\ s_3 & 0 & 0 & 1 \end{array}$
$\begin{array}{c|c|c|c} & s_1 & s_2 & s_3 \\ \hline s_1 & 1 & \gamma & \sum_{k=2}^{\infty}\gamma^k \\ s_2 & 0 & 1 & \sum_{k=1}^{\infty}\gamma^k \\ s_3 & 0 & 0 & \sum_{k=0}^{\infty}\gamma^k \end{array}$
S1 → S2 → S3
1
1
1
γ
$\sum_{k=2}^{\infty}\gamma^k$
$\sum_{k=1}^{\infty}\gamma^k$
$\sum_{k=0}^{\infty}\gamma^k$

Figure 3.4: Transition Matrix and Successor Representation matrix of a simple Markov chain of 3 states. While the MB representation (left) involves one-step transition information, the SR contains longer term, multi-step transition information. In this very simple illustration, the model based agent in s_1 does not know that it will visit s_3 in the future. The SR agent knows, and can therefore plan its optimal policy faster. The value of the connection $M(s_1, s_3)$ is higher than the value of $M(s_1, s_2)$ for $\gamma > 1/2$, in which case the agent is more likely to directly jump from s_1 to s_3 than to visit s_2 first when performing a tree-search. One can see, in such an example, how a high discount factor speeds up tree-search and leads to future projections that are further in time.

is called the successor representation (SR), introduced in 1993 by Peter Dayan [Dayan, 1993]. The SR is the matrix M that links each state to all the other states within an environment according to how many (discounted) times one can expect to visit the other states in the future from this current state. It therefore contains some predictive information. In the SR, the computation of the prediction of the environment and the computation of the rewards are separated (equation (3.20)). One row of the SR matrix M, $(M(s,i))_{i\in 1\cdots N}$, represents how many discounted times can the agent expect to visit all the states starting from state s. One column, $M((i,s))_{i\in 1\cdots N}$, represents how many discounted times the agent can expect to visit state s from all other states.

The SR can be learned using TD methods. As the value function estimates the future reward reception, the SR measures the future state occupancy. Therefore, a temporal difference between the predicted state occupancy and the observed one can be formulated, which can be used to update any estimate $\hat{M}$ of the SR [Russek et al., 2017]. The occupancy of a state can be learned using the following TD error:

$$\delta_t(s) = \mathcal{I}(s_t, s) + \gamma \hat{M}(s_{t+1}, s) - \hat{M}(s_t, s), \tag{3.21}$$

where $\mathcal{I}(s_t, s) = 1$ when $s_t = s$ and otherwise 0.

Additionally, the SR representation is efficient for planning. The SR can be used to compute the value function using the same planning steps as described in (3.16) but using the SR matrix. A tree search using a model based representation is on average longer and more computationally costly than a tree search using the successor representation, as the SR selects the scenarios that are more likely to be expected in the long run. Figure 3.4 illustrates the different representations of MB and SR approaches in a very simple environment made of three states s_1, s_2, s_3, and a simple transition matrix which only allows the agent to move forward, *i.e.* from s_1 to s_2, from s_2 to s_3, and s_3 is a terminal state (the agent can only stay in s_3 from s_3). On the left side of figure 3.4, the MB agent has information about the single-step transition probabilities. On the right side of figure 3.4, the SR agent has information about the long term state occupancy. If the agent is in state s_1 and wants to plan ahead, the SR agent is usually faster at doing so, because in one single computation it can have access to information of what will happen at s_3, whereas the MB agent has to go through s_2 to have access to s_3, and therefore requires one more computation to have access to information at s_3 than the SR agent. Moreover, the SR matrix coefficients determine the probability of selecting branches for tree search. In the example, if $\gamma > 1/2$, the value of the connection connection between s_1 and s_3, $M(s_1, s_3)$, is higher than the value of the connection between s_1 and s_2, $M(s_1, s_2)$ (i.e. for $\gamma > 1/2$, $\sum_{k=2}^{\infty} \gamma^k > \gamma$), therefore the SR agent is most likely to search directly s_3 from s_1 than to pass by s_2 first when performing tree-search. In this simple example, one can see that the larger the discount factor is, the more likely the SR agent will search far ahead, and the faster it will explore the more likely scenarios, which is consistent with the role of discount factor as defined in equation (3.1). In equation (3.5), the discount factor γ referred to as a temporal discounting of rewards. In (3.20), γ is transferred to discount over state occupancy, because the SR representation is obtained from the factorisation in

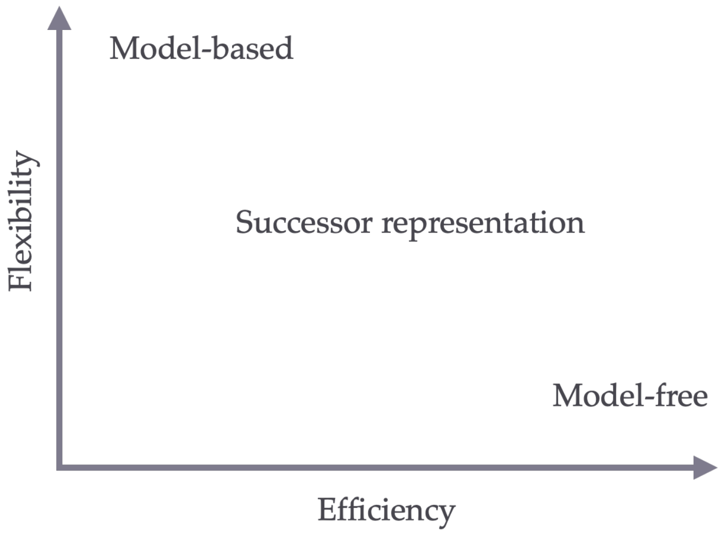


Figure 3.5: Different RL systems organised within an efficiency/flexibility trade-off. The efficiency axis represents the computational requirement of the type of model mentioned: from costly (left, *e.g.*, tree-search) to cheap (right, *e.g.*, only linear update) computation. The flexibility axis represents the ability of the model mentioned to adapt flexibly to changes in the environment, from needing a lot of experience after the change to gather the necessary data for the estimates to converge to the current new value (bottom), to architectures that need very little new experience with the environment to compute the new value estimates (top). Adapted from Gershman [2018].

equation (3.19), which originates from the formulation of the value function.

This simple agent illustrates that the SR agent will be faster at planning than the MB agent. On a graph of states with many branches of different depth, SR search is on average much faster. The drawback is that when a change in reward distribution occurs, the SR agent will take longer to find the new optimal policy than the MB agent. Figure 3.5 illustrates the different RL approaches within an efficiency/flexibility trade-off [Gershman, 2018].

In this section, three main RL approaches have been described. All these approaches propose different solutions to the flexibility/computational cost trade-off. As such, they can be used to investigate which form of solution the brain might have found to also comply with this trade-off. The next section presents some navigating agents and their performance and discusses some evidence of implementation of model-free or model-based strategies in the brain.

3.3 Comparison of RL agents with humans and rodents

In this section, I discuss results in the literature comparing the behaviour of rodents and humans to that of different RL agents in RL tasks. The comparison of behaviour enables to shed light on the underlying model that rodents or humans use in order to take a decision. I will explain that both humans and rodents use both model-based and model-free approaches. Then, I will discuss the example of navigating in the watermaze, and argue that the representation that rats seem to use lies in between model-free and model-based.

3.3.1 To which degree are brains model-free or model-based?

Comparing the behaviour of humans and animals to those of simulated agents of each type in different reward-based tasks permits one to gain understanding of the type of cognitive strategies underpinning human and animal decision-making. I have presented two computational RL perspectives, model-free and model-based, both aiming at generating a policy that maximises the expected number of rewards, the former updates the value according to outcomes of experiences, and the latter plans ahead which actions are best. A very prominent task designed for this purpose that has been used first on humans and then adapted to rodents is the two-step decision task [Da Silva and Hare, 2019, Miller et al., 2017, Daw et al., 2011]. In this task, participants first choose between 2 states, which afterwards lead to final states with different probabilities, usually unequal, making one transition “rare” and the other “common”. The final states have unbalanced reward probability distributions (see 3.6 for a diagram of the task for both humans and rodents). Investigating how subjects adjust to rare transition outcomes indicates whether they have access to the model or not. A model-free agent will adjust its behaviour only based on the outcome, whereas a model-based agent will adjust also according to the probability of this transition. Additionally, by changing either the structure of the transition, or the reward distributions, one can gain different insights

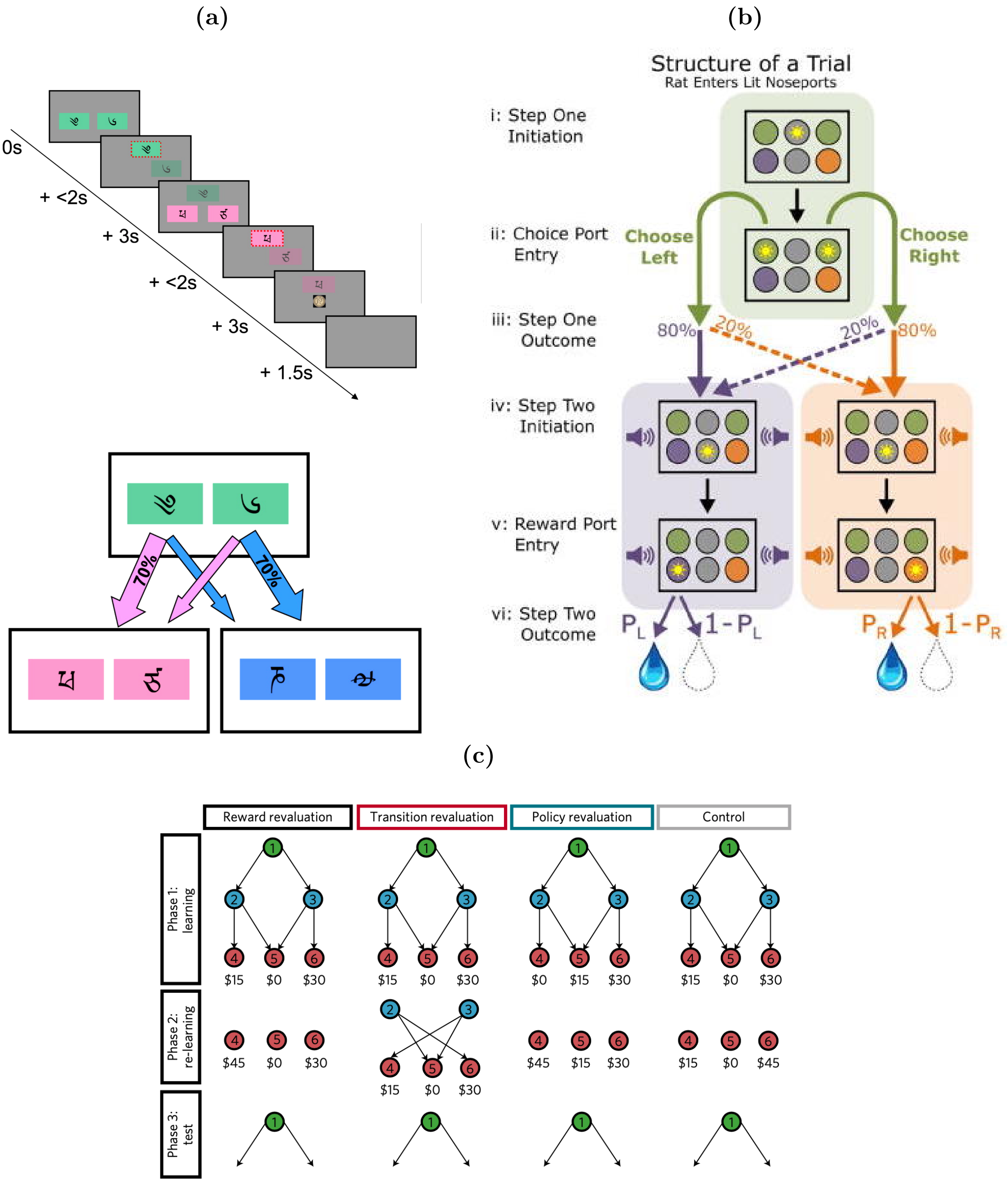


Figure 3.6: Schematic of the 2 steps decision task. (a) Original task developed by Daw et al. [2011] to investigate whether humans use model-free or model-based approaches. Participants have to choose between two stimuli (shown as symbols) on the first step. The first choice determines the next state (pink or blue) with different probabilities. The last decision step (left or right) determines a reward with varying probability. Participants do not have access to the model of the task and have to infer the transition probabilities to maximise their outcome in this task. Figure adapted from Daw et al. [2011]. (b) Adapted version of the task for rodents by Miller et al. [2017]. Animals first reach the initiation port i, from which they make a decision ii (left or right), which leads them iii to the second stage of the task, indicated by a tone (speakers), either in the purple or orange state iv, with different probabilities (plain and dashed arrows). After initiation iv, rodents reach a final port v, and receive a reward that depends on the first choice in ii. Figure adapted from Miller et al. [2017] with permission. (c) Different types of changes are possible to investigate different important elements of the decision, adapted from Momennejad et al. [2017] with permission. Green colours refer to the first decision step, blue colours to the second and red to the final outcome, in which a monetary reward in $ is given. Reward revaluation simply changes one of the reward values but in fact also changes the optimal solution. Transition revaluation changes the transition matrix so that the options at the choice step (first step) are now inverted. The policy revaluation switches one non rewarded state to a rewarded state. Figure adapted from Momennejad et al. [2017].

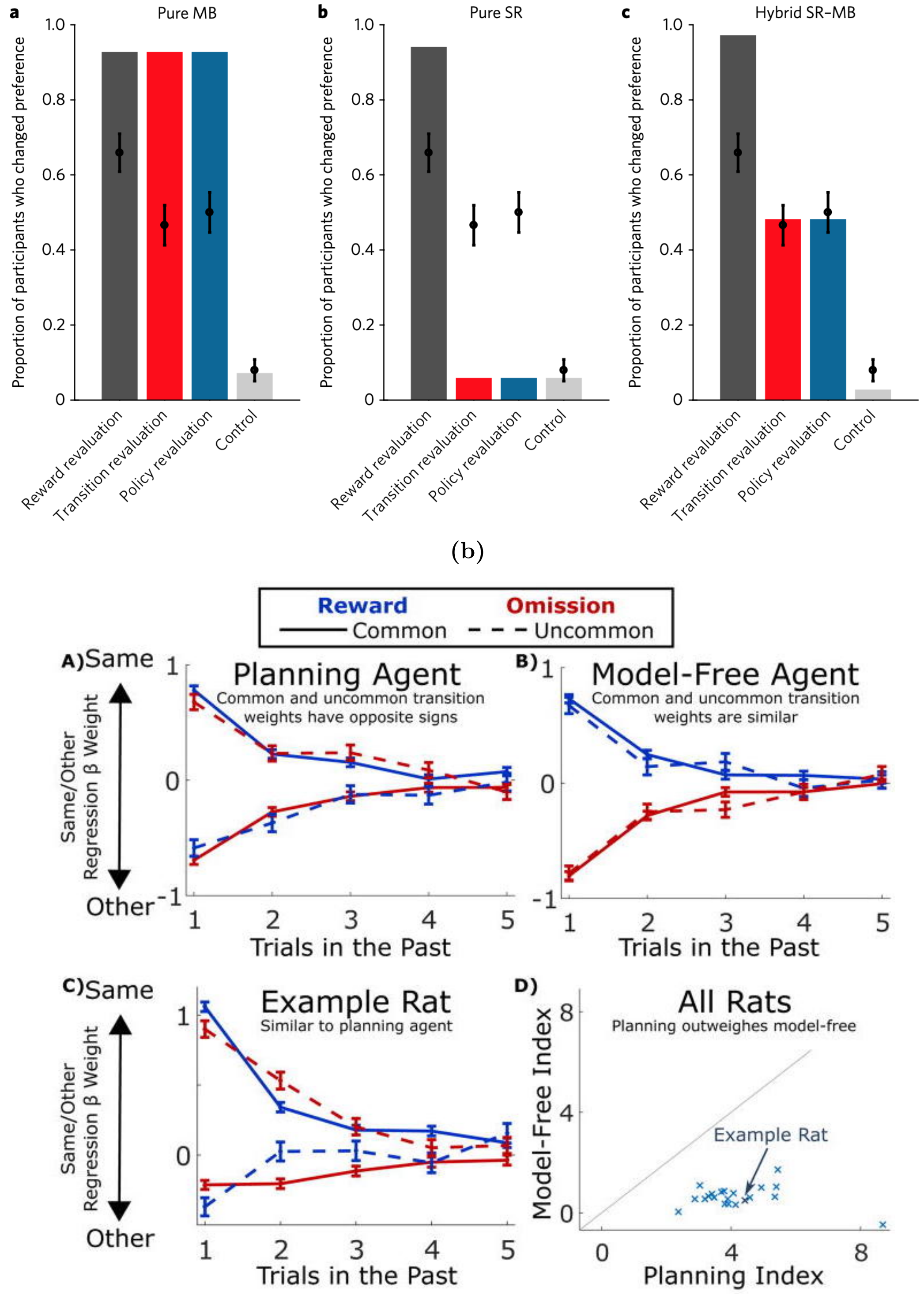
about the adaptation mechanisms of humans and animals compared to those of RL agents (see figures 3.6 and 3.7). Both humans and animals show behavioural correlates of model-based and model-free agents [Yin and Knowlton, 2006, Keramati et al., 2011, Gershman, 2017, Daw et al., 2011, Miller et al., 2017], to differing degrees depending on the paradigm used to assess this comparison (see figure 3.7).

Looking at how human participants and agents adapt to changes in the task enables to assess to which degree participants have access to the underlying model of the task. Figure 3.7 compares the behaviour of rats and humans to different agents in different paradigms based on this task. Momennejad et al. [2017] implemented three types of agent (pure model-based, SR, and a hybrid between model-based and SR which computed the tree-searched values using a mean between the values computed from the SR and the MB), in three types of task configuration (see figure 3.6c). The training phase for this task is shown in figure 3.6c: Momennejad et al. [2017] use a two step decision task that terminates with the reception of one of three different rewards values depending on the options selected at the two decision points. The authors studied and compared the adaptation of participants and of the agents towards three different types of changes within the task. The first modification consists of modifying the reward distribution given at the end of the task: the value of the intermediate reward during the training phase is replaced by a higher value than the best reward ("reward revaluation" in figure 3.6c). During phase 2, the participants learn this new reward distribution without being exposed to the transition anymore. On a second modification, the transitions between the second decision stage and the final states are modified ("transition revaluation" in figure 3.6c). Therefore, choices at the second stage do not lead to the same outcome as during the learning phase. During phase 2, participants are exposed to the second stage of the task and observe the new outcome of their decisions. The last modification used by Momennejad et al. [2017] consists of replacing the previously null-option (leading to a reward of 0) by a very high reward value ("policy revaluation" in figure 3.6c). In all modifications, the best option is no longer preserved, and an optimal adaptation requires changing the decision at step 1. They found that participants' adaptation profiles correspond to those of the hybrid agent. Figure 3.7a depicts their results. The black error bars show that the participants adjust suboptimally to changes. On average, only half of the participants adapted their decision according to the change. A model based agent, because it contains information on both reward and transition probabilities, will adjust perfectly to any change. A pure SR agent, however, will adjust to reward revaluation, but will not be able to adjust to changes in transition and policy. The hybrid is less exhaustive on the full model of the environment but uses replay to compute the SR and make a decision from it. This suggests that humans use a form of trade-off: they do not seem to have access to the full model of the environment, but are better than a SR agent. The difference between pure SR and hybrid SR/MB is that the hybrid agent can improve the estimation of the SR offline but the SR agent cannot, and only builds the SR by interacting with the environment. This result suggests that humans perform a form of nonexhaustive planning search in this task.

Rodents have access to transition probabilities when trained on a two-step decision tasks. Miller et al. [2017] looked at and compared the behaviour of agents and rats from the outcome of different transitions in the two step task (see figure 3.6b). The probabilistic nature of the task generates uncommon and common transition (*i.e.* transitions happening respectively the least and most frequently; in figure 3.6b, for example, 20% of the time for the common transition and 80% of the time for the uncommon one). Ideally, the outcome of uncommon transitions should not influence future decisions, as they are the exception rather than the rule. Instead, an ideal decision should take into account the most likely scenario. Figure 3.7b presents the behaviour analysis of a model-based agent (top left), a model-free agent (top right), and an example rat in the task (bottom left). The behaviour analysis performed by Miller et al. [2017] consists of computing a measure of the likelihood that rats or agents make a similar choice on the current decision that was made on several past trials (in that example, 5), which are referred to as decision weights. Positive weights indicate a greater likelihood to make the same choice, and negative weights a greater likelihood to make the opposite choice [Miller et al., 2017]. A model-based agent, which has access to the exhaustive model of the environment, is likely to repeat choices from common rewarded transition and from uncommon rewarded transition (figure 3.7b, top left). The latter is because after an unrewarded uncommon outcome, repeating the same decision is more likely to lead to a reward. Contrarily, as model-free agents learn through trial and error, they adjust to uncommon transitions regardless of their rarity (figure 3.7b, top right), but only depending on the observed outcome. The bottom panel shows the behaviour of an example rat, closest to a model-based agent. The authors computed a "planning index" and a "model-free index" which represent how well model-free and model-based agent fit to the rats' behaviour, and confirm that, statistically, rodents' behaviour fits better with model-based agents in this task [Miller et al., 2017].

In section 3.2.4, I have already discussed the trade-off between flexibility and computational cost in artificial RL agents (figure 3.5). I described that model-based approaches require the calculation and storage of the transition probability matrix and tree-search computations [Huys et al., 2013]. As the number of states can be very high depending on the complexity of the problem and the precision required, model-based methods are usually computationally costly [Huys et al., 2013]. However, as they contain exhaustive information about the available routes between states, they are more flexible as regards changing goal locations than model-free approaches [Keramati et al., 2011].

A study of spatial navigation in human participants showed that, although paths to the goal were shorter, choice times were higher in trials when the behaviour matches that of a model-based agent compared to trials where it matches that of a model-free agent [Anggraini et al., 2018]. In studies involving rats in a T-maze, vicarious trial and error (VTE) behaviour (short pauses that rats make at decision points, see section 2.3.2) tend to get shorter with repetitive exposure to the same goal location [Redish, 2016]. Experimental studies suggest that VTE behaviour reflects simulations of scenarios of future trajectories to make a decision [Redish, 2016], which would correspond to a model-based approach of task solving [Pezzulo et al., 2017, Penny et al., 2013]. This suggests that model-based strategies, when implemented by humans or rodents, require more processing time than model-free strategies, which is thought to represent "planning"



(a)

(b)

Figure 3.7: Behavioural correlates of a) rodents and b) humans with different models in the 2 steps decision task. a) Momennejad et al. [2017] compared the behaviour of human participants to a model-based (left), SR (middle), and hybrid between SR and model-based (right) agent from different changes within the task. Full bars show the behaviour of the model, error bars show the behaviour of the participants. The results suggest that humans use a hybrid strategy: they do not have access to the full model but show more adaptation than a SR agent. Figure adapted from Momennejad et al. [2017]. b) Miller et al. [2017] have compared the behaviour of rats and agents to rare outcomes. A pure model-based agent should not adjust its policy from the reception of a rare reward (Reward/Uncommon), neither from the omission of a reward after a rare transition (Omission/Common). They show that rats behave the same (bottom left), and that rats' behaviours are best explained by model-based agents on average (bottom right). Figure adapted from Miller et al. [2017].

time [Keramati et al., 2011]. This suggests that brains are concerned with a similar trade-off between flexibility and computational cost than the agents described before (section 3.2.4, see for example figure 3.5).

In order to adjust to the variety and stochasticity of the environments animals live and act in, to the demands of adaptation when danger arises, and to the need for planning that certain tasks require to pursue long-term rewards, brains need different forms of control, gradually distributed between flexibility and efficiency. Computational models have investigated how the control of behaviour could be coordinated between model-free and model-based systems, either depending on uncertainty [Daw et al., 2005], depending on a trade-off between the cost of engaging in complex computations and the associated improvement in the value of decisions [Pezzulo et al., 2013], or depending on how well the different systems perform on a task [Dollé et al., 2018]. Tasks of varying regularity enable one to investigate how the behaviour and its neural representations are shaped by task demand.

When the task gives an illusion of determinism (*i.e.* when the reward, rule and environment are stable across trials), for example when the task is overtrained (*i.e.* when rodents or human participants perform the same task, with the same reward, rule, and environment, for more trials than the number they need to learn the task), the neural representations and the behaviour shift from model-based, purposeful behaviour, to habitual behaviour, which is faster but less flexible to any change [Smith and Graybiel, 2013]. Dezfouli and Balleine [2019] investigated choices profile of rats placed in an evolving multi-step decision task. In the early stages of the task, rats had to learn very simple associations between lever presses and rewards. Progressively, another decision step was introduced, so that rats had two decision stages. At this point already, in order to learn the task, rodents have to adapt from a task configuration that immediately rewards choices, to a more complex task design, which requires animals to adjust their decision based on a delayed feedback. After this phase, Dezfouli and Balleine [2019] then introduced randomness in the decision steps. They observed that the rats' decisions progressively reflected the multi-step and stochastic aspect of the task. This adaptation suggests that the state and action representations that enable solution of a task seem to evolve dynamically on the timescale of learning to adjust to the task requirements [Dezfouli and Balleine, 2019]. When the situation is inherently stochastic, the neural representations and behaviour adjust according to the uncertainty of the environment [Tomov et al., 2020, Akam et al., 2020]. During learning in a multi-step decision-making task, experiments suggest that neural representations evolve to incorporate the multi-step dependencies, and simultaneously prune the tree of possible outcomes depending on the most likely scenarios [Miller and Venditto, 2021, Huys et al., 2012, Van Opheusden et al., 2017]. The adaption of neural representations and behaviours to task demand suggests that a continuum of state and action representations for behavioural control, between the two extremes of model-based and model-free systems, exists in the brain [Dezfouli and Balleine, 2019]. This seems to also apply to the control of the behaviour in the example of the watermaze task that this thesis focuses on, which is discussed in the next section.

3.3.2 Neither purely model-free nor model-based approaches are realistic candidates for rapid place learning in the watermaze

The rapidity with which rats adjust to a changing goal location in the DMP task (one exposure only, as discussed in section 2.5.3) indicates that some kind of "model" is being used to enable adaptive route selection. The size of the graph to model the environment, necessitating a high number of states for fine discretisation, and the width and depth of the trees to search at possible decision points (for example, at the start location) that would be required to account for such behaviours suggest that the control in this task cannot be accounted for by a traditional model-based agent. Moreover, rats do not reach optimal trajectories in the DMP task, but the escape latencies between trial 2 and 4 remain higher than would typically be observed following incremental place learning in the watermaze (figure 3.8), which would correspond to the asymptotic performance of the rats, and than would be expected from a model-based agent [Sutton and Barto, 2018].

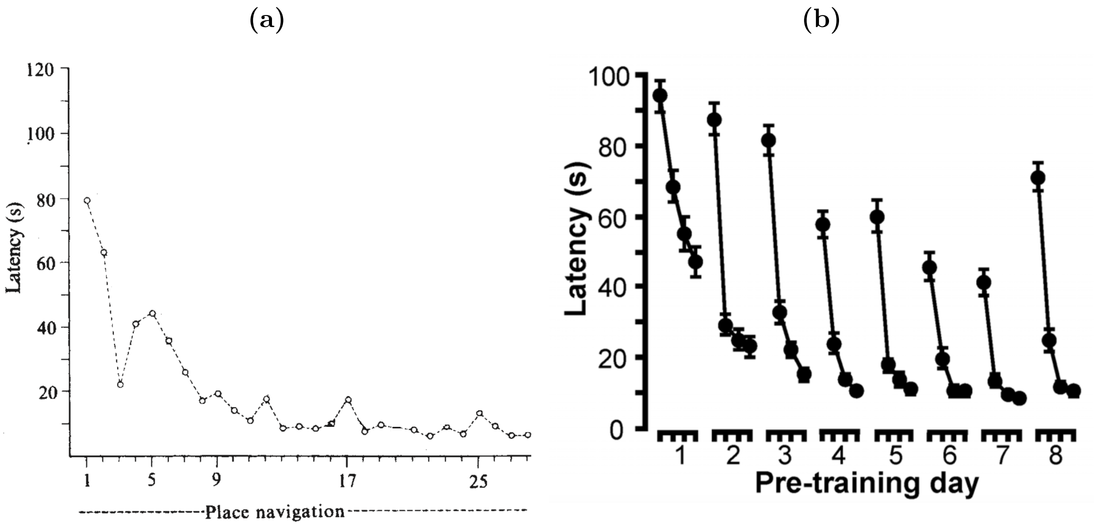


Figure 3.8: Comparison of performances after repeated exposure to the same goal location (a) and on the DMP task (b). After extensive training (a), rats reach an average latency that is lower than the latencies observed in the second, third and fourth trials to a new goal location in the DMP task (Figure adapted from Morris et al. [1982]) (b), in which an improvement can be observed between the second, third and fourth trial across days, with the latencies on the fourth trial reaching a similar value, but still slightly higher, than the value of the latency obtained after many days of training to the same goal location (Figure adapted from Bast et al. [2009]).

In humans, Anggraini et al. [2018] suggest that participants that used more model-based approaches more often took the shortest path to goal locations. This suggests that the control in the watermaze DMP task cannot be explained by a model-free RL mechanism alone because of its lack of flexibility, but also is unlikely to be fully model-based.

3.4 Conclusion

In this chapter I have introduced how RL maps states to actions, and can be used in a spatial navigation context to select directions to a goal, based on the reception of a reward. Three main lines of approach in this field have been described and compared. *Model-free* agents learn on the fly, are very fast but are not flexible. *Model-based* agents learn or are given exhaustive information about the transition of the states and the reward from which they can compute the shortest paths ahead. They are very flexible but the search requires a high computational cost. In between, a *Successor Representation* agent can learn the long term-statistics of the environment that enable a faster planning process, and are less computationally costly than a model-based agent but more flexible than a model-free agent.

Comparing behaviours between an artificial agent and human/rodent participants on a similar task enables one to investigate the trade-off between flexibility and efficiency that brains operate on and the degree to which brains encode the transition probabilities of the environment. One beautiful example of this applies to games that require planning ahead, such as chess or tic-tac-toe, in which one can extract the depth of the tree that humans on average are able to search ahead, among other cognitive measures [Van Opheusden et al., 2017]. The time that the brain takes to make a decision seems to correlate with the computational representation that best explains the decision. Investigating the trade-off between flexibility and computational cost can be informative about the type of computations the brain uses to perform certain behaviours. In the particular case of the watermaze task, the adaptation to the new goal location is fast, one trial only, but at the cost of a less optimal trajectory, which suggests that similar constraints also apply to this task.

In this thesis, I present different agents that learn to navigate to a goal location mimicking the platform in the watermaze experiment and investigate their mechanisms of flexibility. I start by presenting a model-free agent and discuss its performance, biological implementation, and limitations in the following chapter 4.

Chapter 4

Actor-critic architectures in discrete time for spatial navigation in the Morris watermaze

4.1 Introduction

This chapter presents a modelling framework that allows the investigation of how place cells can be used to perform spatial navigation towards a goal location using an RL framework. I implemented an actor-critic architecture published in Foster et al. [2000] used to perform spatial navigation in a virtual watermaze, towards fixed and changing goal locations. This chapter includes the implementation of the model proposed in Foster et al. [2000] and the reproduction of their results. It contains an analysis of the influence of important parameters that shape the behaviour of the agent implemented from the model, through which I confirm findings already known in the literature in the particular example of the model considered here. Finally, I propose a novel discussion over the biological realism of the models architecture in light of recent experimental findings.

The chapter presents two different approaches, both using place cells to represent the position of the agent. The biological realism of these approaches is discussed, the performances are compared to the ones of rodents in the equivalent watermaze tasks, and the architecture is contrasted with experimental findings concerning the neural substrates involved in learning in the watermaze or other spatial navigation tasks.

I first consider a classic actor-critic model for navigation to a fixed goal location [Foster et al., 2000]. In an actor-critic approach, a TD error is used to simultaneously learn the value function over the maze, which estimates the future reward expectations of an agent in the space, and the best action to choose in every location based on place cell inputs, used to perform navigation. I reproduce findings from Foster et al. [2000] which show that the actor-critic agent successfully learns how to navigate to a fixed platform location within only a few trials. I extend their work in providing a detailed analysis regarding important mechanisms which influence learning by comparing the results of the implementations of the model for different values of influential parameters. In particular, I show that the scale of the spatial representation influences the speed of learning and the precision of the final performance, which is consistent with results of the literature implicating different divisions of the dorsoventral hippocampus in incremental and rapid-place-learning in the Morris watermaze.

I show that the model is partly biologically realistic. The contribution of actor-critic mechanisms to DMP performance is consistent with neurobiological findings implicating the striatum and hippocampo-striatal interaction in DMP performance, given that the striatum has been associated with actor-critic mechanisms.

I confirm the limitations of the actor-critic approach raised by Foster et al. [2000] to account for rapid place learning in the Morris watermaze. When the platform position changes, the agent takes many trials to adjust to this new location, contrary to rats and humans in the DMP task, who only necessitate one visit to a new goal location to remember how to get back to it on the subsequent trials [Steele and Morris, 1999, Bast et al., 2009, Buckley and Bast, 2018].

To adjust to the changing goal locations of the DMP tasks, Foster et al. [2000] adapted the architecture. They propose that a goal-independent representation of space through a 'map-like' representation of locations over the space can be used to compute vector-based displacement towards goal locations. I implement their approach in a watermaze DMP task and reproduce their findings that coordinate-based goal-directed navigation leads to flexibility towards changing goal locations [Foster et al., 2000]. The approach consists in learning coordinates within the space using a similar TD update measuring errors in self-motion estimation. Then, the coordinates are used for goal-directed displacement. The agent adjusts to changes in goal locations in the DMP task. I present and discuss the performance of the agent compared to the ones of rodents in the watermaze. I extend the approach to represent other behavioural measures relevant to spatial learning in the Morris watermaze. I discuss the biological realism of such implementation, which I show is consistent with recent findings of goal and goal-vector correlates in neuronal activities.

4.2 An actor-critic architecture maps locations to directions

In this section, I present an actor-critic architecture in which units mimicking place cells are used as a basis for learning the value function and the policy around the space using TD learning.

Foster, Morris, and Dayan [2000] proposed an actor-critic framework to perform spatial learning in a watermaze equivalent (see figure 4.1 for a schematic of the model architecture). The agent's location is represented through a network of N_P units mimicking hippocampal place cells, which have a Gaussian activation profile around their preferred location (the further the agent is from the place cell's preferred location, the less active the unit will be):

$$f_i(p_t) = \exp\left(-\frac{||p_t - c_i||^2}{2\sigma^2}\right), \text{ for } i = 1 \dots N_P. \tag{4.1}$$

Here $p_t = (x_t, y_t) \in \mathbb{R}^2$ represents the position of the agent at time t, $c_i \in \mathbb{R}^2$ denotes the centre of the ith place cell and σ the width of the activity profile. σ determines the overlap between place fields. Each place cell projects to an actor network and a critic unit through plastic connections (respectively denoted Z_t and W_t in figure 4.1).

The estimation of the value of the current state is provided by a critic cell, whose

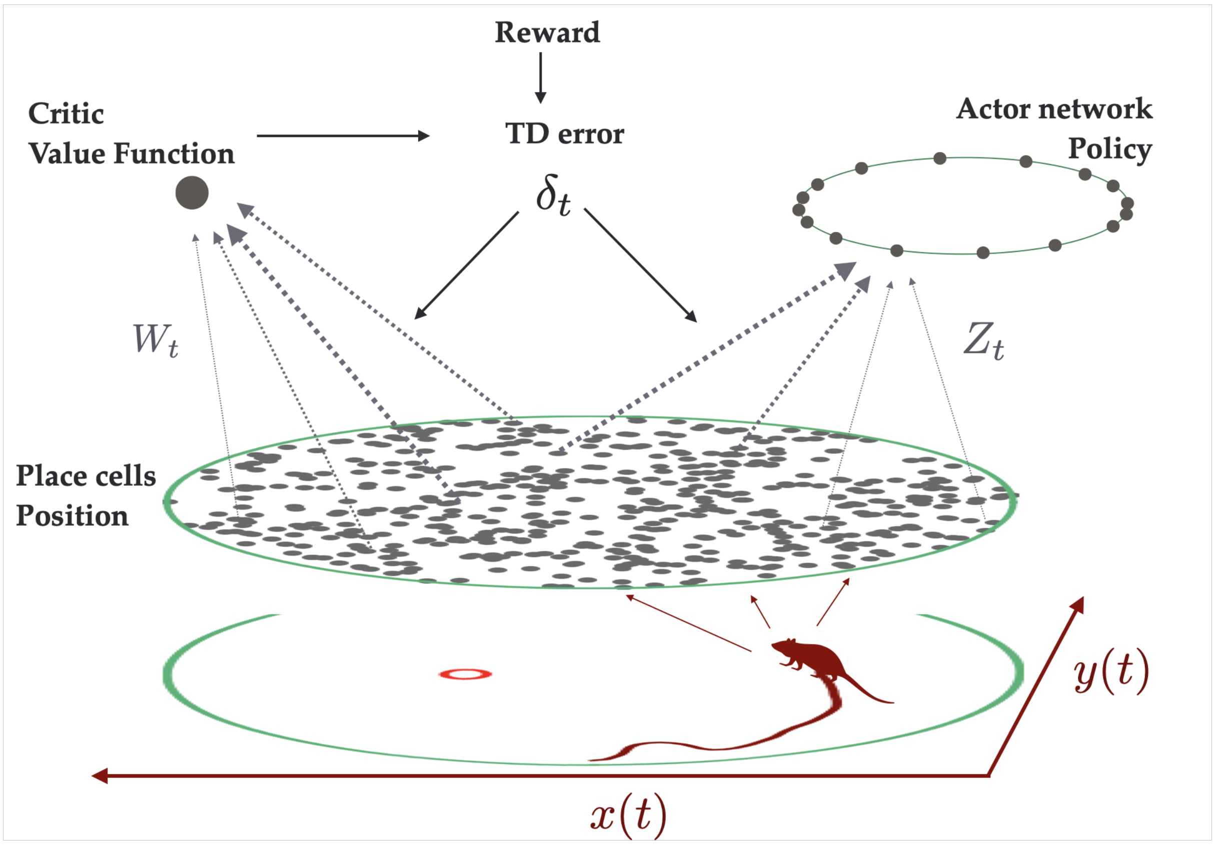


Figure 4.1: Classical actor-critic architecture for a temporal-difference (TD) agent learning to solve a spatial navigation problem in the watermaze, as proposed in Foster et al. [2000]. The state (location of the agent) is encoded within a neural network (in this case, the units mimic place cells in the hippocampus). State information is fed into an action network, which computes the best direction to go next, and to a critic network that computes the value of the states encountered using feedforward weights W_t and Z_t. The difference in critic activity along with the reception or not of the reward (given at the goal location) are used to compute the TD error δ_t, such that successful moves (that lead to a positive TD error) are more likely to be taken again in the future, and less likely otherwise. Simultaneously, the critic's estimation of the value function is adjusted to be more accurate.

activity is computed according to:

$$C(p_t, W_t) = \sum_{i=1}^{N_P} W_t^i f_i(p_t), \tag{4.2}$$

where W_t^i is the weight from the ith place cell to the critic cell. From its current position p_t and at every subsequent time step the agent chooses a direction according to the activity of "action cells" that receive inputs from the place cells. The actor network is composed of $N_A = 8$ units, which represent 8 possible directions (north, north-east, east, south-east, south, south-west, west and north-west). The activity of the jth action cell is computed according to:

$$a_j^t(p_t, Z_t) = \sum_{i=1}^{N_P} Z_t^{ji} f_i(p_t), \quad \text{for} \quad j = 1 \dots N_A, \tag{4.3}$$

where Z_t^{ji} is the connection weight form the ith place cell to the jth action cell. To determine the probability of choosing the jth action, given the activities a_j, a softmax probability distribution of the activities is computed according to:

$$P_j = \frac{\exp(\beta a_j)}{\sum_{k=1}^{N_A} \exp(\beta a_k)}. \tag{4.4}$$

In equation (4.4), the parameter β acts as an inverse temperature. Small values lead to a more uniform probability distribution, increasing randomness, and therefore promoting exploration of the environment, while larger values lead to more exploitation of known state-action pairs.

After the selection of the jth action according to the previous probability distribution, the agent moves according to:

$$p_{t+1} = p_t + \alpha \begin{bmatrix} \cos(\theta_j) \\ \sin(\theta_j) \end{bmatrix}, \tag{4.5}$$

where θ_j denotes the angle of the selected jth action cell, and α is the speed of the agent. After having moved to the location p_{t+1}, the agent receives a reward $r_t = 1$ if it has reached the goal location (representing the platform in a watermaze experiment), and

$r_t = 0$ otherwise. The agent can then compare the value of its current state against its prediction through the following TD error:

$$\delta_t = r_t + \gamma C(p_{t+1}, W_t) - C(p_t, W_t), \tag{4.6}$$

where $C(p_{t+1}, W_t)$ refers to the estimated value of the current state p_{t+1} from the weights at the previous time t (see equation (4.2)).

The TD error (equation (4.6)) is used to improve the estimator by updating the weights from place cells to the critic (equation (4.2)) via

$$W^i_{t+1} = W^i_t + \Delta W^i_t, \tag{4.7}$$

using:

$$\Delta W^i_t \;=\; \chi_{\mathrm{C}}\; \delta_t f_i(p_t), \tag{4.8}$$

where χ_{C} is the learning rate. This learning rule guarantees that the value associated to the current state s_t increases (respectively decreases) if the following state has a higher (respectively lower) value, as characterised by a positive (respectively negative) δ_t. The actor network is also updated to improve the policy. The actor weight Z^{ji} between the jth action cell and the ith place cell evolves according to

$$Z^{ji}_{t+1} = Z^{ji}_t + \Delta Z^{ji}_t, \tag{4.9}$$

using:

$$\Delta Z^{ji}_t = \chi_{\mathrm{A}} \delta_t f_i(p_t) g_j(t), \tag{4.10}$$

where χ_{A} is the learning rate, $g_j(t) = 1$ if the jth action is chosen and is 0 otherwise. The connection between a place cell and an action cell is strengthened if choosing the action at the given location results in an increase in the value function, and is weakened if this results in a decrease in the value function. In the next section, I present the results of the implementation of this model in a watermaze equivalent in which the goal stays at the same location. All parameters, implementation details and link to corresponding code are provided in Appendix A. The values of the connection weights (4.8) and (4.10) are initialised at 0. The conditions of convergence have not been investigated in this thesis, rather, I was interested in the behaviour produced by the model over a few trials. The conditions for convergence on actor-critic methods are developed in Konda and Tsitsiklis [2000].

4.3 The architecture enables navigation to a goal location

The agent can learn to navigate to a fixed goal location. Figure 4.2 shows the escape latencies of the agent over repeated trials to the same goal location. Within a few trials, the agent reaches asymptotic performances. The TD error contains two pieces of information: it reflects both how good the decision that has just been made is, and the accuracy of the critic at estimating the value of the state. The update of connection weights W_t and Z_t via the learning rules defined by equations (4.8) and (4.10) improves the actor and the critic: it updates the connection weights according to the current error and the place cell activity, such that the probability of making the same decision in the same location increases if it leads to a new position that has a higher value, and decreases otherwise. As the connection weights are initialised at 0, the first trial corresponds to a random walk. Hence, the first trial is on average very long, and the agent is placed at the goal location if it does not find it by itself within 120s. After the first trial, the agent has received a reward, therefore is not completely blind about the reward location anymore, which explains the drastic reduction in latency between the first and second trial. Multiple ways to obtain learning rules can be used, for example, see Doya [2000].

Learning can also be seen in the value function that becomes maximal around the goal location, and in the action map that indicates the direction of the goal. Figure 4.3 compares the different features of the model before and after learning. Before learning,

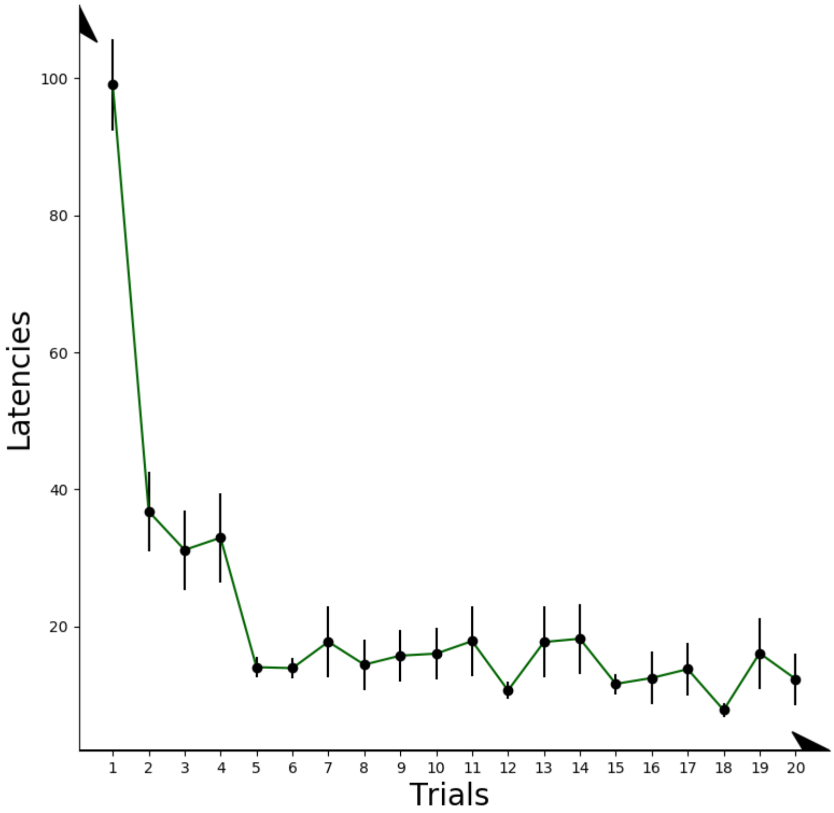


Figure 4.2: Latencies of the agent, obtained by implementing the model from Foster et al. [2000]. The time that the agent requires to get to the goal ("Latencies", vertical axis) reduces with trials (horizontal axis) and reaches almost a minimum (after trial 5). Implementation details and parameter values are given in Appendix A.

the value function (Figure 4.3a.i) is initialised at zero, the actor weight too, leading to a random action map (Figure 4.3a.ii), and to a random trajectory (Figure 4.3a.iii). After learning, the value function is maximal around the goal location and smoothly decays towards the edges of the maze (Figure 4.3b.i), the actor network is more tuned to a particular direction (Figure 4.3b.ii), and this leads the agent to take a direct path to the goal (Figure 4.3b.iii). In figure 4.3b.iii, the trajectory is not direct to the goal. This could be due to the fact that the agent has only been trained during 20 trials and therefore the behaviour has not reached an optimal yet, or that the width of the place fields is too high for the behaviour to be precise. Note that the maximum of the value function should be located at the goal location, but because the continuous place representation induces spreading of the value function, its maximum appears slightly on the side of the value function.

In this section, I have replicated the findings from Foster et al. [2000] showing that the actor-critic agent learns to navigate to a fixed goal location. In the following section, I provide a detailed analysis of the important parameters that regulate the behaviour of the agent by comparing the performance for different values of these parameters. In particular, the scale of spatial representation influences the speed of learning and the precision of the trajectories obtained, and I confront my results to experimental results implicating different areas of the dorsoventral axis of the hippocampus in learning in the watermaze. I also discuss an addition to a TD method used in the literature to accelerate learning, which I relate to the discussion about the width of place representation.

4.3.1 Important variables that shape the behaviour, and ways to speed up learning

In this section, I analyse the effect of the spatial scale and the discount factor on the performance of the agent. This discussion can also be found in existing papers of the literature, for example see [Doya, 2002, Gustafson and Daw, 2011, Sutton, 1996]. I discuss how the spatial scale influences generalisation of experience across the state space and the asymptotic performance of the agent. I confront these results to experimental

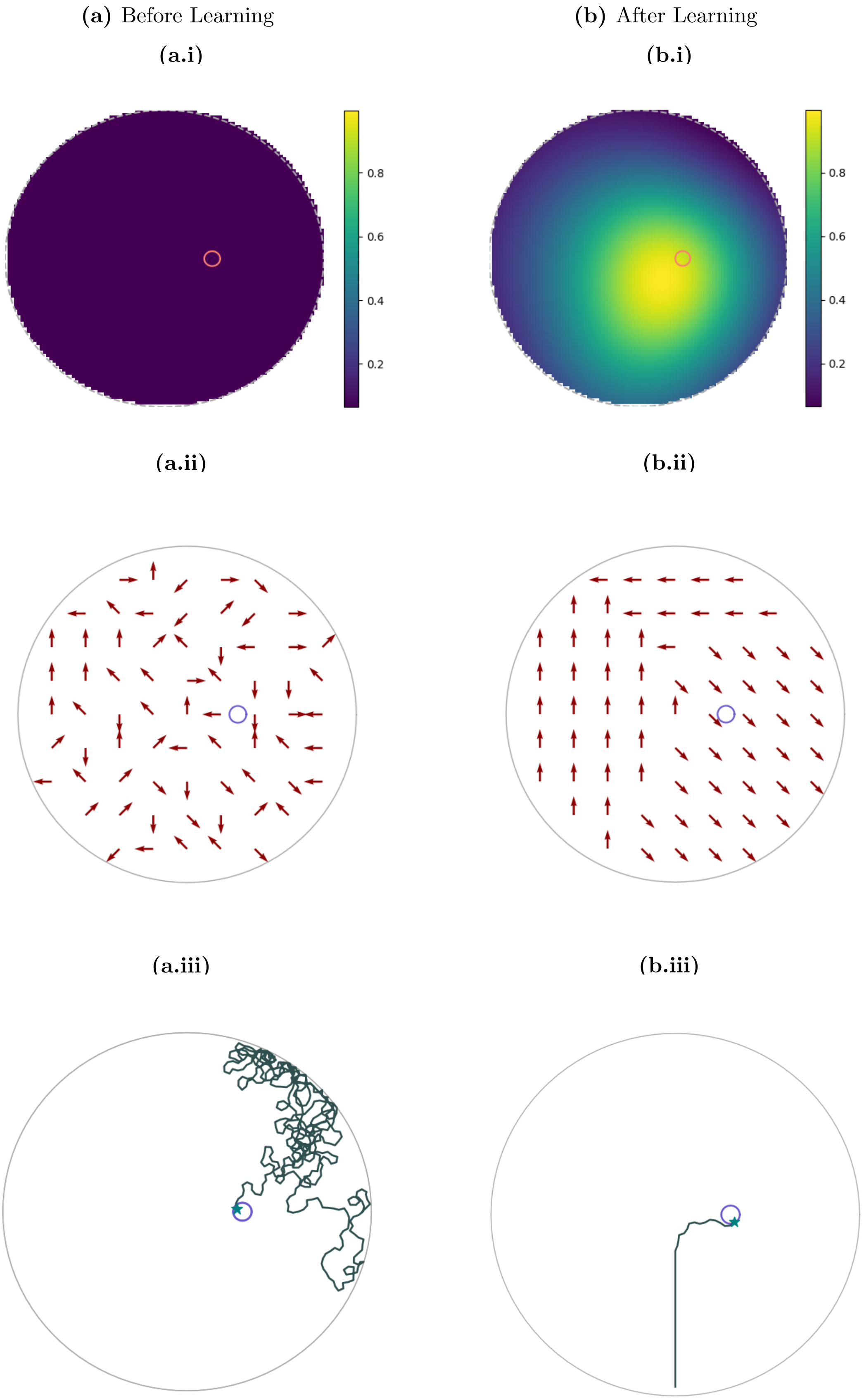


Figure 4.3: The value function and action map are shaped with learning. a) Before learning, the value function (a.i) is initialised as zero, the actor weight too, therefore leading to a random action map (a.ii), and to a random trajectory (a.iii). b) After learning, the value function is maximal around the goal location and progressively decreasing towards the edges of the maze (b.i), the actor network is more tuned to a particular direction (b.ii), and this leads the agent to take a direct path to the goal (b.iii). (a.i) and (a.ii) have been implemented from equation (4.2), (a.ii) and (b.ii) from equation (4.4).

results which implicate different sections of the dorsoventral axis of the hippocampus for incremental and one-shot learning in the Morris watermaze. I additionally discuss another method commonly used to accelerate learning in which past decisions are updated as a result of the current observations [Sutton and Barto, 2018], and I discuss the link between the temporal scale of these updates and the width of the place representation.

4.3.1.1 Place cell representations enable generalisation, and discount factor value propagation

The previous section illustrated how place cells can be integrated into a network for spatial navigation through the association of values and actions formed through TD learning. A benefit of this approach is that i) it allows for relatively fast learning, within only a few trials, and ii) the agent obtains information about which action could lead to the goal from variable and distal start positions. These two properties rely on two major components.

First, the TD error allows to "backpropagate" value information from successive states, with the speed of this backpropagation modulated by the discount factor γ. The update of the estimated state value $C(p_t, W_t)$ (4.2) and of the policy (4.3) at the position p_t, after having moved to the new position p_{t+1}, depends on the TD error (4.6). Let us consider the latter, defined by equation (4.6): the TD error is the difference between the received reward and the estimated discounted value at the next state (given by $r_t + \gamma C(p_{t+1}, W_t)$, the first two terms in equation (4.6)), and the value at the current state ($C(p_t, W_t)$, the last term in equation (4.6)). Note that the update of the state value is performed "in the future" for "the past": the value underlying the decision taken at the time t will only be updated at the time $t+1$. Moreover, the extent to which the future is taken into account is modulated by the parameter γ.

Let us consider extreme values. The sum (3.1) is defined for $0 \leq \gamma < 1$. In the case $\gamma = 0$, the agent is completely myopic and only learns about actions that produce immediate reward. The closest γ is to 1, the more future sight the agent has about future rewards. Ideally, the discount factor γ should be adjusted to maximise the slope of the value function, so that the policy is "concentrated" on the optimal choice, and to obtain

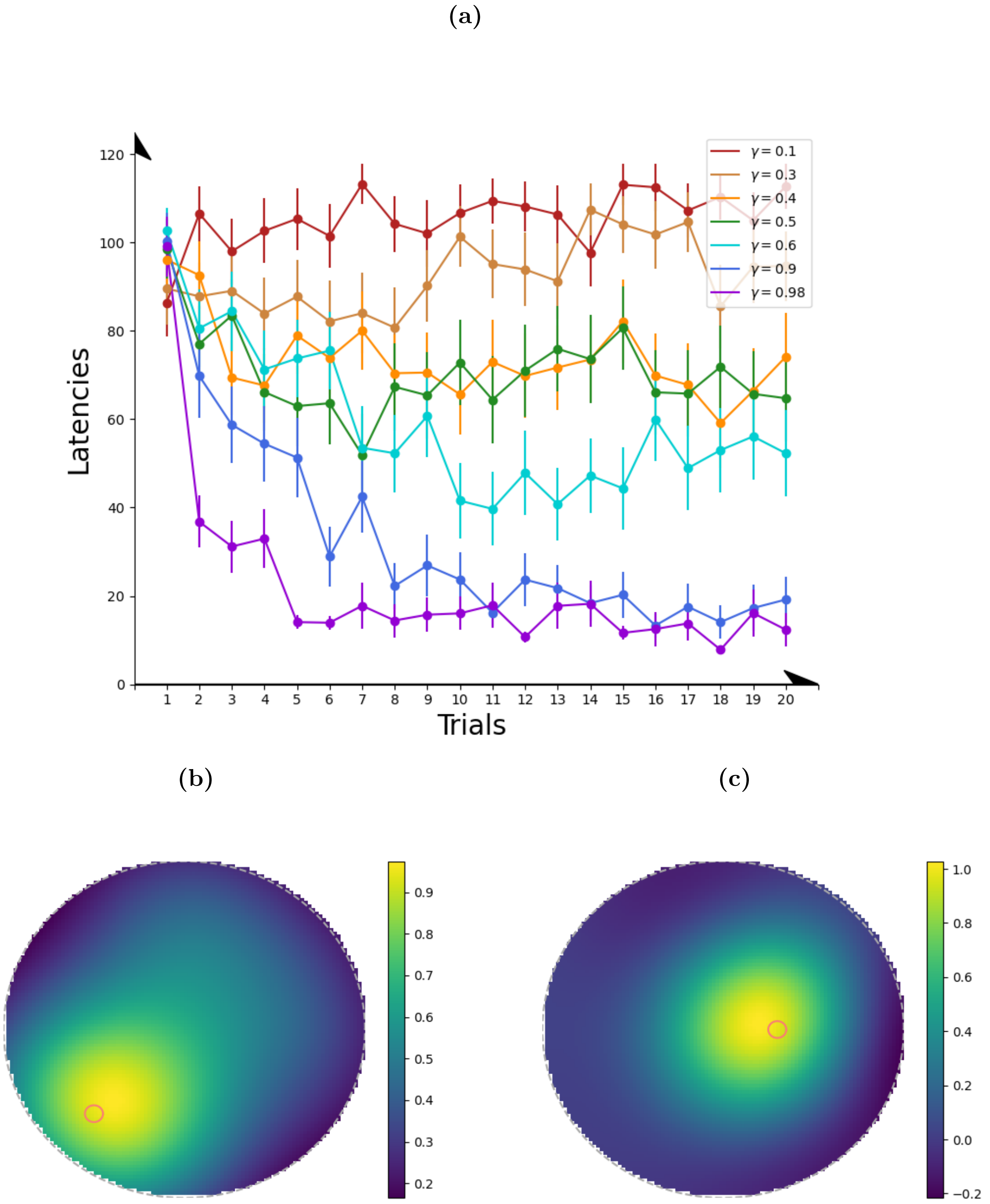


Figure 4.4: Effect of the discount factor on learning. (a) Latencies obtained on 20 independent runs for different values of γ. (b,c) Value function for (b) a low value of γ and (c) a high value of γ. A low γ leads to a narrower value function. The agent is not able to backpropagate to previous states about the value of the future state. This leads to the agent not having enough information about optimal actions at the borders of the maze, and prevents learning, as can be seen in the latencies (a).

a uniform slope across space, as this guarantees that the agent has good information on which to base its decision from any location within the environment. In general, if the discount factor is too high for a certain task, risks are that non-relevant outcomes are taken into account to update the decision. One can consider a simple scenario to generate a formula for the discount factor with respect to the number of states. I consider here an environment with N states, and I write the discount factor as $\gamma = \mathrm{e}^{-1/\tau}$, with $\tau > 0$ a constant representing the discount rate per timestep. The sum of discounted future rewards, also known as return, and sometimes written as G_t [Sutton, 1991], averaged in the value function (3.1), can be written:

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} = \sum_{k=0}^{\infty} \mathrm{e}^{-k/\tau} r_{t+k}. \tag{4.11}$$

This expression gives an intuition on the link between the scale of the environment and the discount. Rewards that are much further away than τ timesteps from the decision will be more discounted than rewards which are closer. For the agent to take into account rewards at the end of a trajectory in its environment, τ should be of the order to the number of states N, which leads to $\gamma > \mathrm{e}^{-1/N}$. Figure 4.4 illustrates the effect of changing γ on the latencies and the value function.

The second parameter that strongly affects speed and precision of place learning is the spatial scale of the place cell representation, determined by the width of the neurons' place fields. The state representation through place cells enables the generalisation of learning from a single experience across states, *i.e.*, to update information on many locations based on the experience within one particular location. Every update amends the value and policy for all states depending on the current place cell activity, such that more distal locations are less concerned by the update than proximal ones. The spatial reach of a particular update increases with the width of the place cell activity profiles. This process speeds up learning, because when the agent encounters a location with a similar place cell representation to those already encountered, the prior experiences have already shaped the current policy and value function and can be used to inform subsequent actions.

If the width is very small, the agent cannot generalise enough from experience, and

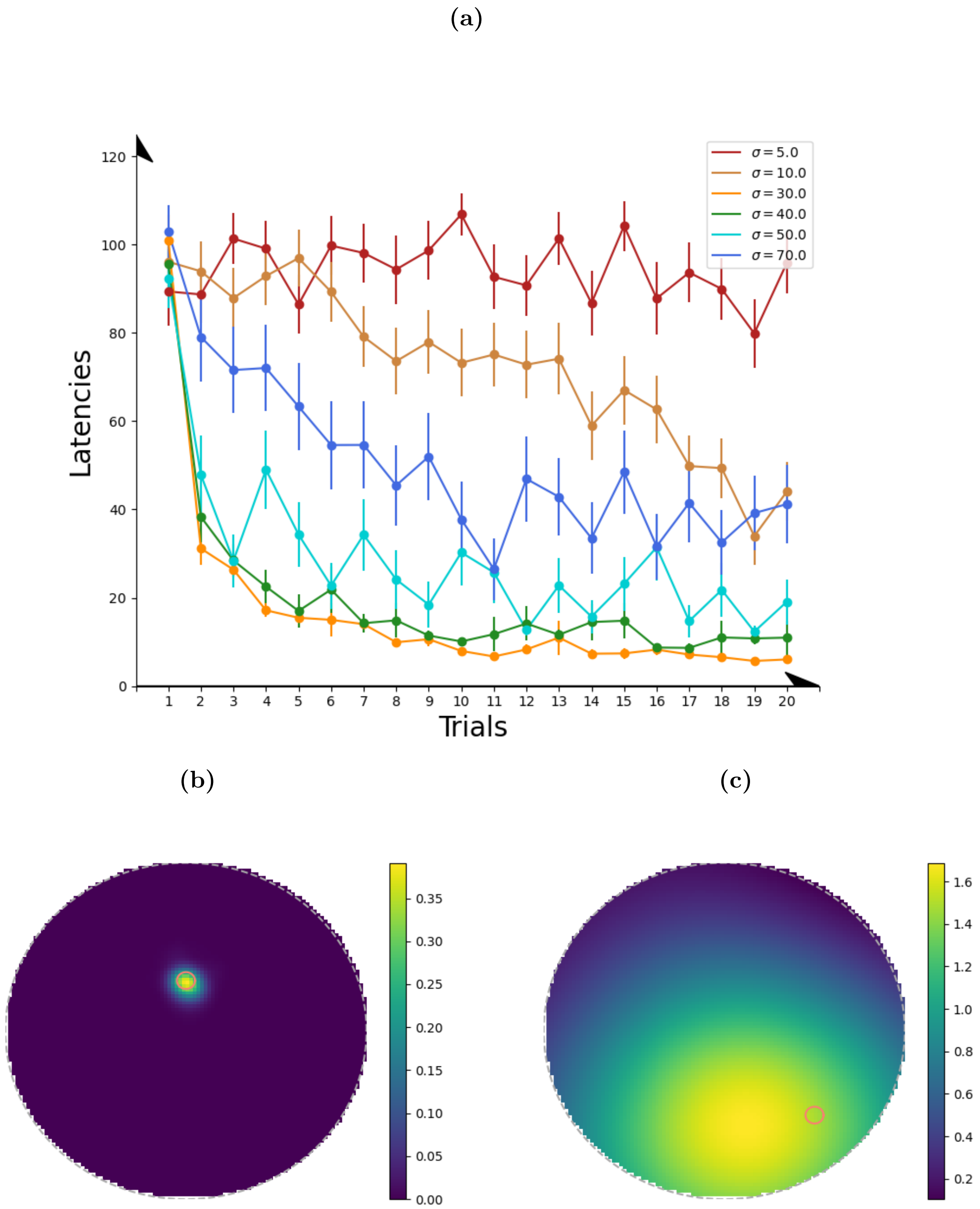


Figure 4.5: Effect of changing the width of the place cell activity distribution σ on learning. (a) Latencies obtained on 20 independent runs for different values of σ. σ should be chosen intermediary depending on the size of the maze and of the goal. (b,c) Value function for (b) a very low value of σ and (c) a high value of σ. (b) A low value of σ leads to a narrow value profile and doesn't permit enough generalisation of experience. (c) A too high value of σ leads to a less sharp increase in the value function.

this considerably slows down learning, as the agent must comprehensively search the environment in order to learn. At the opposite extreme, if the activity profile is very wide, generalisation occurs where it is not appropriate: for example at opposite ends of the goal, when the best actions to choose would be opposite to each other, as at the North end it would be best to go South, whereas at the South end it would be best to go North. Therefore, a high spatial scale enables faster learning thanks to enhanced generalisation, but a too high spatial scale compromises learning because of interference around the goal location. The optimal width lies in a trade-off between speed of learning and precision of knowledge: it should be scaled to the size of the environment to speed up learning and is constrained by the size of the goal.

Different widths of place cell activity profile have different advantages for learning. A recent RL model suggests that smaller scales of representation would support the generation of optimal path length, whereas larger scales would enable faster learning, defining a trade-off between path optimality and speed of learning [Scleidorovich et al., 2020]. Figure 4.5 shows that a wide activity profile of place cells leads to suboptimal routes in the watermaze task with a fixed goal location, characterised by escape latencies that stay high.

Place cells in the intermediate and dorsal hippocampus could have different roles in supporting place learning in the watermaze, with the intermediate hippocampus providing a flexible trade-off between accuracy and efficiency, and the dorsal hippocampus more involved in supporting shorter and well-learned trajectories. Along the hippocampal longitudinal axis, places are represented over a continuous range of spatial scales, with the width of place cell activity profiles gradually increasing from the dorsal (also known as septal) towards the ventral (also known as temporal) end of the hippocampus in rats [Kjelstrup et al., 2008]. Bast, Wilson, Witter, and Morris [2009] found that the intermediate hippocampus is critical to maintain DMP performance in the watermaze, particularly search preference. Moreover, the trajectories employed by rats in the watermaze DMP task are suboptimal, *i.e.*, path lengths are higher, compared to the incremental learning task (see figure 3.8). Place cells in the intermediate hippocampus have place fields of an intermediate width and, thereby, may deliver a trade-off between fast and precise learning. These findings may partly reflect that they might be particularly important for navigation performance during the first few trials of learning a new place, as on the DMP task. Another potential explanation for the importance of the intermediate hippocampus is that this region combines sufficiently accurate place representations, provided by place cells with intermediate-width place fields, with strong connectivity to prefrontal and subcortical sites that support the use of these place representations for navigation [Bast et al., 2009, Bast, 2011], including striatal RL mechanisms [Humphries and Prescott, 2010]. With incremental learning of a goal location, spatial navigation can become more precise, with path lengths getting shorter and search preference values increasing [Bast et al., 2009]. Interestingly, the model by Scleidorovich et al. [2020] suggests that such precise incremental place learning performance may be particularly dependent on narrow place fields, which are shown by place cells in the dorsal hippocampus [Kjelstrup et al., 2008]. This may help to explain why incremental place learning performance has been found to be particularly dependent on the dorsal hippocampus [Moser et al., 1995].

In this section, I have shown that the learning rate enables the propagation of reward information in the direction of motion. I have discussed that it should be chosen maximal but less than 1. I additionally discussed that the width of the place cell activity profile enables to fasten learning via generalisation and is linked to the precision of the asymptotic trajectory obtained. In the case of navigating to a fixed goal location, it should be chosen optimally as a trade-off between the scale of the maze and the size of the goal. This computational analysis is consistent with experimental results showing that different sections of the dorsoventral axis of the hippocampus are involved differently in incremental and flexible learning in the Morris watermaze. While place cells enable generalisation, using eligibility traces, which permit the updating of past decision mechanisms based on the outcome of the current experience, also enables to speed up learning.

4.3.1.2 Using eligibility traces allows the updating of past decisions according to the current experience

This section briefly introduces the concept of eligibility traces and links the choice of the timescale of the eligibility trace to the width of the place representation.

Using eligibility traces enables the updating of past decision mechanisms from current observations. The particular actor-critic implementation proposed in Foster et al. [2000] and described in section 4.2 involves a one-step update: only the value and policy of the state that the agent just left are updated. However, past decisions usually affect current situations, and one-step updates only improve the most recent choice and estimate. This can be addressed by incorporating past decisions when performing the current update, weighted according to an *eligibility trace* [Sutton and Barto, 2018]. Eligibility traces keep a record of how much past decisions influence the current situation. This makes it possible to update value functions and policies from previous states of the same trajectory according to the current outcome.

The decay rate at which the past should be concerned by the present outcome is linked to the width of the place representation. The most commonly known example of the use of eligibility traces is TD(λ) learning, which updates the value and policy of previous states within the same trajectory according to the outcome observed after a certain subsequent period, weighted by a decay rate λ. The parameter λ refers to how far in the past the current situation affects previous states' value and policy. One extreme is TD(0) in which the only step updated is the one where the current decision was made, as described in section 4.2. When λ increases toward 1, more events within the trajectory are taken into account, and the method becomes more reminiscent of a Monte-Carlo approach [Sutton and Barto, 2018], where all the states and actions encountered during the trial are updated at every step. In Scleidorovich et al. [2020], the authors show that the optimal value of λ depends on the width of the place cell activity distributions: for wide place fields, adding eligibility traces does not greatly speed up learning (*i.e.*, reducing the number of trials needed to reach asymptotic performance), but it does for narrower place fields.

In this section, I have discussed, using specific examples, how the behaviour of the actor-critic architecture presented in section 4.2 can be influenced by the choice of parameters. In particular, I have explained that the speed of learning can be influenced by the discount factor and the width of the place cell activity profiles. I have shown that experimental results are consistent with computational analysis implicating narrow place fields in learning precise trajectories and wide place fields in providing more flexibility via generalisation. I have shown that using eligibility traces can fasten learning and that an adapted decay rate should be chosen depending on the width of the place representation. In the following section, I confront the architecture of the model to existing experimental results of spatial navigation and learning to discuss its biological realism.

4.3.2 The model is partly in line with experimental findings

In this section, we will discuss the biological realism of the model, and explain that aspects of the model are partly consistent with experimental results implicating the striatum as an actor-critic and dopamine for TD learning. However, the model also contains aspects that are inconsistent with experimental literature, in particular, the TD error should influence other plasticity mechanisms and hippocampal plasticity mechanisms should be accounted for.

4.3.2.1 Striatal and dopaminergic systems as candidate substrates for the actor and critic components

In the RL literature, the ventral striatum is often considered to play the role of the "critic" [Humphries and Prescott, 2010, Khamassi and Humphries, 2012, O'Doherty et al., 2004, van Der Meer and Redish, 2011]. The tonic firing of neurons in the ventral striatum increases when rats approach a goal location in a T-maze [van Der Meer and Redish, 2011], consistent with the critic activity representing the value function in actor-critic models of spatial navigation [Foster et al., 2000, Frémaux et al., 2013]. Striatal activity also correlates with action selection [Kimchi and Laubach, 2009], and with action-specific reward values [Roesch et al., 2009, Samejima et al., 2005].

In line with the architecture proposed in the model of Foster et al. [2000] (figure 4.1), there are hippocampal projections to the ventral and medial dorsal striatum [Groenewegen et al., 1987, Humphries and Prescott, 2010]. Studies combining watermaze testing with manipulations of ventral and medial dorsal striatum support the notion that these regions are required for spatial navigation. Lesions of the ventral striatum [Annett et al., 1989] and of the medial dorsal striatum [Devan and White, 1999] have been reported to impair spatial navigation on the incremental place learning task. In addition, crossed unilateral lesions disconnecting hippocampus and medial dorsal striatum also impair incremental place learning performance, suggesting hippocampo-striatal interactions are required [Devan and White, 1999]. To my knowledge, it has not been tested experimentally if there is a dichotomy between “actor” and “critic”. The experimental evidence outlined above is consistent with both actor and critic roles of the striatum [van Der Meer and Redish, 2011], but whether distinct or the same striatal neurons (or regions) act as actor and critic needs to be addressed. Perhaps, an architecture such as SARSA (State–Action–Reward–State–Action, Sutton and Barto [2018]), in which the values of a state-action pair are learned instead of the values of states only, could be considered, as it unites the actor and critic computation within the same network.

Phasic dopamine release from dopaminergic midbrain projections to the striatum has long been suggested to reflect reward prediction errors [Schultz et al., 1997, Glimcher, 2011], which are also reflected by the TD error in the model in figure 4.1, and dopamine release in the striatum shapes action selection [Gerfen and Surmeier, 2011, Morris et al., 2010, Humphries et al., 2012]. Direct optogenetic manipulation of striatal neurons expressing dopamine receptors modified decisions [Tai et al., 2012], consistent with the actor activity in actor-critic models of spatial navigation [Foster et al., 2000, Frémaux et al., 2013]. Moreover, 6-hydroxydopamine lesions to the striatum, depleting striatal dopamine (and, although to a lesser extent, also dopamine in other regions, including hippocampus) impaired spatial navigation on the incremental place learning task in the watermaze [Braun et al., 2012]. These findings suggest that aspects of the dopaminergic influence on striatal activity could be consistent with the modulation of action selection by the TD errors in an actor-critic architecture.

However, although there is long term potentiation (LTP)-like synaptic plasticity at hippocampo-ventral striatal connections, consistent with the plastic connections between the place cell network and the critic and actor in the model by Foster et al. [2000], a recent study failed to provide evidence that this plasticity depends on dopamine [LeGates et al., 2018]. The absence of dopamine modulation of hippocampo-striatal plasticity contrasts with the suggested modulation of connections between place cell representations and the critic and actor components by the TD error signal in the RL model. Thus, currently available evidence fails to support one key feature of the architecture described in section 4.2 [Foster et al., 2000].

4.3.2.2 The model does not capture the requirement of hippocampal plasticity

In the implementation of the model described above, plasticity takes place within the feed-forward connections from the place cell network modelling the hippocampus and the actor and critic networks that, as discussed above, could correspond to parts of the striatum. The model does not capture the finding that hippocampal NMDA receptor-dependent plasticity is required for incremental place learning in the watermaze if rats have not been pretrained on the task [Morris et al., 1986, 1989, Nakazawa et al., 2004].

In this section, I showed that actor-critic mechanisms shaped through TD learning are consistent with existing literature and could be supported by the striatum and the dopaminergic system, and future improvements of the model could consist in modifications of the particular action of the TD error, and in incorporation hippocampal plasticity mechanisms. The following section confronts the behaviour of the actor-critic agent to that of rodents in the watermaze.

4.3.3 The agent is less flexible than animals in adapting to changes in goal location

Although the agent is partly biologically realistic, the model does not capture the flexibility shown by rodents in the watermaze. Indeed, when the goal changes (on trial 20

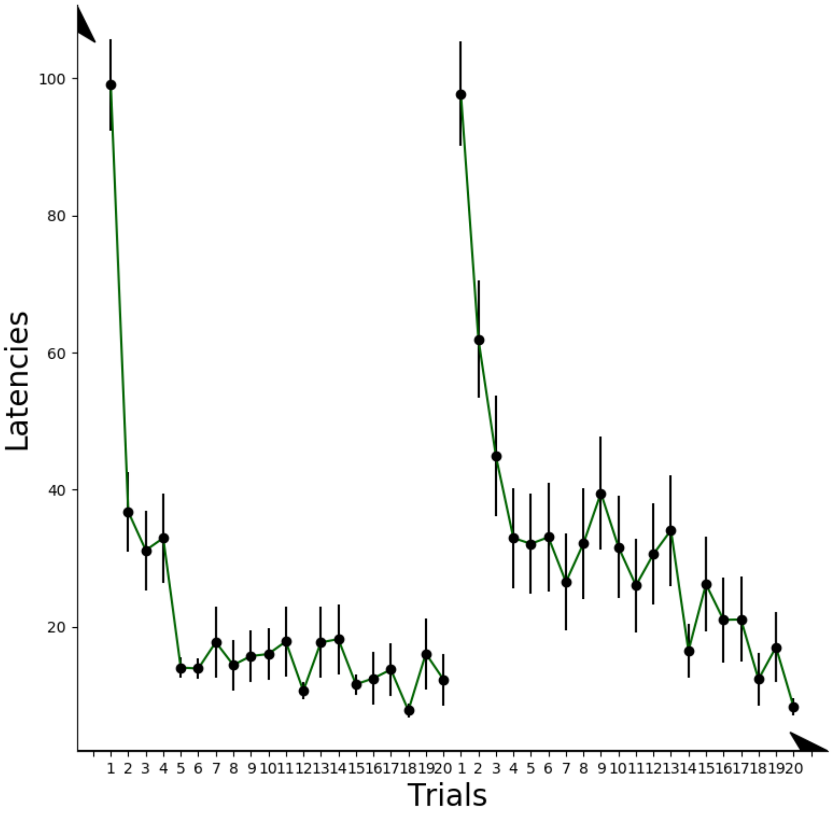


Figure 4.6: The agent is not flexible to changes in goal location. Latencies implemented using the model from Foster et al. [2000]. The time that the agent requires to get to the goal ("Latencies", vertical axis) reduces with trials (horizontal axis) and reaches almost a minimum (after trial 5). When the goal changes (on trial 20), the agent takes a very long time to adapt to this new goal location.

in figure 4.6), the agent takes many trials to adapt and takes more than 10 trials to reach asymptotic performance levels (also see figure 4a in Foster et al. [2000]). As the connection weights (4.8) and (4.10) are initialised at 0, the first trial is a random walk in the environment. Therefore, the end of the first trial, characterised by the reception of the reward, leads to a sudden improvement in policy and value function estimation, that can be seen in a drastic reduction between the escape latency of trial 1 and 2 of the first goal location. After trial 20, when the goal location is changed, the agent is not naive anymore. As the connection weights are now tuned to reach the previous goal location, on average it takes longer to adjust to the new goal location, which can be seen in the latency reduction between trial 1 and 2 is much smaller compared to the first goal location.

The high accuracy, but limited flexibility with overtraining, are well-known features of TD RL methods (*e.g.*, as discussed in Sutton and Barto [2018], Gershman [2017], Gershman et al. [2014a], Botvinick et al. [2019]). These *cached* methods have been proposed to account for the progressive development of habitual behaviours [Balleine, 2019]. TD learning is essentially an implementation of Thorndike's law of effect [Thorndike, 1927], which increases the probability of reproducing an action if it is positively rewarded. In the RL model discussed above (figure 4.1), a particular location, represented by activities of place cells with overlapping place fields, is associated with only one "preferred action", due to the unique weights that need to be fully relearned when the goal changes. Therefore, the way actions and states are linked only allows navigation to one specific goal location.

The model produces a general control mechanism, that, in this example, makes it possible to generate trajectories to a particular goal location. This control mechanism could be integrated within an architecture that allows more flexibility, for example, the goal location may be represented by different means than a unique value function computed via slow and incremental steps from visits to single goal locations. The next section considers how the RL model of figure 4.1 can be used, along with a uniform representation of both the agent and the goal location, to solve the watermaze DMP task.

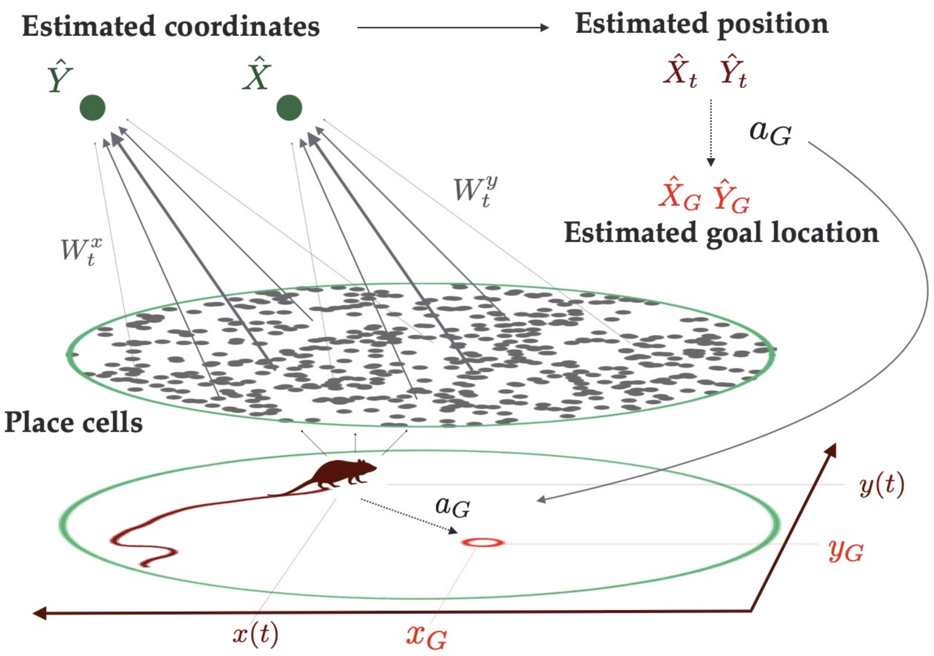


Figure 4.7: Architecture of the coordinate-based navigation system, which was added to the actor-critic system shown in figure 4.1 to reproduce accurate spatial navigation based on one-trial place learning, as observed in the watermaze DMP task [Foster et al., 2000]. Place cells are linked to coordinate estimators through plastic connections W_t^x, W_t^y. The estimated coordinates $\hat{X}, \hat{Y}$ are used to compare the estimated location of the goal $\hat{X}_{goal}, \hat{Y}_{goal}$ to the agent estimated location $\hat{X}_t, \hat{Y}_t$ to form a vector towards the goal, that is being followed when choosing the "coordinate action" a_G. The new action a_G is integrated into the actor network described in figure 4.1.

4.4 Coordinate-based navigation for solving the DMP task

The RL architecture discussed above (figure 4.1) cannot adjust to changes in goal location as observed in rodents and humans in the watermaze, but instead, there is substantial interference between successive goal locations (figure 4.6). To reproduce flexible spatial navigation based on one-trial place learning as observed on the DMP task, Foster et al. [2000] proposed to incorporate a coordinate system into their original actor-critic architecture (figure 4.7).

4.4.1 Learning coordinates for goal-directed displacement

The coordinate-based agent uses estimations of cartesian coordinates within the maze to form a goal-directed displacement towards the goal, supported by an additional action cell. The coordinate system (figure 4.7) is composed of two additional cells $\hat{X}$ and $\hat{Y}$ that learn to estimate the real coordinates x and y throughout the maze. These cells receive input from the place cell network through plastic connections W_t^x, W_t^y according to:

$$\hat{X}(p_t) = \sum_{i=1}^{N_P} W_{t,i}^x f_i(p_t), \qquad \hat{Y}(p_t) = \sum_{i=1}^{N_P} W_{t,i}^y(t) f_i(p_t). \tag{4.12}$$

The connections are updated pursuant to a TD error that represents the difference between the displacement estimated from the coordinate cells and the real displacement of the agent. The weights between the coordinate cells and the place cells are updated using self-motion information, according to

$$W_{t+1,i}^{x/y} = W_{t,i}^{x/y} + \Delta W_{t,i}^{x/y}, \tag{4.13}$$

with:

$$\begin{aligned} \Delta W_{t,i}^x &= \chi_{\text{Coord}}(\Delta\hat{X}_t + x(p_{t+1}) - x(p_t)) \sum_{k=1}^{t} \lambda^{t-k} f_i(p_k), \\ \Delta W_{t,i}^y &= \chi_{\text{Coord}}(\Delta\hat{Y}_t + y(p_{t+1}) - y(p_t)) \sum_{k=1}^{t} \lambda^{t-k} f_i(p_k), \end{aligned} \tag{4.14}$$

in which χ_{Coord} defines the learning rate for the coordinates, and $\Delta\hat{X}_t$ and $\Delta\hat{Y}_t$ represent the self-motion estimate in the x and y directions, *i.e.*, the difference between the estimated position at the previous location and at the current position, using the previous estimator. They are computed according to:

$$\Delta\hat{X}_t = \sum_{i=1}^{N_P} W_{t,i}^x \left(f_i(p_{t+1}) - f_i(p_t)\right), \qquad \Delta\hat{Y}_t = \sum_{i=1}^{N_P} W_{t,i}^y \left(f_i(p_{t+1}) - f_i(p_t)\right), \tag{4.15}$$

and $x(p_{t+1}) - x(p_t)$ and $y(p_{t+1}) - y(p_t)$ refers to the real displacement along the x and y direction. The term $\sum_{k=1}^{t} \lambda^{t-k} f_i(p_k)$ acts as an eligibility trace [Sutton and Barto, 2018], adding more importance to the most visited locations. The weights between place

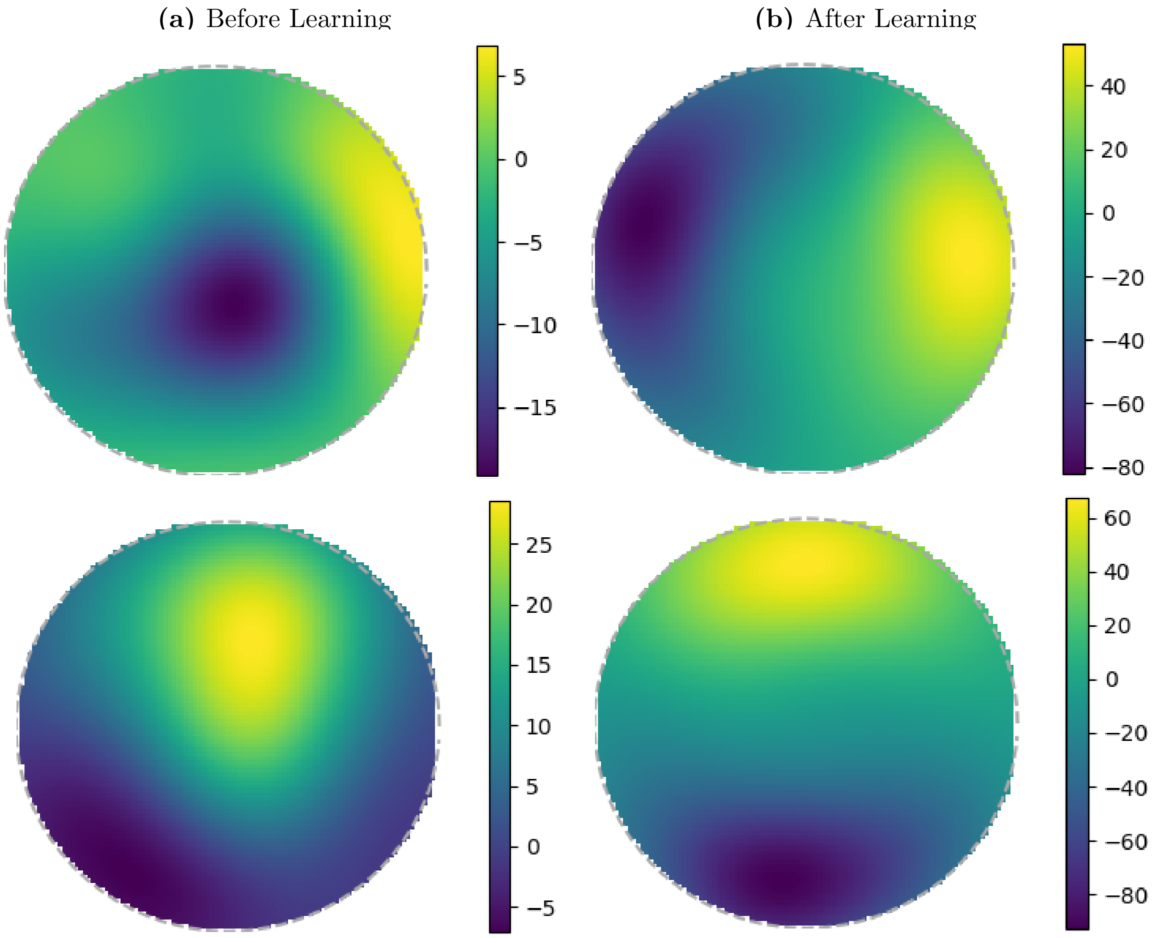


Figure 4.8: Estimated coordinates throughout the maze. Up: $\hat{X}$, Down: $\hat{Y}$. (a) very early on learning, the coordinates are not consistent throughout the maze. (b) after learning, the estimates follow a consistent gradient compared to the real coordinates, which enables correct computation of the motion toward the goal.

cells and the coordinate cells are reduced if the estimated displacement is higher than the actual displacement, and increased if it is lower, so that the estimated coordinates progressively become consistent with the real coordinates (figure 4.8).

The coordinates are used for goal-directed navigation. Every time the goal location is encountered, its estimated position $(\hat{X}_G, \hat{Y}_G)$ is stored (see figure 4.7). Thus, during subsequent trials, the agent has an estimate of its current location and of the location of the goal at every time step. Foster et al. [2000] added an additional action to the set of actions already available. Instead of defining movement in specific allocentric directions, as the other action cells do, going North, East, etc., that will be refered to as "allocentric direction cells", the coordinate action cell a_G points the agent in the direction of the estimated goal location. The estimation of x and y are used to compare the agent's estimated position $\hat{X}_t, \hat{Y}_t$ to the estimated goal location $\hat{X}_G, \hat{Y}_G$ (which is stored after the first trial of everyday) in order to form a vector leading to the estimated goal location (figure 4.7). The selection of the coordinate action induces the following change in position:

$$p_{t+1} = p_t + \alpha \frac{\hat{X}_G - \hat{X}_t}{\left\| \hat{X}_G - \hat{X}_t \right\|}. \tag{4.16}$$

A further distinguishing feature of this action cell is in the update of weights linked to this cell. The learning rule does not depend on position; instead, the weight between the coordinate action cell a_G and the i place cell is computed according to:

$$Z_{t+1}^{Gi} = Z_t^{Gi} + \Delta Z_t^{Gi}, \tag{4.17}$$

with:

$$\Delta Z_t^{Gi} = \chi_G \delta_t g_G(t), \text{ for } i = 1, \ldots, N_P, \tag{4.18}$$

where $g_G(t) = 1$ if the coordinate action cell a_G has been selected and zero otherwise, and δ_t refers to the error computed from equation (4.6). When there is no goal coordinate in memory, the direction is chosen at random among the N_A other directions and the coordinate action cell weights are not updated. This update rule is identical as the one used in Foster et al. [2000]. The update rule is heuristic, and does not depend on the place cell activities: as the coordinate action defines a motion from the agent's location to the goal's location, its reinforcement should not depend on the position of the agent. All of the other weights Z^{ji}, $j = 1, \ldots, N_A$, $i = 1, \ldots, N_P$, evolve according to equation (4.10). All parameters and implementation details are provided in Appendix A.

4.4.2 Coordinate-based navigation enables flexibility to changes in goal location

The agent very quickly adapts to new goal locations, reproducing performance similar to rats and humans on the watermaze (figure 2.13a) and virtual [Buckley and Bast, 2018]

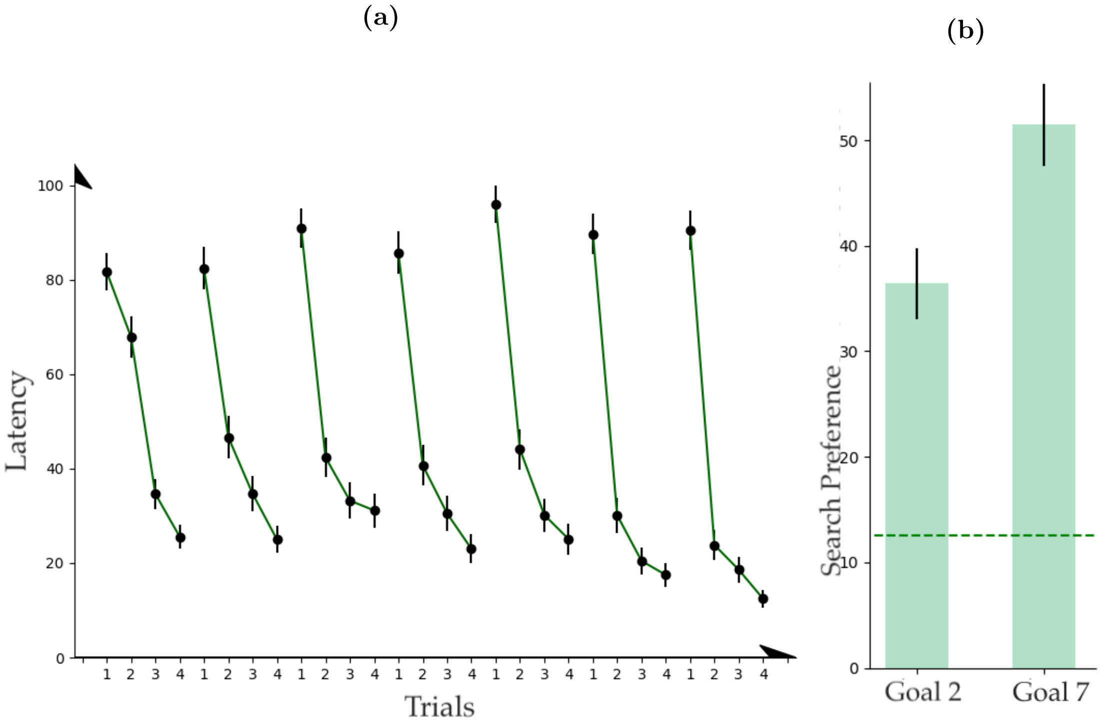


Figure 4.9: Performance of the extended model using coordinate-based navigation. (a) Escape latencies of the agent when the goal location is changed every four trials, mimicking the watermaze DMP task. (b) Search preference for the area surrounding the goal location, as reflected by the percentage of time the agent spends in an area centered around the goal location when the second trial to a new goal location is run as a probe trial, with the goal removed (stippled line indicates the percentage of time spent in the correct zone by chance, *i.e.*, if the agent had no preference for any particular area), computed for the second and the seventh goal locations. One-trial learning of the new goal location is reflected by the marked latency reduction from trial 1 to trial 2 to a new goal location (without interference between successive goal locations), and by the marked search preference for the new goal location when trial 2 is run as a probe. The data in (a) were obtained by computing the model in Foster et al. [2000], and the data in (b) by adapting the model to reproduce search preference measures when trial 2 was run as a probe trial. The increase in search preference observed between the second and seventh goal location is addressed in section 4.4.4.

DMP task, respectively. Using the model of Foster et al. [2000], I can replicate their finding that the model reproduces key aspects of the pattern of latencies shown by rats and human on the DMP task, *i.e.*, a marked reduction from trial 1 to 2 to a new goal location and no interference between successive goal location (figure 4.9a). Figure 4.9a shows a gradual improvement in the performance, as characterised by an increasing difference between the trial 1 and 2 of the same goal location across different goal locations. This reflects the progressive refinement of the coordinates, which gradually become consistent over the maze (figure 4.8). As a consequence, the selection of the coordinate action becomes more successful at leading to the goal. Therefore, the agent progressively learns to use the coordinate action, which leads to the goal, as the estimated coordinates are consistent. On the last goal location, the agent almost reaches optimal performances from the second trial.

I additionally computed the search preference (described in section 2.5) for an area centred around the goal location, showing that the agent also persists around the goal location when the reward is not provided, reproducing the behaviour shown by rodents in the task [Bast et al., 2009]. To do so, the reward is not provided to the agent on trial 2 of the second and seventh goal location, which would be the equivalent to removing the platform on the second and seventh day of the experiment in the watermaze DMP task (described in section 2.5.3), and the time that the agent spends navigating around an area centred around the goal location is compared to the total time spent navigating in 8 areas of equal surface covering the whole maze (method similar to the one used in experiments, detailed in section 2.5.2). Figure 4.9b shows that the coordinate model reproduces the markedly above-chance search preference for the vicinity of the goal location also demonstrated by rodent in the watermaze when trial 2 to a new goal location is run as probe trial where the platform is removed (figure 2.13b). Moreover, the search preference increases between goal 2 and goal 7. This is the result of the strength of the coordinate action cell which is becoming more reliable as the coordinates become more consistent.

The marked search preference for an area around the goal location indicates that additionally to adjusting to changes in goal location, the agent is also persistent around the goal location when the goal is removed. This successfully mimics the marked search preference around the goal location by rodents in the watermaze DMP task, suggesting that coordinate-based navigation could support aspects of spatial learning in the watermaze DMP task.

In this section, I have shown that the performance of the agent is consistent with the performance of rodents in the watermaze DMP task (see figure 2.13), by replicating the escape latencies from Foster et al. [2000], and by extending the approach to measure the search preference around the goal location. Moreover, the approach is partly biologically realistic, a point that the next section discusses.

4.4.3 The implementation of goal-directed displacement is partly biologically realistic

In this section, I discuss the biological implementation of the coordinate actor-critic model presented in subsection 4.4.1. I show that the global architecture is consistent with actor-critic mechanisms and with experimental results showing goal and goal-directed neuronal correlates. I further argue that goal memory could be incorporated using hippocampal plasticity mechanisms.

4.4.3.1 The model's actor-critic component and striatal contributions to DMP performance

Given the association of actor-critic mechanisms with the striatum [Joel et al., 2002, Khamassi and Humphries, 2012, O'Doherty et al., 2004, van Der Meer and Redish, 2011], the actor-critic component in the model is consistent with recent findings that the striatum is associated with rapid place learning performance on the DMP task. More specifically, using functional inhibition of the ventral striatum in rats, Seaton [2019] showed that the ventral striatum is required for one-trial place learning performance on the watermaze DMP task; moreover, using high-density electroencephalogram (EEG) recordings with source reconstruction in human participants, Bauer et al. [2020] found that theta oscillations in a circuit including both temporal lobe and striatum are associated with one-trial place learning performance on the virtual DMP task.

The model suggests that, after a few trials, once the action probability for the coordinate action has reached one, the movement is predefined by following a vector pointing to the goal location. The critic becomes inconsistent, as the action now does not follow the value gradient anymore, therefore there is no control over the behaviour from the TD error. The continued association of the striatum with DMP performance, beyond the first few trials, is consistent with the role of the striatum as the "actor" [van Der Meer and Redish, 2011], and the model would suggest that the striatum computes a goal-directed vector using estimated locations.

In the following section, I discuss that the estimated goal location is consistent with recent findings of goal and goal-directed encoding in the hippocampus and striatum.

4.4.3.2 Neural substrates of the goal representation

The storage of a goal representation is consistent with recent findings of goal and goal-directed correlates in the brain. The findings of goal-vector cells in the bat hippocampus [Sarel et al., 2017] and of "predictive reward place cells" in mouse hippocampus [Gauthier and Tank, 2018] support the idea implemented in the model that consistency of representations, *i.e.*, unified representations of goals and locations across tasks and environment, could help goal-directed navigation. In particular, egocentric boundary encoding neurons have been found in the striatum of rats, although in the dorso-medial striatum [Hinman et al., 2019]. As rats navigate in the watermaze using surrounding cues, these cells could inform striatal navigation in the DMP task [Bicanski and Burgess, 2020].

Experimental results suggest that the goal information should be stored within hippocampal representation. In the extension to the classical TD architecture [Foster et al., 2000], the encounter with a new goal location does not involve a change in place cell representation, and the formation of the memory of the new goal location is not addressed. Experimental evidence suggests that a goal representation could lie within hippocampal representations themselves [Hok et al., 2007, Poucet and Hok, 2017, McKenzie et al., 2013, Gauthier and Tank, 2018]. McKenzie et al. [2013] studied hippocampal CA1 representations during learning of new goal locations in an environment where previous places were already associated with goals. They showed that neurons coding for existing goals would also encode new goal locations, and that these representations progressively separate with repetitive learning of the new goal location, but maintain an overlap of representations between all goal locations. Moreover, Hok et al. [2007] observed an increase in firing rate around goal locations outside of place cells' main firing field, and Dupret et al. [2010] showed that learning of new goal locations by rats in a food-reinforced dry-land DMP task is associated with an increase in the number of CA1 neurons that have a place field around the goal location. Furthermore, Dupret et al. [2010] showed that both this accumulation of place fields around the goal location and rapid learning of new goal locations is disrupted by systemic NMDA receptor blockade. These findings suggest that goal representation can be embedded within the hippocampus, that new goal locations are represented within similar networks as previous goal locations, and that the hippocampal remapping emerging from new goal locations is linked to behavioural performance and may depend on NMDA-receptor mediated synaptic plasticity.

In line with this suggestion, studies, combining intra-hippocampal infusion of an NMDA receptor antagonist with behavioural testing and electrophysiological measurements of hippocampal LTP, showed that hippocampal NMDA-receptor dependent LTP-like synaptic plasticity is required during trial 1 for rats to learn a new goal location in the watermaze DMP task [Steele and Morris, 1999] and in a dry-land DMP task [Bast et al., 2005]. LTP-like synaptic plasticity may give rise to changes in place cell representations [Dragoi et al., 2003], which could contribute to changes in hippocampal place cell networks associated with the learning of new goal locations [Dupret et al., 2010].

Map-like representations of locations, integrated within an RL architecture, may be part of neural mechanisms that enable flexibility to changing goal locations in the watermaze DMP task. However, cartesian coordinates are convenient here because the task is implemented within an open-field arena, but do not seem to provide a biologically realistic implementation of spatial navigation problems in general. For example, they do not allow navigation in an environment with walls, for which geodesic coordinates would be more appropriate [Gustafson and Daw, 2011]. Moreover, the approach described here does not address how the goal representation comes about, and the model does not specify how the policy adjusts when the agent does not encounter the predicted goal. Chapter 6 describes how a hierarchical architecture can provide a solution to this problem. Additionally, the improvement in performance highlighted in section 4.4.2 across trials, both in terms of latencies and search preference, is not consistent with the performance of rodents and humans in the lab, a point that the next section addresses.

4.4.4 Limitations of the model in reproducing DMP behaviour in rats and humans

The coordinate approach relies on computational "tricks" that are required to make the approach work, but for which plausible neurobiological substrates remain to be identified. Early in training, the movement of the agent is based on the activity of the "allocentric direction cells", which are used to lead the exploration of the environment. This exploratory phase allows learning of the coordinates. As the estimated coordinates $\hat{X}, \hat{Y}$ (4.15) become more consistent with the real coordinates, the coordinate action a_G becomes more reliable, as it will always lead the agent in the direction of the goal. During the first trial to a new goal location, the coordinate action cell encodes random displacement, until the goal is found and its estimated location is stored. During this trial, the coordinate action is not reinforced, a trick that prevents its devaluation. On the subsequent trials, the coordinate action encodes the displacement towards the stored estimated goal location (as described before) and is reinforced. Therefore, the probability of choosing the coordinate action gradually becomes one, and it comes to be the only action followed.

One consequence of this is that, unlike in rats and people, the agent's performance both in terms of latency reduction and in terms of search preference gradually improves

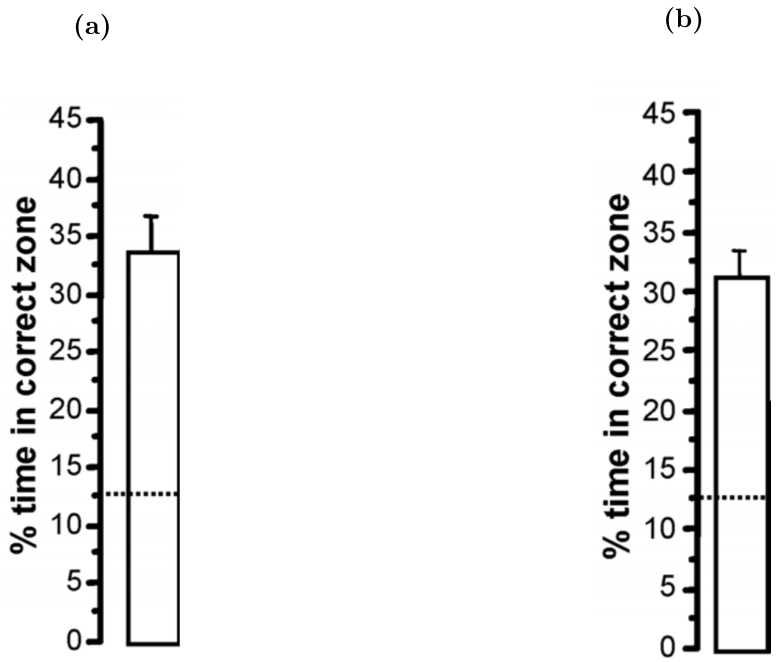


Figure 4.10: Comparison of search preference with exposure to the task. Between the a) first day, and b) the seventh day, rats show similar search preference, as opposed to the model as shown in figure 2.13b. Figures adapted from Bast et al. [2009]. .

across successive new goal locations (see figure 4.9a and 4.9b). The gradual improvement of latency reductions and search preferences does not fit with behaviour shown by rats (Compare figures 2.13b and 4.10) and human participants [Buckley and Bast, 2018]. Moreover, the random search during trial 1 is inconsistent with the finding that rats on the DMP task (but not human participants, [Buckley and Bast, 2018]) tend to go towards the previous goal location on trial 1 to a new goal location (Steele and Morris [1999], Pearce et al. [1998], and the own unpublished observations of Dr. Bast's team); in addition, both rats and human participants show systematic search patterns on trial 1 [Buckley and Bast, 2018, Gehring et al., 2015]. The adjustment of the policy when the predicted goal is not encountered is not addressed in the current approach, a point that chapter 6 addresses.

4.5 Conclusions

In this chapter, I presented an actor-critic architecture, originally published in Foster et al. [2000], which leads to action selection based on the difference of estimated rewards to be received. The model [Foster et al., 2000] uses an estimate of the value function over the maze to drive behaviour through a critic network that receives place cell activities as input. It can successfully learn to select which action is best through an actor network, which also receives place cell input and is trained to follow the gradient of the value function from the difference in successive critic activities, also termed TD error. The actor-critic agent can follow trajectories towards a particular, fixed, goal location, that corresponds to the maximum of the value function. However, when the goal location changes, the model needs many trials to adjust in order to accurately navigate to the new goal, which is in marked contrast with DMP performance of rats and humans [Steele and Morris, 1999, Bast et al., 2009, Buckley and Bast, 2018].

To account for one-trial place learning performance on the DMP task, I presented a possible extension to the actor-critic approach using coordinate-based navigation [Foster et al., 2000]. The agent first learns the cartesian coordinates of locations throughout the maze that facilitates the direct comparison between the goal location and the agent's location. This enables the computation of a goal-directed displacement towards any new goal location throughout the maze and the agent's behaviour reproduces flexibility shown by humans and animals towards new goal location. I have extended the approach by Foster et al. [2000] to measure the search preference around the goal location of the agent, showing that the agent is persistent around the missing goal location.

Given that the striatum has been associated with actor-critic mechanisms [Joel et al., 2002, Khamassi and Humphries, 2012, O'Doherty et al., 2004, van Der Meer and Redish, 2011], using an actor-critic agent for flexible spatial navigation is consistent with empirical evidence associating striatal regions with place learning performance on both incremental [Annett et al., 1989, Devan and White, 1999, Braun et al., 2010] and DMP [Seaton, 2019, Bauer et al., 2020] tasks. However, contrasting with the coordinate extension to the actor-critic architecture, experimental evidence suggests that goal location memory may lie within hippocampal place cell representations [McKenzie et al., 2013, Dupret et al., 2010] and that one-trial place learning performance on DMP tasks in rats requires NMDA receptor dependent LTP-like hippocampal synaptic plasticity [Steele and Morris, 1999, Bast et al., 2005]. Moreover, the model does not address how the goal location is remembered and selected, a limitation that chapter 6 addresses.

Findings with the model examined in this chapter support the important idea that, embedded within a model-free RL framework, a map-like representation of locations within an environment may allow computations by the agent to produce efficient navigation to new goal locations within as little as one trial. This idea is also present in recently proposed agents that are capable of flexible spatial navigation based on an RL system complemented by path integration mechanisms to derive a grid-like map of the environment (which resembles entorhinal grid cell representations) that can be used to compute trajectories from the agent's location to the goal [Banino et al., 2018] and has also lead the watermaze DMP task to be solved using a graph search algorithm in Dollé et al. [2018]. Future models should consider using geodesic rather than cartesian coordinates, to be adaptable to other environments.

To conclude, elements of an actor-critic architecture may account for some important aspects of rapid place learning performance in the DMP watermaze task. Together with a map-like representation of location, an actor-critic architecture can support the efficient, fast, goal-directed computations required for such performance. The next chapter provides an improvement in the realism of the implementation of the network: it uses a similar architecture but in continuous time and action space.

Chapter 5

An actor-critic architecture in continuous space, actions and time for navigation in the Morris watermaze

5.1 Introduction

In this chapter, I present a novel version of the actor-critic architecture examined in chapter 4 which considers continuous representations of actions and time. I present a mathematical framework consistent with continuous representations, and discuss the necessity of precision of the representation, by comparing spiking, discrete and continuous RL approaches, to perform certain tasks, and in particular in the case of watermaze navigation. This chapter contains a review of the literature about spike coding in biological and artificial networks, and the implementation of a novel RL model in continuous time, action and space using rate networks, inspired from Frémaux et al. [2013]. The model adapts the approach proposed by Frémaux et al. [2013] using spiking units for spatial navigation to an approach using rates for spatial navigation in the Morris Watermaze. I will describe how to adapt a spiking approach while preserving the classical activity patterns observed in biological neurons, and implement the adapted approach for navigation in the Morris watermaze.

Most RL models use discrete representations of the relevant variables that constrain the precision of the behaviour to the chosen coarseness of the representation, and that cannot efficiently comply with the requirements of natural situations. In chapter 4, the state, the position of the agent, was encoded within a network of units mimicking place cells, of which activity was defined by a Gaussian function around their preferred location (equation (4.1)). This is referred to as a continuous state representation, as opposed to grid-world agents, in which the possible states of the agent are organised within a grid covering the whole environment. In section 4.3.2, I have reddiscussed that the use of place cells as a basis for state representation within an actor-critic architecture is partly consistent with experimental evidence. From these discussions and observations, one can assume that the neural representations of locations underlying spatial navigation are continuous. Although certain neurons can be found that seem to be tuned to one particular discrete entity, for example, that respond only to the presentation of a particular face [Quian Quiroga and Kreiman, 2010], it is hard to conceive that a discrete encoding of locations only could be the basis of flexible navigation capabilities in many diverse and changing situations. Without any efficient representation, the state space would very quickly scale up with the number of locations to account for. In chapter 4, the overlapping Gaussian activity profiles provide a basis of space in which all locations have their own representation, and permit generalisation of experience across the state space, as discussed in section 4.3.1. In particular, both value and action-outcome information are generalised across space: if an action is beneficial at a certain location, it is certainly also beneficial at a neighbouring location; and if a certain location has a high value, a neighbouring location certainly should have a high value too.

Time, actions, and rewards can also be considered as continuous variables. Continuous representation of actions has the advantage of enabling smoother control and generalisation. The former is because as the representation is not restricted anymore to a coarse description pre-defined by the choice of how to partition the variable in question, more nuances are possible. The latter directly follows the discussion over the use of place cells as a basis for state representation.

Similar to a physical space, a continuous time variable seems intuitively more biologically realistic, as it is hard to think that all biological phenomena could be subject to the constraints of the choice of a particular time-step. Although one could argue that, computationally, one can always choose the time step of a discrete time representation to be as small as necessary to encode the precision required, performing biologically realistic computations should require the choice of a consistent mathematical framework. As explained in section 3.2.2.2, the RL framework can be adapted to a continuous time representation. Theoretically, a continuous time framework can account for as much temporal precision as required; however, perhaps, the most temporally precise method of information transmission in the brain can be considered to be the 'spike', which is often approximated as a Dirac-delta function, a temporally very localised impulse. Many computational neuroscientists therefore investigate how to perform diverse tasks using spiking networks, but evidence of networks using spiking representation being better at certain RL tasks than less temporally precise networks is still lacking.

In this chapter, I discuss the continuity of time and actions and its link with temporal precision. In section 5.2.1, I discuss results from Doya [2000] which compared two similar actor-critic architectures, one based on discrete representations, and the other based on continuous representations, in an exemplar RL task. In section 5.2.2, I discuss results from recordings suggesting that spike coding can be necessary or useful when temporal precision is critical, and discuss results from computational models suggesting that a spike train enables many forms of information code. I then compare a spiking [Frémaux et al., 2013] and a rate [Doya, 2000] network on the same task and thereby indicate that spiking networks do not outperform rate networks in model-free RL tasks. Finally, I implement a novel continuous actor-critic rate network of my own devising in a virtual watermaze task and compare its performance to that obtained on a spiking network actor-critic architecture proposed by Frémaux et al. [2013] on a similar spatial navigation task. I show that a continuous action representation enables a smooth control of the direction selection which leads to more biologically realistic trajectories. I then discuss the necessity of spiking precision in spatial navigation.

5.2 Discrete, continuous, and spiking representations: advantages and necessity

In this section, I give examples of the advantage of the use of a continuous versus discrete representation. I demonstrate that it enables finer precision, by comparing the performance of a continuous to that of a discrete representation on a similar task [Doya, 2000]. As a fine precision seems to be beneficial to the performance in certain tasks, I then discuss spiking representations and their advantages, and I compare the performance of a spiking agent to that of a rate agent on a similar RL control task and show that both agents show comparable performance in this model-free RL task.

5.2.1 Continuous representations for smoother control and generalisation

A discrete action representation reduces the opportunities for generalisation of experience. In section 4.2, the action space representation was discrete, with $n = 8$ action cells representing different directions of the space [Foster et al., 2000]. The action selection was discrete too, as only one direction determined by one action cell was expressed at every time step, based on a probability distribution given by equation (4.4). After displacement, only the connection weight of the chosen action was updated (equation (4.10)), such that a neighbouring action cell, coding for a similar direction, was not reinforced from the outcome observed from that decision. Therefore, there was no generalisation across the action space. In this section, I discuss the benefit of allowing generalisation through a continuous action space, by comparing a continuous actor-critic network proposed by Doya [2000] to a discrete actor network.

In RL tasks that require fine control, a continuous representation enables better performance. Doya [2000] was the first to introduce an actor-critic architecture in continuous time, action, and space. The continuity of time implies a different definition for the value function and the use of a different TD error (as expressed in equations (3.9) and (3.15) in section 3.2.2.2). Doya [2000] first compared a discrete actor-critic method to their proposed continuous actor-critic in a task extensively used in reinforcement learning: the pendulum swing-up task (see figure 5.1a). In this task, the agent has to exert a torque at the joint of a pendulum to lift its outermost tip upright. In the discrete version of the actor-critic architecture, the time is a discrete variable, as presented in section 3.2.2.1, and the action space is also discretised: the torque can take two possible values $F = \pm F_{\text{max}}$. In the continuous actor-critic, time is a continuous variable, as presented in section 3.2.2.2, and the action space is also continuous, the torque can take any value such that $F_{\text{max}} \geq |F|$. Note that the continuous action representation is similar to the actor-critic architecture presented in chapter 4; however, in Doya [2000], there is only one action cell, $n = 1$, as the only decision concerns the value of the torque. In addition to a continuous action representation, the learning rule used to update the weight between the states and the action cell depends on the value of the action, so that similar torque values to the one chosen during a decision are also informed depending on the outcome of the decision [Doya, 2000]. Doya [2000] compared the number of trials that each agent requires to reach 10 successful attempts, in which success is defined by achieving $|\theta| < \pi/4$, where θ refers to the angle of the pendulum, for more than 10 seconds. Figure 5.1b shows that the continuous actor needs five times fewer trials to reach 10 successful trials than the discrete continuous actor. The example of the pendulum illustrates that a continuous representation of actions can lead to better performance. In that case, it is both because the agent has access to finer control and can generalise the outcome of a decision to similar torque values.

An arbitrarily fine control mechanism is biologically realistic and adapted to the variety of natural tasks that brains perform. Many natural control tasks, such as driving a car or picking up a glass, require smooth and continuous action selection to be performed well. Therefore, using a continuous representation seems a more biologically realistic approach than using an arbitrarily coarse representation. Similarly, continuous time

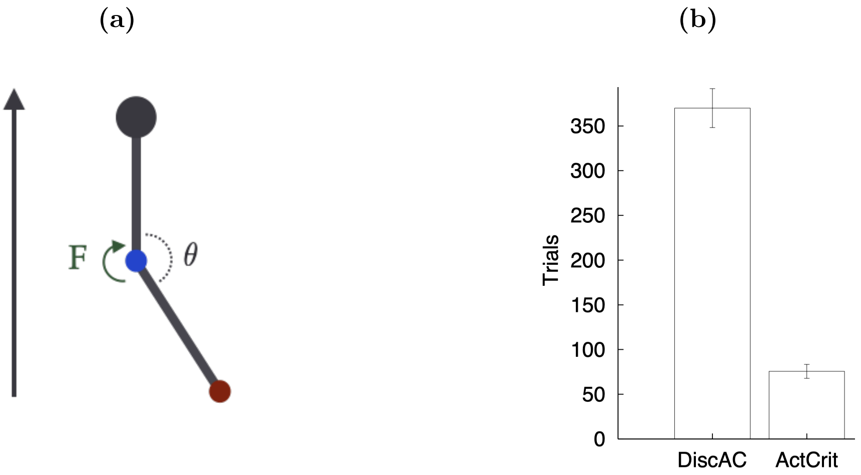


Figure 5.1: Comparison of a continuous and discrete actor on the pendulum swim task. (a) In the pendulum task used by Doya [2000], the agent has to exert a torque at the joint (blue dot) and is given a reward that depends on the height of the pendulum (red dot), $R(\theta) = \cos(\theta)$. The pendulum is subject to its own weight. The action space corresponds to possible values of the torque. (b) performance of the discrete (DiscAC) and continuous actor-critic (ActCrit) agent in this task, shown as the number of trials before the agent achieves 10 successful trials. In the discrete actor-critic, the action space is discrete, with $F = \pm F_{\max}$. In the continuous actor-critic, the action space is continuous, such that $F_{\max} \geq |F|$. A trial is considered successful when the time that the pendulum stays up ($|\theta| < \pi/4$) is more than 10 seconds. The bar plot shows the number of trials necessary before 10 successful trials. The discrete model is five times slower to learn than the continuous model. Figure adapted from Doya [2000].

enables a finer control. In this section, a discrete actor-critic is compared to a continuous actor-critic. It is hard to say whether the continuity of time or the generalisation from the continuity of action representation and update enable faster learning. The two are intuitively related, as a finer time resolution can only lead to a subtle control if the representation of the action is also suitably resolved. The finest temporal resolution of information transmission can be approximated to the emission of a spike (section 2.2). In this thesis, the models presented so far (the discrete actor-critic in section 4.2 and the goal-directed navigation model in section 4.4) use rate coding for information encoding and transmission. From a neural point of view, rate coding requires the integration of spikes over a certain time window (see section 2.2), thus theoretically reducing the temporal precision achievable. The next section describes the possible coding scheme from a spike train and discusses the potential computational advantages of a spike-based scheme.

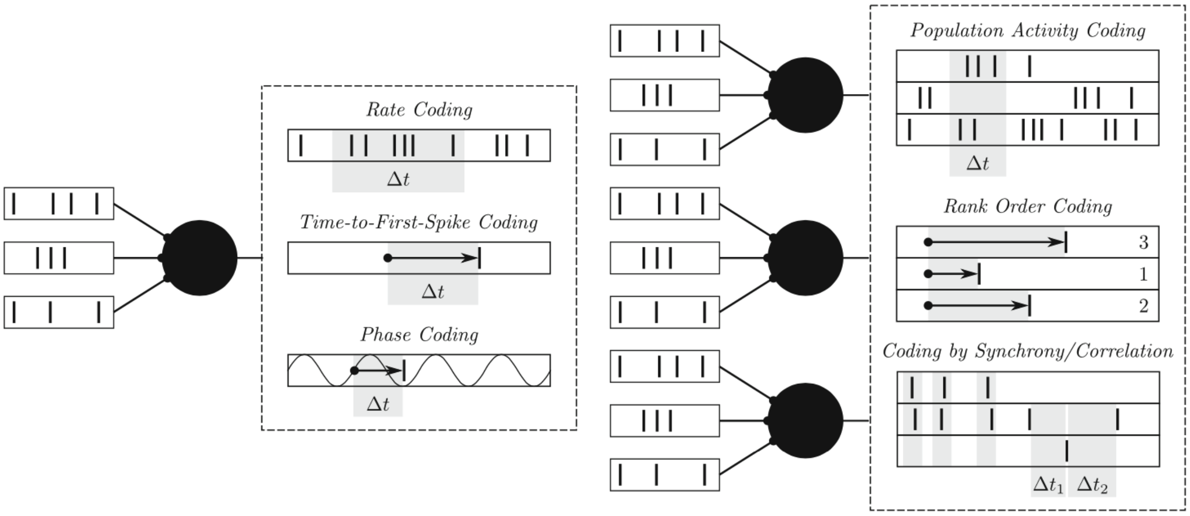

Rate Coding
Δt
Time-to-First-Spike Coding
Δt
Phase Coding
Δt
Population Activity Coding
Δt
Rank Order Coding
3
1
2
Coding by Synchrony/Correlation
Δt_1 Δt_2

Figure 5.2: Different encoding possibilities from a spike train. Three kinds of single-neuron codes are represented: In rate coding, the information is transmitted through the frequency of spikes occurring over a certain time window. The time-to-first spike coding transmits information in the time spent between the presentation of a stimulus or a particular event and the first spikes emitted by the neuron of interest. In phase coding, the information is contained in the timing of a spike compared to a background oscillatory activity. Similar codes can be defined for a population of neurons. Population activity coding is the equivalent of rates but for a population of neurons, in which the activity of the population can be for example defined as the mean of the rates of each neuron. In rank order coding, the information is transmitted depending on the exact order of the spikes from the population. Lastly, synchrony transmits information using populations that emit spikes at the same time. Figure adapted from Walter et al. [2016]

5.2.2 Spikes are precise, but hard to integrate within an RL framework

Spike coding enables different information transmission possibilities at different timescales. Rate coding is a simplification of what appears to be a very complex code. If one removes a layer of simplification, spikes become the unit of transmission in the brain. Looking at spikes is already an approximation, as spikes rely on very complex biological processes, and other more localised mechanisms seem to be involved in transmitting information, for example dendritic spikes [Smith et al., 2013], which are spikes (of smaller amplitude compared to somatic spikes) observed at the level of a dendrite. For a single neuron code, a rate code seems to be the easiest approximation to transmit information. However, the cost of this approximation is an ongoing debate. In particular, it is unknown under what type of assumption this reduction is possible, and whether these are biologically realistic and empirically supported (*i.e.* are biological tissues using rates as a basis for computations, or do rates only emerge from our interpretations of neural recordings) [Brette, 2015]. Trying to grasp what code is used by neurons to communicate, and the potential role of spikes in it, are ongoing questions in neuroscience [Yu et al., 2018, Gerstner, 1998]. Many coding possibilities can be designed from a spike train [Walter et al., 2016]. Figure 5.2 illustrates different options.

Time-to-first spike coding relies on the precise timing of a single spike and enables fast information transmission. All the information is carried by the time between the stimulus onset and the occurrence of a single spike. Neural correlates of time-to-first spike coding have been found in sensory neurons. One famous example of them is H1 neuron in the fly brain [Bialek et al., 1991]. Flies perform visual-guided flights in which they adjust their trajectories on very short timescales (30 ms) [Bialek et al., 1991]. In the visual system of the fly, horizontal motion detection is performed by a very small population of single wide-field, movement-sensitive neurons (called H1) which have a maximum firing rate of 100 to 200 Hz. Therefore, the behavioural decision has to be made dependent on only a few spikes of a handful of cells, making a rate code impossible [Bialek et al., 1991]. The information transmitted by these neurons seems to rely on precise temporal spike coding, and these neurons have been extensively studied to investigate the spiking code [Van Hateren et al., 2005, Haag and Borst, 1997, Warzecha et al., 1998, Strong et al., 1998]. Computationally, rank order coding can be done by making the post-synaptic neuron activity maximal only when its different inputs are activated in the order of their synaptic weights [Thorpe et al., 2001]. Johansson and Birznieks [2004] have found neural correlates of time to first spike coding in humans. They recorded from different afferent pathways (afferent pathways or neurons are pathways/neurons that carry information from peripheral sensory neurons towards the central nervous system) in the median nerve of the upper arm, which provides innervation to the thumb and muscles of the wrist and forearm while exerting various small forces on the fingertips of participants. They observed that the relative timing of the first spike of individual neurons within those pathways carries information about the direction of the force exerted and the surface of contact with the fingertip.

Time-to-first spike coding within a population can efficiently transmit independent information about a stimulus in parallel. While recording neuronal responses to different forces applied on the fingertip, Johansson and Birznieks [2004] additionally showed that the sequence coding within the population of neurons, in relation to events on the fingertip, provides faster propagation of information than rate codes. The authors showed that the *rank* of activation of the various neurons within one exposure to a stimulus could lead to correct discrimination of the stimulus on average 10 to 20 milliseconds faster than rate code [Johansson and Birznieks, 2004]. Rank order coding, in which the information is transmitted within the sequence of the first spike of the neurons involved in the population (see figure 5.2), is a fast solution for population-based coding. A high speed of transmission seems required for natural object manipulation, as the complexity of the stimuli involved (force, surface of contact, speed of movement) requires parallel processing to be transmitted efficiently. The results from Johansson and Birznieks [2004] suggest that fast parallel encoding can be achieved by encoding different dimensions of information within the first spike of different individual neurons, and that across the whole population a stimulus can be efficiently conveyed using a rank order code [VanRullen et al., 2005].

Information can also be extracted from a spike train when looking at the phase of spikes of a set of neurons relative to a periodic background oscillation informative of a population code, which is known as phase coding [Walter et al., 2016], and seems to enhance precision. A very famous example of phase coding is theta phase precession observed in hippocampal place cells [O'Keefe and Recce, 1993]. In an electroencephalograph (EEG), which measures electrical activities using electrodes outside the skull, one can see theta waves generated from the hippocampus. Theta waves are electrical waves of frequency between 7 and 12 Hz. Theta phase precession is the observation that hippocampal neurons exhibit a particular relationship to the theta wave, first observed during movement through place fields [O'Keefe and Recce, 1993]. More precisely, bursts of spikes have been observed consistently at a particular phase of the theta wave at the start of motion through the field. Moreover, this particular phase at which neurons fire moves forward within the theta phase at each theta cycle during the traversal of the field [O'Keefe and

Recce, 1993]. The phase is highly correlated with spatial position. This phenomenon is hypothesised to enable more spatial precision of the place cell code [O'Keefe and Recce, 1993], and to underlie temporal and spatial relationship of hippocampal memory traces [Skaggs et al., 1996].

Lastly, population synchrony, in which information is transmitted depending on which neurons of a population are synchronously firing (see figure 5.2), enables rapid classification of stimuli. Synchrony has been extensively studied in neurons in the visual cortex, in which neurons can synchronise their activity within a few milliseconds if they encode the same stimulus [Maldonado et al., 2008], and whose synchronisation is shaped through learning with exposure to different repeated stimuli [Ghisovan et al., 2008]. Synchronisation between two neurons can occur if they represent a similar stimulus and can disappear if they belong to different populations that represent different stimuli [Gerstner and Abbott, 1997]. Temporally precise population synchrony has also been found to carry stimulus-specific information that is not reducible to a rate code or to individual spike train within the population in auditory processing pathways in songbirds [Theilman et al., 2021].

All potential coding solutions using precise spike timing seem to be required when high temporal accuracy is necessary [Walter et al., 2016, Cessac et al., 2010]. Spike-based learning can pick up correlations within the inputs on timescales of milliseconds [Kempter et al., 1999]. In tasks in which the processing speed is required to be high, for example in visual, auditory or tactile discrimination tasks [VanRullen et al., 2005], the timing of single spikes can be precise and reliable [Bohte, 2004]. Evidence also suggests that precise spike synchronisation helps precise motor coordination [Riehle et al., 1997], but this precision comes at a computational cost, and such precision might not be necessary for every task [Cessac et al., 2010].

Evidence suggests that while sensory areas exhibit fast dynamics and therefore might critically depend on spike-timing precision, areas that are further along in the stream of information, such as frontal areas, exhibit slower dynamics, with a hierarchy of timescale processing that depends on the position of the area in the processing stream [Rossi-Pool et al., 2021]. Indeed, the integration window of neurons increases from sensory to association areas [Chen et al., 2015], with frontal areas maintaining information about recent task-relevant stimuli and information that can persist for hours after the task [Fritz et al., 2010]. While the timescale of information seems to vary depending on the area, the speed, noisiness, and dimensionality of spike coding could nonetheless be a key aspect of brains' rapid adaptation and fast processing [Denève et al., 2017, Gerstner et al., 2005].

While many computational models reproduce natural behaviour using either rates or spiking networks, it is very hard to find a computational paper in which the performance of a rate versus that of a spiking network is compared on a similar task. Clopath et al. [2010] developed a voltage-dependent learning rule, which, as the voltage sharply increases when a spike is emitted, can be associated with a spike-timing-dependent rule. They compared the emerging connectivity of a spiking network to the emerging connectivity of a rate network under the same plasticity rule. They show that their rule, provided a spatiotemporally correlated simulation is given to the network, generates asymmetric connectivity within the spiking network which cannot be obtained in a rate network, where the network is simulated by different stationary patterns. Clopath et al. [2010] discuss that this could explain the development of unilateral connectivity underlying fast processing in certain sensory areas. This is a nice example of the emergence of a directed pathway that cannot be obtained on a rate network, but which has not been proven to be critical for a certain function. Sussillo and Abbott [2009] used a reservoir network trained to produce motor activity patterns. Reservoir computing maps an input signal to a higher dimensional neural activity space which dynamics can be non-linear. They are used in general for signal processing, in particular, among other topics, used in speech processing [Araujo et al., 2020]. The authors were able to reproduce cycles of activity measured from the monkeys' motor cortex while performing a movement. Nicola and Clopath [2017] used the same training procedure using a spiking neural network. They were able to reproduce songs from birds. While it seems that bird songs require more temporal precision due to the very fast changes in frequencies [Nicola and Clopath, 2017], it is unclear whether a rate network would not also have achieved such a performance. Reservoir computing relies on the high dimensionality of its dynamics that enables read-out weights to be trained to reproduce any temporal signal. By using spiking neurons instead of rates, the dimensionality of the activity is therefore increased, which underlies the precise behaviour of the network obtained in Nicola and Clopath [2017].

Existing RL models do not seem to perform better when implemented in spiking rather than rate models. Figure 5.3 compares the performance of a spiking [Frémaux et al., 2013] and a rate [Doya, 2000] network on a classical reinforcement learning paradigm: the cartpole task. In a cartpole task, a pendulum is fixed on a cartpole which can move freely along a vertical track, and the agent (both spiking and rate) has to exert a horizontal force on it in order to raise the end of the pendulum up under the influence of gravity (figure 5.3a). The reward given to the agent depends on the angle of the cartpole, so that it is maximal when the cartpole is pointing directly upwards. A successful trial refers to a trial in which the tip of the pendulum spends at least 10 seconds pointing upwards (pointing upwards refers to $|\theta| < \pi/4$, where θ is the angle between the vertical vector pointing upwards and the pendulum). Figure 5.3b presents the performance of a spiking actor-critic agent, and figure 5.3c the performance of a rate actor-critic agent. A median spiking agent takes on average 3500 trials to reach 100 successful trials [Frémaux et al., 2013]. A rate agent takes on average slightly more than 2500 trials to reach 100 successful trials [Doya, 2000]. Although the previous discussion of the temporal precision of spikes would have suggested that a spiking network might be better at this subtle control task, this comparison suggests that it is not the case. Indeed, the continuous agent considered in Doya [2000] seems to perform slightly better than the spiking agent developed in Frémaux et al. [2013]. In this example, it seems coherent that the continuous actor performs better, as the task is of continuous nature Doya [2000], Sutton and Barto [2018]. Thus, the continuous representation enables the fine representation perhaps necessary in this task.

In the next sections, I adapt the spiking actor-critic proposed by Frémaux et al. [2013] to a novel rate network in continuous action and time. Compared to Frémaux et al. [2013], in which neurons activity is represented via a spiking representation, all neurons of the network considered here are modelled using a rate representation. I show that it can learn to reach a goal location in a watermaze equivalent task, and compare its performance to that of the discrete actor-critic presented in chapter 4. I first start by presenting how I proceed to adapt a spiking network to a rate network.

5.3 From a spiking to a continuous rate representation

Let us consider a neuron with an output spike train $Y(t)$ defined by:

$$Y(t) = \sum_{t^f \in \mathcal{F}} \delta(t - t^f), \tag{5.1}$$

where $\mathcal{F}$ is the set of spike times t^f, and δ the Dirac-delta function, defined by $\int_{-\infty}^{+\infty} \delta(s) f(t-s) \mathrm{d}s = f(t)$ for all continuous compactly supported function f, with $\int_{-\infty}^{+\infty} \delta(s) \mathrm{d}s = 1$. In Frémaux et al. [2013], the instantaneous firing rate of the neuron $\rho(t)$ is determined by a convolution of its spike train with a kernel κ:

$$\kappa(t) = \frac{\mathrm{e}^{-t/\tau_\kappa} - \mathrm{e}^{-t/\nu_\kappa}}{\tau_\kappa - \nu_\kappa} \mathcal{H}(t), \tag{5.2}$$

with

$$\rho(t) = \int_{-\infty}^{\infty} Y(s) \kappa(t-s) \mathrm{d}s \equiv (Y \cdot \kappa)(t), \tag{5.3}$$

where τ_κ is the rise constant, ν_κ the decay constant of the kernel and $\mathcal{H}(t)$ the Heaviside step function. The membrane potential of a postsynaptic neuron i, u_i, which receives input from a population of neurons indexed by j, can be computed using the spike response model:

$$u_i(t) = \sum_j w_{ij} \sum_{t_j^f \in \mathcal{F}^j, t_j^f > \hat{t}_i} \varepsilon(t - t_j^f) + \xi \mathcal{H}(t - \hat{t}_i) \exp\left(\frac{\hat{t}_i - t}{\tau_m}\right), \tag{5.4}$$

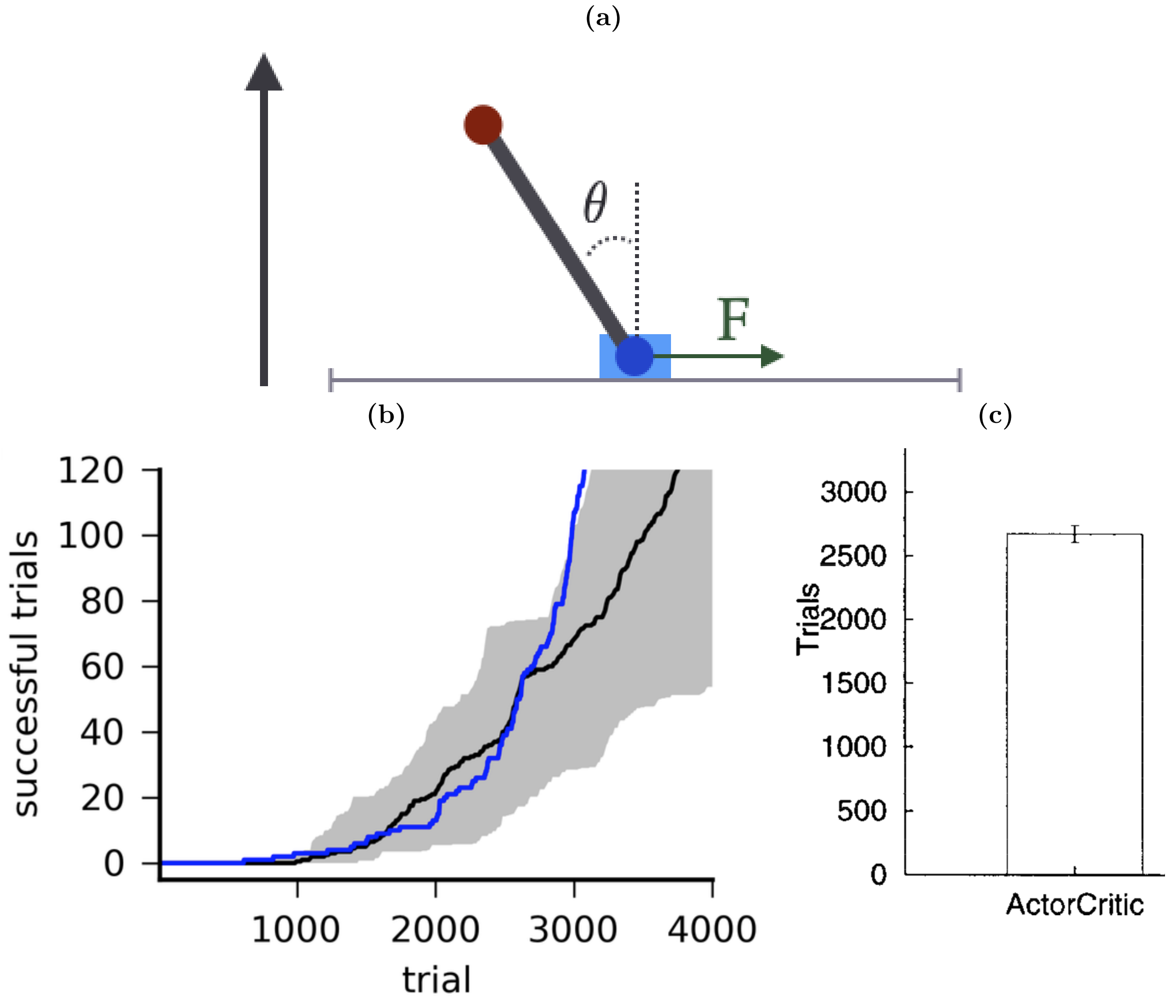


Figure 5.3: a) Cartpole task. The agent has to move the tip of the pendulum (red) up by exerting a horizontal force on the cart (green rectangle) which can move freely along a limited vertical track. A successful trial is defined as a trial where the pendulum is maintained up ($|\theta| < \pi/4$) for more than 10 seconds. A trial is interrupted either when the cartpole reaches the end of the trail or when the pendulum over-rotates ($|\theta| > 5\pi$). Reward depends on the height of the pendulum, $r(t) \propto \cos(\theta)$. b) Performance of the spiking agent from Frémaux et al. [2013], the dark line shows the median and the blue line an example agent, and the grey shaded area the quartiles of 20 independent agents. It takes the "median agent" approximately 3500 trials to achieve 100 successful trials. Figure adapted from Frémaux et al. [2013]. c) Comparison of the number of trials until 100 successful swing-ups with the actor-critic, value-gradient-based policy with an exact and learned physical models. Figure adapted from Doya [2000].

where w_{ij} is the synaptic strength from neuron j to neuron i, and $\mathcal{F}^j$ the set of firing times t_j^f of neuron j. In the second term of equation (5.4), $\mathcal{H}$ is the Heaviside step function, $\hat{t}_i$ is the last spike of neuron i before time t, and τ_m is the membrane time constant of the neuron. Here, $\xi < 0$, the second term in equation (5.4), scales the refractory effect: it reduces the membrane potential for a short period after the neuron's last spike $\hat{t}_i$, so that the neuron cannot fire in this time window. The function ε defines the time course of an excitatory postsynaptic potential and its expression is functionally identical to the expression of $\kappa(t)$ in (5.2), only to one detail of a scaling factor, and with different rise and decay time values. Given a membrane potential u_i, a spike train can be generated using an inhomogeneous Poisson process in which the probability of emitting a spike is defined according to the membrane potential [Frémaux et al., 2013]. An inhomogeneous Poisson process defines a spike train which is non homogeneous in time: sometimes the cell can fire more than during other periods of time [Frémaux et al., 2013], as opposed to homogeneous Poisson process in which bursting periods are not possible. Figure 5.4a shows the classical shape of an excitatory postsynaptic potential.

I propose that adapting a spike representation to a rate representation can be done by incorporating the distributed time delays involved in the convolution of a neuron's spike train with the kernel κ in equations (5.3) and (5.4) in the activity dynamics. I adapt equation (5.4) to a rate representation, by defining the evolution of the ith neurons' membrane potential according to a second order differential equation:

$$Qu_i(t) = \varepsilon \sum_{j=1} w_{ij}\rho_j(t), \tag{5.5}$$

where w_{ij} denotes the synaptic strength from the jth neuron to the ith neuron, ρ_j refers to the firing rate of the j neuron, defined in equation (5.7) and the operator Q is a second order differential operator defined by:

$$Q = \left(1 + \tau_m \frac{\mathrm{d}}{\mathrm{d}t}\right)\left(1 + \tau_s \frac{\mathrm{d}}{\mathrm{d}t}\right). \tag{5.6}$$

A solution u of $Qu = \delta$, where δ refers to a Dirac-delta input mimicking a presynaptic spike, gives a post-synaptic potential of the form shown in figure 5.4. which approximates very closely the responses of neurons to presynaptic spikes. Figure 5.4a shows the recording of a neuronal response from a single spike (left) and three successive spikes (right). Figures 5.4b and 5.4c show the evolution of the responses of a unit whose activity evolves according to the second order linear operator (5.6) with input given by one Dirac-delta and three Dirac-delta functions respectively, mimicking spike inputs from presynaptic neurons. By acting on the neuron's activity u_i, the operator Q generates a response that mimics the delay and decay times of biological neurons, that are modelled as a convolution of the kernel κ and the spike train in equations (5.2) and (5.4). Figure 5.4 shows that the simulated responses mimic quite closely the responses observed in neuronal recordings.

From the membrane potential u_i, I define the firing rate of neuron i using a sigmoid function, so that $\rho_i(t) = \rho(u_i(t))$, where:

$$\rho(u) = \frac{\rho_0}{1 + \exp\left(-\beta\left(u - h\right)\right)}, \tag{5.7}$$

where ρ_0, β and h are parameters that define the threshold and the sharpness of the activation function. From a biological point of view, this function mimics the firing rate profile of biological neurons [Galizia and Lledo, 2013], representing the fact that biological neurons have a maximal firing rate, and a variable response range, defined by the choice of ρ_0, β and h. Figure 5.5 shows the shape of the sigmoid for a wide range of inputs. The next section presents how this continuous framework is integrated within an actor-critic architecture.

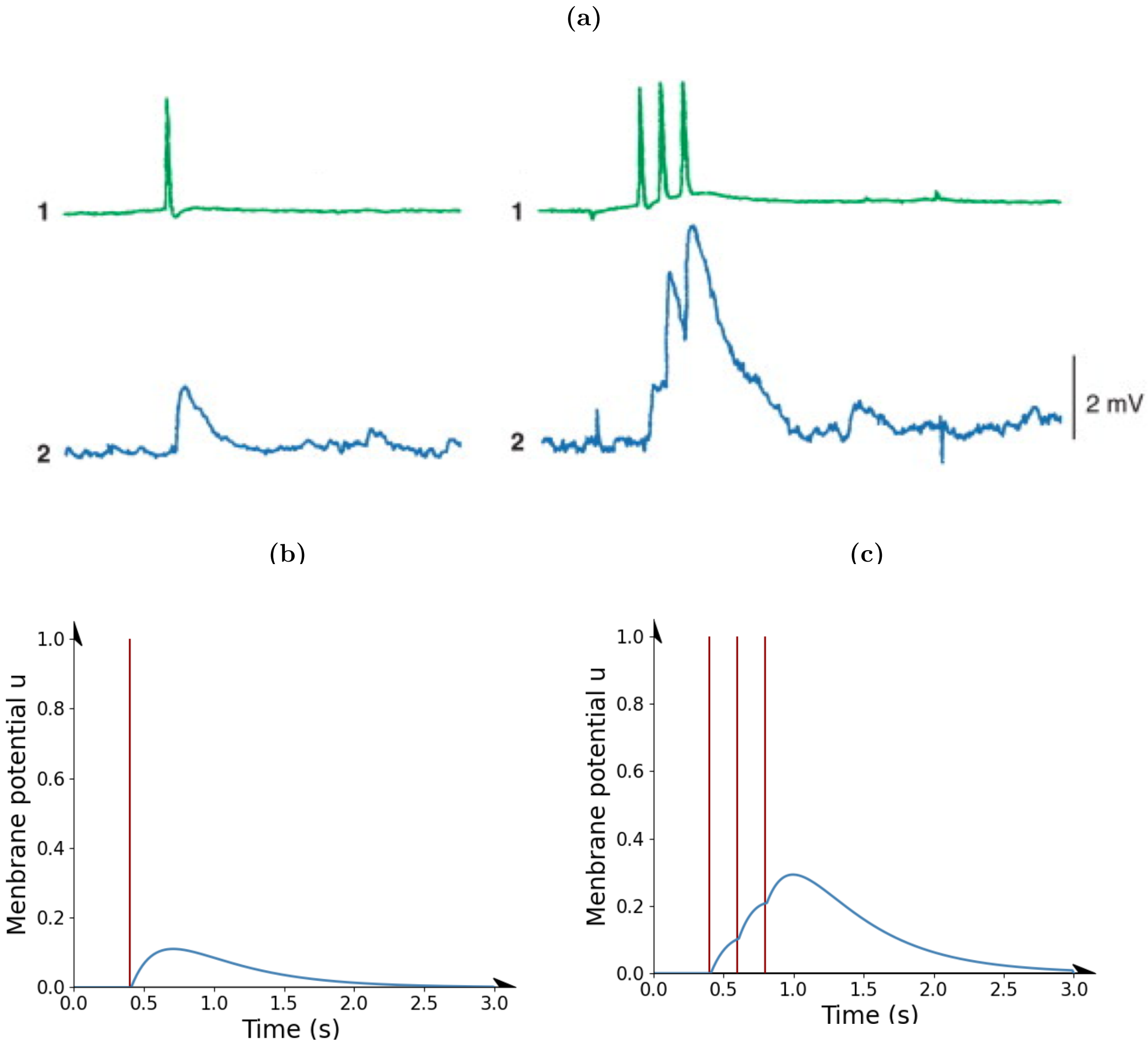


Figure 5.4: Excitatory Post Synaptic potential (EPSP) of a typical neuron (a) and of a simulated neuron (b) from equation (5.4). (a) EPSP of a post-synaptic neuron (down, blue) in response of a single pre-synaptic spike (left) and a sequence of three spikes (right). Adapted from Hammond [2014]. (b) response of a simulated neuron (green) from a single-spike input (red), and (c) response of a simulated neuron (green) to three spikes input (red). The membrane potential u is simulated using equation (5.5) in response to one or three Dirac-delta inputs.

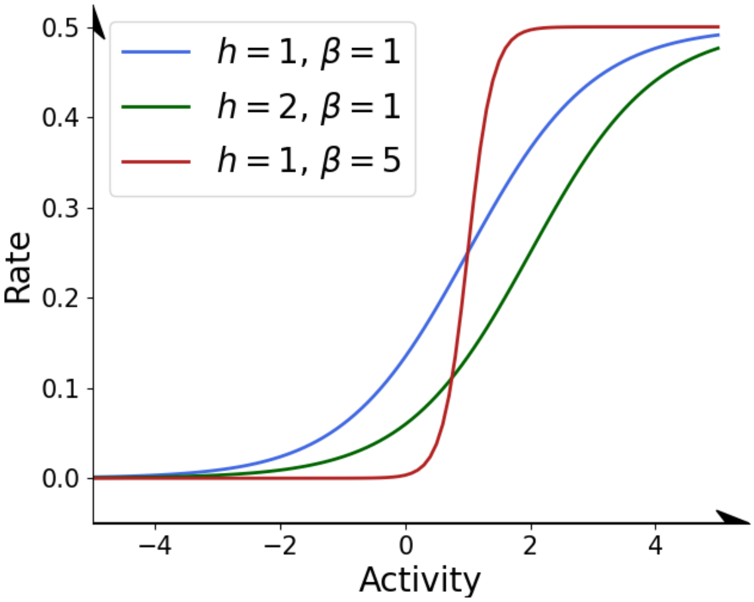


Figure 5.5: Sigmoid function expressed by equation (5.7). h defines the value of the activity where the slope is the highest, which also characterises a threshold of activation of the response, β determines the sharpness of the slope, which also shapes the sensitivity of the response around the activation threshold. Different lines have been plotted for different values of the parameters, in red, the slope is very high ($\beta = 5$), in green ($h = 2$), the threshold is higher than in blue ($h = 1$). The different parameter values are summarised in the legend. All curves have a maximal asymmptotic value $\rho = 0.5$.

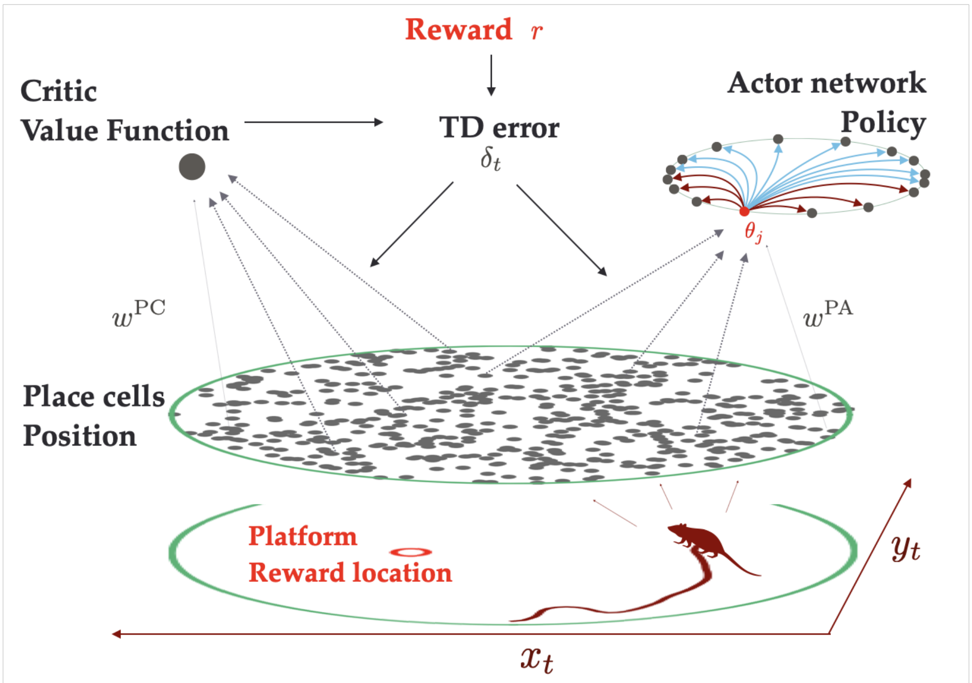


Figure 5.6: Schema of the network proposed in Frémaux et al. [2013], adapted to the Morris watermaze. A layer of place cell units covering the whole environment (bottom) encodes the position of the agent (red mouse). Their activity is transmitted to a critic cell which computes the value function according to equation (5.9), and simultaneously to the actor network, from equation (5.12), which is used to determine the policy. The actor network is modelled as a ring attractor network, in which every units is excitatory towards units encoding similar directions (red arrows), and inhibitory towards units encoding opposite directions (blue arrows). After a move, the critic activity is used to form the TD error according to equation (3.15), which is used to update the connection weights w^{PC} and w^{PA}. Note that compared to the model by Frémaux et al. [2013], the critic network is composed of only one cell, as Frémaux et al. [2013] used a network in order to be able to take an average to compute the value function (5.11).

5.4 An actor-critic architecture in continuous space, actions and time for navigation in the watermaze

In this section, I present the dynamics involved in the continuous actor-critic model, and implement it for navigation in a watermaze equivalent. The continuous model, adapted from Frémaux et al. [2013], is very similar to the actor-critic architecture presented in chapter 4, with the addition that every neuron has dynamics governed by the action of the second order differential operator given by equation (5.6), and that the action representation is continuous rather than discrete. All the equations have been adapted from the spiking network proposed by Frémaux et al. [2013] into a rate network, by using the correspondence between spiking and rate responses presented in section 5.3. Figure 5.6 presents a schema of the model, very similar to that of figure 4.1, in that the position of the agent is encoded within a place cell network, composed of $N_P = 500$ neurons, used by a critic to estimate the value function and an actor network to estimate the policy. If the agent is in position p_t, the activity of the jth place cell is defined by:

$$\rho_j^{\mathrm{P}}(t) = \rho_{\mathrm{PC}} \exp\left(-\frac{\|p_t - c_j\|}{\sigma_{\mathrm{PC}}^2}\right), \tag{5.8}$$

where σ_{PC}^2 defines the width of the activity profile, c_j the centre of the jth place cell and ρ_{PC} is a parameter that defines the global activity of the place cell network. The dynamics of the critic cell activity is given by:

$$Q^{\mathrm{C}} u^{\mathrm{C}}(t) = \varepsilon_{\mathrm{PC}} \sum_{j=1}^{N_P} w_j^{\mathrm{PC}}(t) \rho_j^{\mathrm{P}}(t), \tag{5.9}$$

where $w_j^{\mathrm{PC}}(t)$ denotes the weight from the jth place cell to the critic cell, $\varepsilon_{\mathrm{PC}}$ tunes the global activity level and the operator

$$Q^{\mathrm{C}} = \left(1 + \tau_m^{\mathrm{C}} \frac{\mathrm{d}}{\mathrm{d}t}\right)\left(1 + \tau_s^{\mathrm{C}} \frac{\mathrm{d}}{\mathrm{d}t}\right) \tag{5.10}$$

is a similar operator to that presented in equation (5.6), with the constants τ_m^{C} and τ_s^{C} the rise and decay time of the critic neuron activity. The value function is directly computed from the activity of the critic cell:

$$V(t) = \nu u^{\mathrm{C}}(t) + V_0, \tag{5.11}$$

where ν and V_0 are constants representing the gain and basic level of the value function.

The actor network is composed of $N_A = 180$ action cells covering discretised directions from 0 to 2π (see figure 5.6). The ith action cell is associated with the angle $\theta_i = 2\pi i/N_A$. The dynamics of its activity is given by the sum of a place cell input, a recurrent input, and a noise term:

$$Q^{\mathrm{A}} u_i^{\mathrm{A}}(t) = \varepsilon_{\mathrm{PA}} \sum_{j=1}^{N_P} w_{ji}^{\mathrm{PA}}(t) \rho_j^{\mathrm{P}}(t) + \varepsilon_{\mathrm{A}} \sum_{l=1}^{N_A} w_{li}^{\mathrm{AA}}(t) \rho^{\mathrm{A}}(u_l^{\mathrm{A}}(t)) + n_i(t), \tag{5.12}$$

where $Q^{\mathrm{A}} = \left(1 + \tau_m^{\mathrm{A}} \mathrm{d}/\mathrm{d}t\right)\left(1 + \tau_s^{\mathrm{A}} \mathrm{d}/\mathrm{d}t\right)$ represents the natural dynamics of a neuronal response as in equation (5.6) (see figure 5.4), with τ_m^{A} and τ_s^{A} the rise and decay time of the actor neurons activity, and $w_{ji}^{\mathrm{PA}}(t)$ is the weight from the jth place cell to the ith action cell. $\varepsilon_{\mathrm{PA}}$ and ε_{A} are parameters tuning the strength of the place cell and recurrent inputs. The second term of equation (5.12) is a recurrent term, in which w^{AA} is a matrix of coupling weights within the actor network allowing the formation of a bump of activity within the recurrent network. The recurrent weight between the lth and the ith action cell is defined as follows:

$$w_{li}^{\mathrm{AA}} = \frac{w_-}{N_A} + w_+ \times \frac{\exp(\xi \cos(\theta_i - \theta_l))}{\sum_k \exp(\xi \cos(\theta_i - \theta_k))}. \tag{5.13}$$

The recurrent weights guarantee that an action cell is locally excitatory towards cells encoding similar directions, and inhibitory towards cells encoding different angles. w_- and w_+ define the strength of inhibition and excitation, and ξ defines the width of the recurrent connections, such that a higher ξ value represents a wider connectivity profile. Figure 5.7 shows the shape of the recurrent weight profile for an exemplar direction cell θ_j. The recurrent weights are represented by blue arrows for inhibitory connections and red

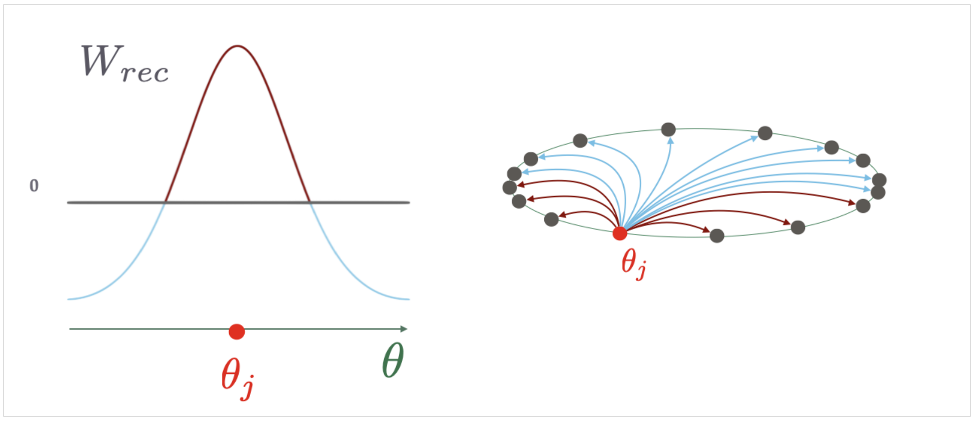


Figure 5.7: Recurrent weight profile from an exemplar action cell θ_j, computed from equation (5.13) using $\xi = 8$, $w_- = -1$ and $w_+ = 1$. The grey line highlights $y = 0$ for more clarity. Above that line, the weights are excitatory (red section), and under it, inhibitory (blue section).

arrows for excitatory connections within the actor network in figure 5.7, and added to the main figure 5.6. In equation 5.12, the recurrent weights act on a sigmoid transformation of the action cell activity using the sigmoid function defined in equation (5.7).

The last term of equation (5.12) is a stochastic noise term $n_i(t)$, evolving according to a stochastic differential equation, added to the activity of action cell i to allow some exploration of the environment. For action i, the noise term $n_i(t)$ evolves according to:

$$\frac{\mathrm{d}n_i(t)}{\mathrm{d}t} = -\frac{n_i(t)}{\tau_n} + n_i^{In}(t), \tag{5.14}$$

where n_i^{In} is a stochastic input term to the noise defined according to:

$$n_i^{In} = \rho_n(t) \exp\left(- \frac{\sin\left(\frac{\theta_i - \theta_{c_n(t)}}{2}\right)^2}{2\sigma_n^2} \right), \quad \forall i \in 1 \ldots N_A, \tag{5.15}$$

where θ_i refers to the angle of the direction of action i, σ_n is a set parameter that defines the width of the noise input around its centre $\theta_{c_n(t)}$, indexed $c_n(t)$, and $\rho_n(t)$ is the amplitude of the noise input. Both $\rho_n(t)$ and $\theta_{c_n(t)}$ depend on the performance of the agent, which I define from the amplitude of the learned place cell input $\sum_{j=1}^{N_P} w_{ji}^{\mathrm{PA}}(t)\rho_j^{\mathrm{P}}(t)$ in equation (5.12). I first define the variable $h_n(t)$ as a measure of learning of the agent:

$$h_n(t) = \max_i \left(\sum_{j=1}^{N_P} w_{ji}^{\mathrm{PA}}(t)\rho_j^{\mathrm{P}}(t) \right) - \min_i \left(\sum_{j=1}^{N_P} w_{ji}^{\mathrm{PA}}(t)\rho_j^{\mathrm{P}}(t) \right). \tag{5.16}$$

The actor weights w_{ji}^{PA} are initialised from a normal distribution with mean $\mu = 0.005$ and standard deviation $\sigma = 0.0001$. Early on in learning, the difference $h_n(t)$ is therefore very low. With learning, the actor weights w_{ji}^{PA} progressively evolve (according to equation (5.25)) so that the place cell input $\sum_{j=1}^{N_P} w_{ji}^{\mathrm{PA}}(t)\rho_j^{\mathrm{P}}(t)$ becomes more selective to the direction leading to the goal location, and the difference $h_n(t)$ increases. The difference $h_n(t)$ is hence a measure of the sharpness of the place cell drive to the actor network. $h_n(t)$ is used to define both the amplitude and the mean of the input to the noise term.

In equation (5.15), $\rho_n(t)$ determines the amplitude of the level of noise provided to the actor network. It is defined according to the current sharpness of the input from the place cells in equation (5.16) by:

$$\rho_n(t) = \frac{\rho_n^o}{h_n(t)}, \tag{5.17}$$

such that when the input from the place cell network to the actor network indicates a clear preference for a particular action, the difference in the denominator should be high, leading to a lower noise level. This guarantees a high level of exploration early on during learning which decreases with learning. Here ρ_n^o is a constant. The action cell indexed $c_n(t)$ which define the centre of the noise input $\theta_{c_n(t)}$ in equation (5.15) is a stochastic variable whose distribution also depends on the strength of the place cell input:

$$c_n(t) \sim \mathcal{P}^c(t), \tag{5.18}$$

where $\mathcal{P}^c(t)$ is a distribution over the action cell indexes $[1, \cdots, N_A]$. I first define the centre of the distribution $\mu_c(t)$ by:

$$\mu_c(t) = \left(\mathrm{argmax}(u^A(t)) \pm \frac{\mu_n}{h_n(t) \times N_A} \;\middle|\; N_A\right) + 1, \tag{5.19}$$

where $\left(\cdot \middle| N_A\right)$ refers to the remainder from the Euclidian division by N_A, and μ_n is a parameter that defines the variation of the noise centre around the maximum of the activity. The centre varies around the preferred direction $\mathrm{argmax}(u^A(t))$ by a factor depending on the strength of the place cell input ($h_n(t)$, equation (5.16)), $\mu_n/(h_n(t) \times N_A)$, so that early on in learning, the noise level centre deviates more from the current preferred direction than later on, to guarantee exploration. The distribution for the centre of the noise input is then:

$$\mathcal{P}_k^c(t) = p_k \exp\left(-\frac{(\mu_c(t)-k)^2}{2\sigma_n(t)^2}\right), \quad \forall k \in [1, \cdots, N_A], \tag{5.20}$$

where p_k is a normalisation constant, and the width $\sigma_n(t)$ of the distribution $\mathcal{P}^c(t)$ is defined according to the performance:

$$\sigma_n(t) = N_A \frac{\sigma_n^0}{h_n(t)}. \tag{5.21}$$

With an input for which both centre (equation (5.20)) and amplitude (equation (5.18)) evolve depending on the strength of the place cell input, the noise level is defined so that it drives the actor network early on in learning. With learning, the centre of the noise input becomes closer on average to the preferred direction encoded within the place cell input, and its amplitude becomes more even across the network, so that the place cell input takes the lead in driving the actor network activity.

The activity of the actor network u^{A} (equation (5.12)) is used to perform a vectorial sum of the possible directions, so that the resulting displacement of the agent is defined by:

$$\frac{\mathrm{d}p(t)}{\mathrm{d}t} = \alpha \times \left(\frac{\sum_{k=1}^{N_A} u_k^{\mathrm{A}}(t) \times a_k}{\left\|\sum_{k=1}^{N_A} u_k^{A}(t) \times a_k\right\|}\right), \tag{5.22}$$

where $a_k = [\cos(\theta_k) \ \sin(\theta_k)]$ is the direction vector encoded by the kth action cell, and $\mathrm{d}p(t)/\mathrm{d}t$ derive the position of the agent.

According to the expression of the value function in continuous time given by equation (3.9), rewards are given to the agent as a reward rate. In Frémaux et al. [2013], the authors explain that this “reflects the fact that ‘natural’ rewards, and reward consumption, are spread over time, rather than point-like events”. It is interesting to think about this as a result of a continuity in the natural progress of events: even if a reward can be perceived as being very temporally defined, for example at the exact moment in which the rat reaches the platform in a watermaze task, the information of having touched the platform, and the resulting activation of the reward networks are extended in time following the first touch of the platform (*e.g.*, perhaps due to the time it takes to feel safe). Therefore, it feels natural to define a reward scalar that also has a rise time and a decay time from the time onset in which the goal is found. Following Frémaux et al. [2013], the reward rate $r(t)$ is defined from the difference of two decaying reward traces r_a and r_b according to the dynamics:

$$\frac{\mathrm{d}r_a(t)}{\mathrm{d}t} = -\frac{r_a}{\tau_a} + R(t); \qquad \frac{\mathrm{d}r_b(t)}{\mathrm{d}t} = -\frac{r_b}{\tau_b} + R(t), \tag{5.23}$$

where τ_a and τ_b are the time constants of the two reward traces, and $R(t)$ represents the reward delivery, such that $R(t) = 0$ most of the time, and a non zero reward is delivered when some event happens: either when the agent finds the goal, which gives rise to a positive reward, or when a collision with the wall occurs, which gives rise to a slightly negative reward. The total reward $r(t)$ is computed as the difference of those two reward traces:

$$r(t) = \frac{r_a(t) - r_b(t)}{\tau_a - \tau_b}. \tag{5.24}$$

At most times, the reward rate is close to 0. When a reward delivery occurs, the reward rate rises up progressively and then decays, with a rise time and decay time that depend on the differences between the two time constants τ_a and τ_b. In fact, by considering the reception of a reward a $t = 0$, one can solve the equation (5.23) analytically for r_a and r_b. Using (5.24) and (5.23) and solving $\mathrm{d}r(t)/\mathrm{d}t = 0$ gives $T_{max} = (1/\tau_a - 1/\tau_b)^{-1} \ln(\tau_b/\tau_a)$. Figure 5.8 shows the evolution of the reward rate and reward traces after the reception of a reward delivery.

From the reward rate and the critic activity, the TD error can be computed from equation (3.15) that is used to update the weights between the place cells and the critic cell and the place cells and the actor network. Heuristically adapting the weight update

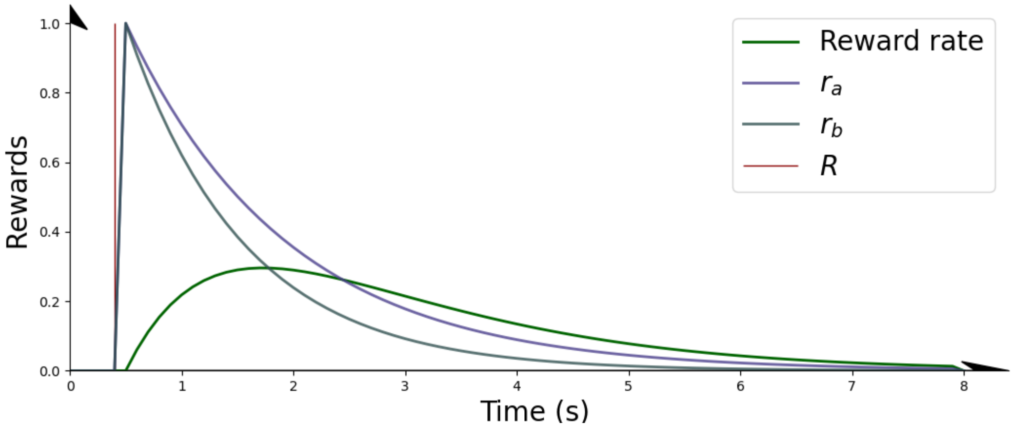


Figure 5.8: Reward dynamics. The reward rate (dark green), obtained by computing equation (5.24) from the reward traces r_a and r_b governed by the dynamic given in equation (5.23) with $R = 1$ at $t = 0.4$ seconds (marked by a dark red vertical line)

rule proposed by Frémaux et al. [2013] to our rate model, the actor weight between the jth place cell and the ith action cell are updated according to:

$$\frac{\mathrm{d}}{\mathrm{d}t} w_{ji}^{\mathrm{PA}}(t) \;\propto\; \delta(t) \frac{\partial u_i^{\mathrm{A}}(t)}{\partial w_{ji}^{\mathrm{PA}}} \times u_i^{A}(t), \tag{5.25}$$

where $\delta(t)$ is the continuous TD error derived in equation (3.15), and u_i^{A} is the current activity of the ith action cell. The weights are reinforced if the error is positive and according to the role of the jth place cell on the ith action cell activity. The critic weight between the jth place cell and the critic cell is updated according to:

$$\frac{\mathrm{d}}{\mathrm{d}t} w_{j}^{\mathrm{PC}}(t) \;\propto\; \delta(t) \times \frac{\partial V(t)}{\partial w_{j}^{\mathrm{PC}}}. \tag{5.26}$$

The following section discusses the performance of the agent, and describes the resulting dynamics of the actor network activity given by equation (5.12).

5.5 The agent can navigate to the goal adopting a smooth trajectory

In this section, I show that the agent is able to learn to navigate to a fixed goal location, with a smoother control than the discrete actor-critic agent described in chapter 4. I also discuss that the complexity of the model leads to unreliable performances when changing the level of noise that the actor network receives (equation (5.12)).

5.5.1 The network learns to navigate to a goal location

The agent can learn to navigate to a fixed goal location. Figure 5.9 shows the average escape latencies of the agents generated from the model. The latencies of the agents decrease with repeated exposure to the goal. Learning relies on equations (5.25) and (5.26), which are very similar to the update rules for the connection weights of the discrete version of the actor-critic architecture presented in chapter 4, given by equations (4.8) and (4.10). Figure 5.10 compares the latencies for the discrete agent presented in section 4.2 and the continuous agent presented here. One can see that the discrete actor-critic agent presented in chapter 4 performs slightly better in this task. The continuous agent can learn how to reach the goal in a few trials. However, after a few trials, the latencies tend to increase. This slight increase could be due to the heuristic of the definition of the learning rules (5.25) and (5.26). All parameters and implementation details are given in Appendix B.

The degree of exploration of the agent reduces with learning. Figure 5.9 shows that the better is the performance, the smaller is the standard deviation around the mean of the latencies of the agents (as can be seen with the smaller error bars in trials of low latency). Additionally, the noise level drives the actor network early on in learning, and the place cell input drives the actor network later on learning. Figure 5.11 shows the balance within the different inputs that the actor network receives, from equation (5.12), and the resulting trajectories, early on in learning (figure 5.11a) and after learning (figure 5.11b). Early on in learning (figure 5.11a.i and 5.11a.ii), the noise level is quite sharp, and the place cell input almost uniform across the action cells. This is a consequence of the definition of the noise level as depending on performance (equation (5.15)). The actor network activity is therefore shaped mostly by the noise input. As a result, the trajectory of the agent is very exploratory (figure 5.11a.iii). Later on in learning (figure 5.11b.i and 5.11b.ii), the input from the place cells is sharper, therefore the noise level is

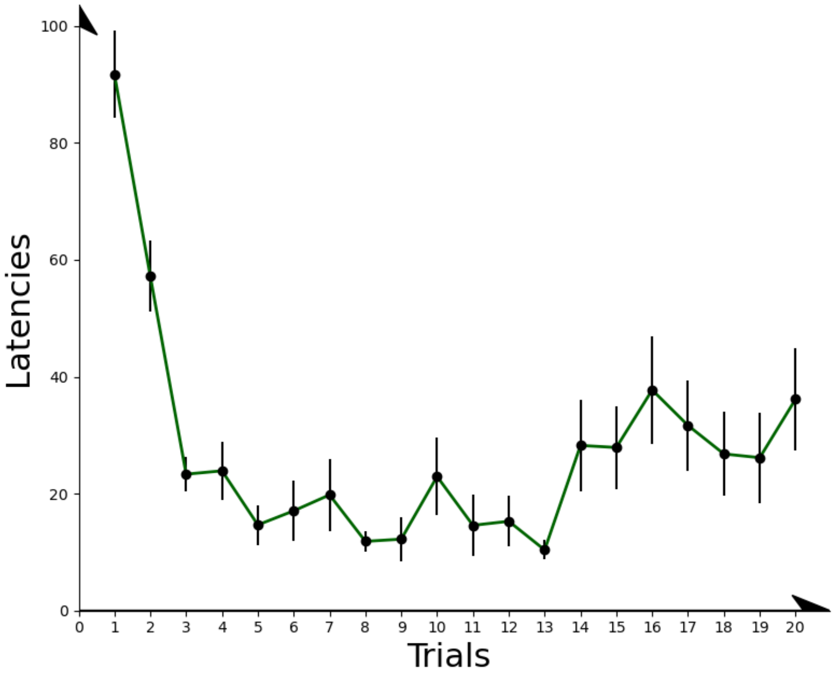


Figure 5.9: Latencies generated by the model, with error bars computed for n=30 independent runs. Parameter values are given in appendix B

almost uniform across the action space. The actor network activity is mostly determined by the input from the place cell network which encodes the best direction to follow. This leads to a trajectory almost direct to the goal (figure 5.11b.iii).

A continuous approach enables the generation of smoother trajectories, underlying a smoother control of the movement. Compared to trajectories of the discrete actor-critic agent (4.3a.iii from the discrete actor), the trajectories generated from the continuous model (figure 5.11a.iii) seem more consistent with animal behaviour (figure 2.10a), which confirms that the control of the trajectory by a continuous approach is smoother than by a discrete approach.

5.5.2 Effect of the level of noise on the dynamics of the network

The agent implemented from the continuous model given in section 5.4 can learn to navigate to a fixed goal location using a smoother trajectory than the ones of a discrete actor-critic agent. However, the dynamics of the network generates less reliable perfor-

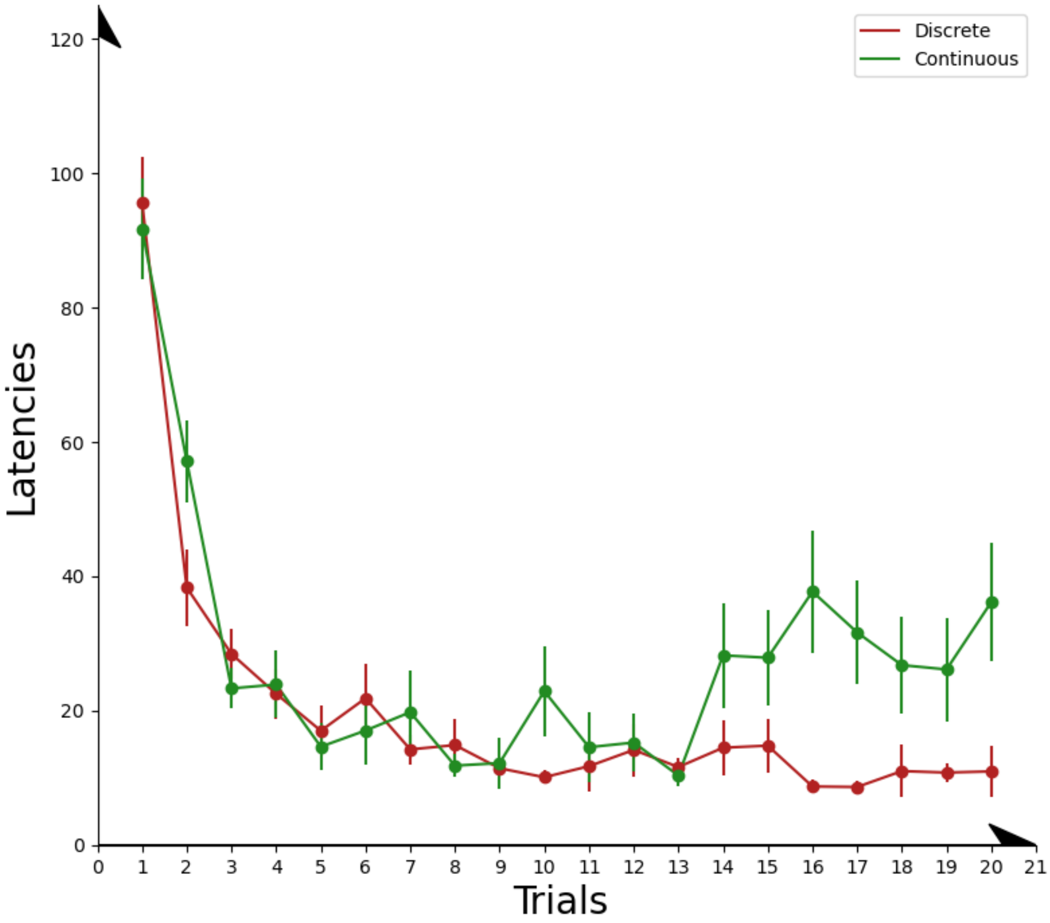
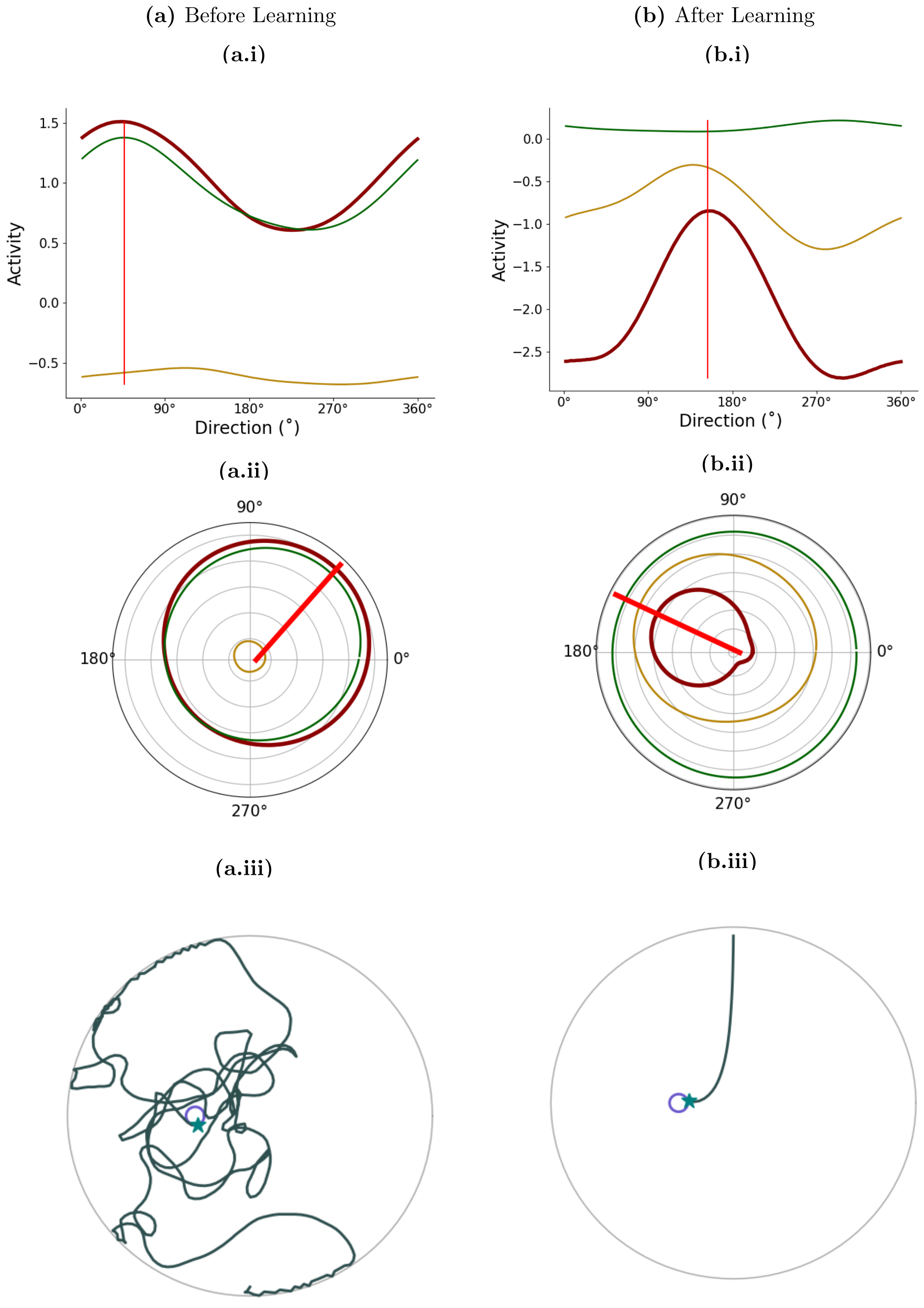


Figure 5.10: Latencies generated by the continuous model (green) and by the discrete model presented in chapter 4 section 4.2 (red), with error bars computed for n=30 independent runs. The continuous agents learn, but the performance is less stable than the one of the discrete actor-critic agent, due to a more complex dynamics within the whole network. Parameter values are given in appendix B.


(a) Before Learning

(b) After Learning

(a.i)

(b.i)

(a.ii)

(b.ii)

(a.iii)

(b.iii)

Figure 5.11: i,ii) Balance of the actor network activity from its different inputs (showing the value of the different inputs to the actor cells in equation (5.12)) and iii) resulting trajectory of the agent, before (a) and after learning (b). (a.i) Before learning, the input received from the place cells (yellow curve) is almost uniform over the action space, therefore the actor activity (dark red) is mainly shaped by the noise level (green). The direction selected is shown by a red bar and is computed using vectorial sum of the actor activities (equation (5.22)). (a.ii) similar plot but in polar coordinates, for more clarity over how the decision is made. (a.iii) As a result, the trajectory is highly exploratory. (b) After learning, the actor activity is shaped by the place cell input, which is sharp around the preferred direction, leading to a noise level almost uniform over the action space, and to a direct trajectory toward the goal location (b.iii).

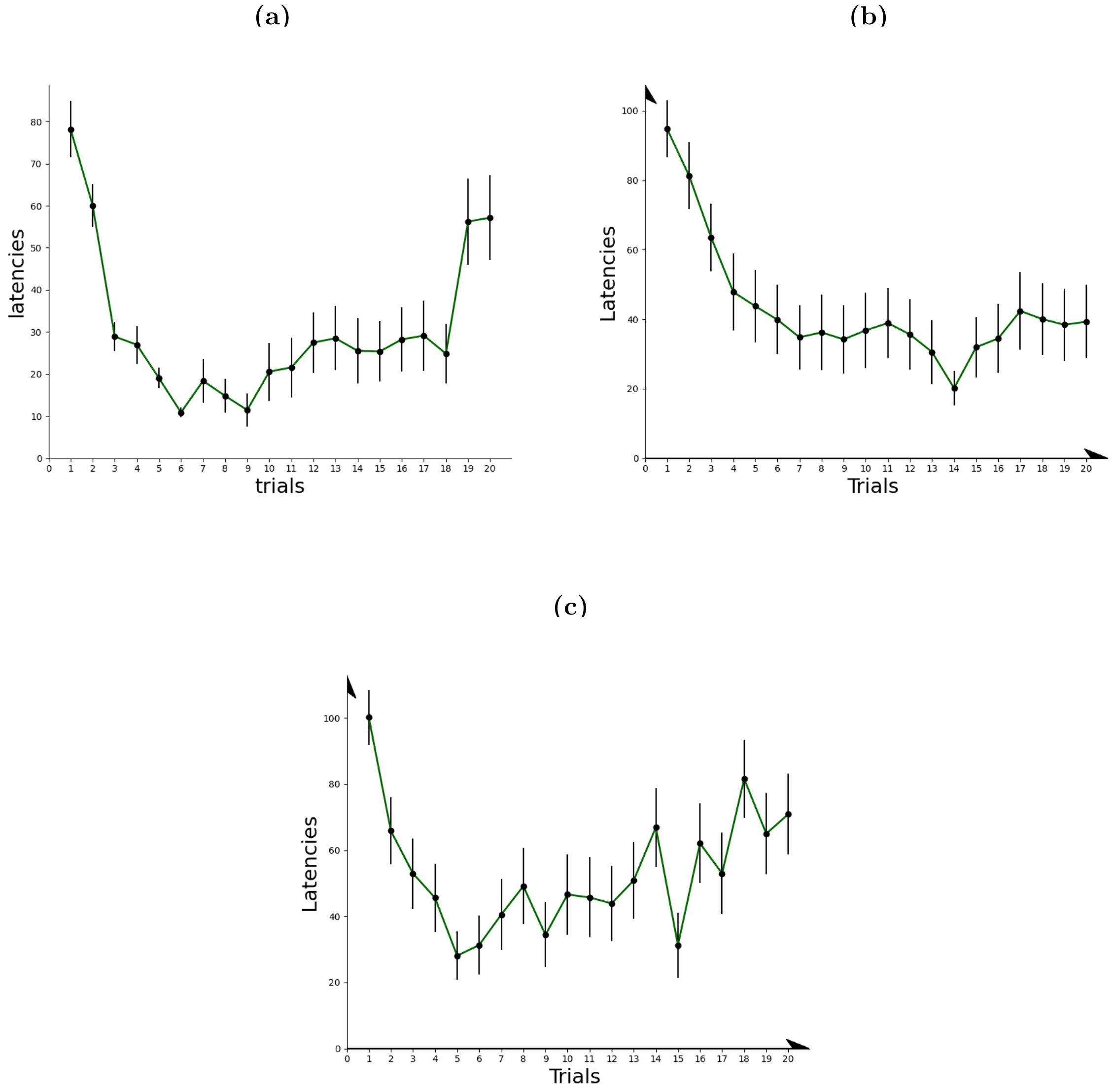
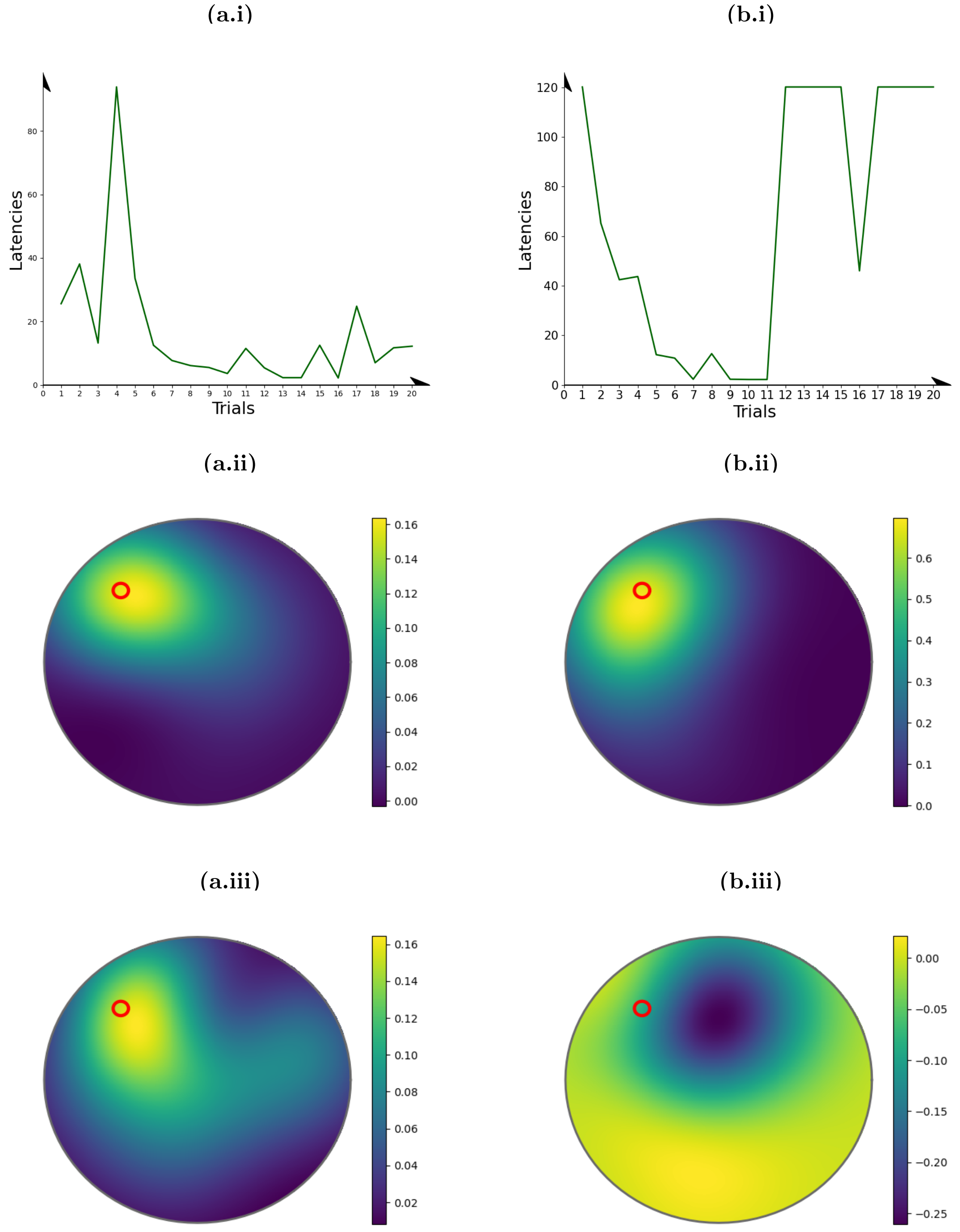


Figure 5.12: Latencies of the agents for a very low (a), a medium (b) and a very high (c) level of noise, with error bars computed for n=30 independent runs. (a) For a very low value of ρ_n^o ($\rho_n^o = 0.001$), the agents reach good performance in a few trials. However, after a few trials with short latencies, the latencies increase, an indication that the performance starts to be impaired. (b) For a medium value of the noise level ($\rho_n^o = 0.5$), the latencies never reach a value as small as obtained with other values of the noise level, but are more stable across the experiment. (c) For a high value of ρ_n^o ($\rho_n^o = 2$), the performance is not steady. The first trials indicate signs of learning, as can be seen on the slow decrease of the mean latencies, but the performance deteriorates afterwards, as the latencies increase.



(a) Successful agent: the value function is stable and consistent across the trials

(b) Unsuccessful agent: the value function becomes inconsistent

(a.i)

(b.i)

(a.ii)

(b.ii)

(a.iii)

(b.iii)

Figure 5.13: Two independent runs of the experiment corresponding to the mean latencies given in figure 5.12a. (a) When the agent shows steady good performance, as seen in the latencies (a.i), the corresponding value function is stable throughout the whole experiment (shown here for trials 5 (a.ii) and 20 (a.iii)). (b) When the performance degrades, as seen in the latencies (b.i) which become maximal after trial 11, the shift in performance is associated with a change in the value function (here showing the shift between trial 11 (b.ii) and trial 12 (b.iii)). The value function is computed from the place cell input to the critic cell (equation (5.9)), shown as a heatmap.

mance compare to that of the discrete actor-critic network (figure 4.2). This could be due to the heuristic definition of the learning rules, or to the choice of the implementation scheme for a stochastic differential equation. Figure 5.12 shows the latencies obtained for different values of the noise amplitude ρ_n^o in equation 5.17. For a very low value of ρ_n^o ($\rho_n^o = 0.001$, figure 5.12a), the agents reach good performance in a few trials. However, after a few trials with short latencies, the latencies increase, indication that the performance starts to be impaired. For a medium value of the noise level ($\rho_n^o = 0.5$, figure 5.12b), the latencies never reach a value as small as obtained with other values of the noise level, but maintain a steady level across the experiment. For a high value of ρ_n^o ($\rho_n^o = 2$, figure 5.12c), the performance is less reliable. The first trials indicate signs of learning, as demonstrated by the slow decrease of the mean latencies, but the performance deteriorates afterwards, as the latencies increase.

The performance can be altered quickly, and the shift is accompanied by an alteration in value function that becomes inconsistent. Figure 5.13 compares the latencies and value functions for two independent runs. A stable successful agent (figure 5.13a), whose performance is stable after learning (figure 5.13a.i), also has a stable value function. Figures 5.13a.ii and 5.13a.iii show the value functions for trial 5 and trial 20, and by comparing the two heatmaps, one can see that the value function stays steady across the experiment. However, an unsuccessful agent can see its performance impaired in one trial (figure 5.13b). Indeed, in figure 5.13b.i, the agent achieves almost optimal performance on trial 11, which then deteriorates, as indicated by the latencies which reach maximum values from trial 12 onwards. At trial 11 (figure 5.13b.ii), the estimated value function still clearly indicates a maximum centred around the goal location, which is consistent with the value function profile necessary to navigate to a singular goal location (discussed in section (4.3.1)). However, during trial 12, the value function changes to a profile that is not consistent with the location of the goal anymore (figure 5.13b.iii). These gaps or jumps in performance seem to be due to the fairly complex dynamics of the network, as the performance depends on many parameters and on a stochastic term, and the behaviour is defined using a heuristic learning rule. Perhaps using another numerical scheme, or other learning rules, as in [Vasilaki et al., 2009], could allow learning to be permanent.

In this section, I showed that the continuous actor-critic agent can learn to navigate to a fixed goal location using a smooth trajectory. I described that its dynamics are very delicate, and illustrated that a slight change in the noise level can impair performance. Nonetheless, our results are consistent with the results Frémaux et al. [2013] and Vasilaki et al. [2009] obtained from the implementation of a spiking policy gradient agent in a watermaze-equivalent task. Indeed, their spiking agent learns to navigate to a goal location in a number of trials comparable to the performance of our continuous actor-critic agent. The rate approach, however, is much more computationally efficient. By computationally efficient, I mean here that a spiking representation requires the generation of independent spike trains from the implementation of stochastic Poisson processes, as described in section 5.3. In the rate network considered here, the activity of cells simply evolves according to a linear operation. Therefore, this suggests that the resolution of spike coding is perhaps not necessary for navigation to a fixed goal location.

5.6 Conclusion

In this chapter, I have shown that the choice of representation influences greatly the behaviour of the agents. I adapted a spiking actor-critic architecture [Frémaux et al., 2013] to a novel actor-critic architecture with continuous time and actions, and implemented it in a virtual watermaze task. I have obtained similar performance to that of the spiking agent [Frémaux et al., 2013], without the expensive computations which come with a spiking representation.

Further experiments on the continuous actor-critic architecture should involve thorough investigation to obtain reliable performance with diverse levels of noise. In this implementation, I used a first order Euler's scheme with parameters mentioned in Appendix B. Future work could involve testing and comparing other numerical methods for stochastic differential equations to implement the model, for example listed in [Higham, 2001]. Moreover, Vasilaki et al. [2009] implemented different spiking agents in the watermaze task and compared the performance obtained from different reward-modulated learning rules. In this chapter, my continuous actor-critic agent is also a rate equivalent to one of their agents, based on a reward-modulated learning rule with a Hebbian term, and showed similar performance to that obtained in Vasilaki et al. [2009]. Further investigation could compare the performance of the rate agent presented here when changing the learning rules in equations (5.25) and (5.26), for example to rules equivalent to those considered in Vasilaki et al. [2009], adapted to a rate network.

A continuous representation facilitates learning via generalisation, and permits smoother and more precise control, by alleviating the constraints due to the choice of an arbitrary partition of states, actions, and time. The example of the cartpole task [Doya, 2000] discussed in section 5.2.1, illustrated this aspect, in which the discrete actor-critic agent performs very poorly compared to the continuous actor-critic agent. In the example of the watermaze task, the continuous actor-critic has access to the whole continuum of angles, generated from a weighted sum from $N_A = 180$ action cells (equation (5.22)), as opposed to the discrete version in which only one direction among the cardinal directions encoded is selected at every time step. Contrarily to the example of the cartpole task [Doya, 2000], I did not find that the continuous actor-critic learns faster or reaches better performance in navigating to a fixed goal location compared to a discrete actor-critic agent. However, I found that a continuous action representation leads to a smoother trajectory compared to when the action representation is coarser. The fine control observed in behaviour, and especially in spatial navigation, suggests that the neural representation of directions in spatial navigation is continuous.

As continuous actions enable finer control, the latter is only ultimately possible with a temporal resolution that is consistent with the precision of control required. In this chapter, I have discussed the continuity of time. I presented a mathematical framework that does not assume any time discretisation, which is consistent with the observation that behaviours and events can occur at any timescale. I have adapted a continuous actor-critic architecture for navigation in the watermaze. Time continuity enabled us to represent the natural delay in the biological neuronal responses faithfully, as seen in section 5.3. However, my continuous actor-critic approach, as well as spiking actor-critic agent in watermaze tasks [Frémaux et al., 2013, Vasilaki et al., 2009], do not outperform the discrete actor-critic agent presented in chapter 4. As discussed in section 5.2, the advantage of a spiking representation has not been shown yet. Based on experimental results, it could be expected that spiking representations could outperform rate representations when the task requires very fast control in high dimension, or for multi-tasks, as spikes seem to enable fast parallel information processing. This leads to the investigation of how much temporal precision spatial navigation requires.

In general, control occurs at many timescales, and certain spatial navigation situations can critically require the speed enabled by spike coding resolution. I showed that considering information transmission through spikes provides more opportunities for information transmission and is faster than rates. However, model-free learning uses incremental rules which are in essence quite slow. While model-free agents can learn to navigate to a platform using incremental learning rules, they are not flexible to changes in goal location. Therefore, it is unclear whether flexibility in spatial navigation requires a temporal precision that ultimately could only be accounted for using spike representations. Perhaps, learning how to reach a new rewarding location is most of the time not critical enough to require very fast processing through spikes. Avoiding a dangerous situation while driving, or reacting to the sudden appearance of a threat, are potential situations in which the rapidity of spike information transmission could be crucial.

Even though model-free learning mechanisms are slow, using an RL framework seems relevant to address flexibility in spatial navigation. Dopamine, at the core of RL mechanism, is also important for movement vigor [Lau et al., 2017]. Moreover, dopamine influences spike-timing-dependent plasticity mechanisms [Izhikevich, 2007], which shape neural pathways according to the relative spike timing of successive neurons [Sjöström and Gerstner, 2010], suggesting that global reinforcement learning mechanisms can still influence the right synapse at the right time [Izhikevich, 2007].

Spike representations in spatial navigation could contribute to rapid accurate classification of many stimulus-response associations important to account for the infinity of situations that animals face. As Nicola and Clopath [2017] discussed, spike coding increases the dimensionality of the neural code, which could be a crucial aspect of adaptation to a variety of situations at different timescales. Similar situations can require different responses, and similar response pathways can require different speeds depending on the situation. While an animal can safely take time to return to their shelter after foraging, the same action chain mechanisms need to be executed at a much faster timescale in case of the presence of a threat. The state and actions in spatial navigation are never the same, and the timing at which actions should be selected in a state can be critically short (avoiding a threat) or quite long (planning during a hike). Fast spike processing perhaps enables rapid classification of the current situation and selection of the relevant response pathways, that could have been previously shaped through slower, dopaminergic, reinforcement mechanisms with experience. As seen in section 5.2, the temporal precision of spikes through theta phase precession is hypothesised to help the formation of a precise place code and of temporal coherence within hippocampal networks [Skaggs et al., 1996]. Therefore, one-shot learning in the watermaze, even though observed at a longer timescale than milliseconds, could perhaps be facilitated by fast information transmission available through spikes, in particular helping the planning of trajectories [Van Der Meer and Redish, 2011], and potentially involves pre-existing generalisable goal representations that have been shaped through slower mechanisms with exposure to spatial navigation [McKenzie et al., 2013].

Chapter 6

A hierarchical agent for flexibility in a watermaze place-learning task

6.1 Introduction

At the start of the 20th century, Fred W. Taylor, a foreman in a steel production company, published in a book entitled "The Principles of Scientific Management" the results of his experimentation on efficiency of production that he had been operating since 1880 [Taylor, 1919]. Taylor, obsessed with efficiency, realised that by separating tasks into subtasks, and by simplifying maximally the subtasks that every worker had under their responsibility, the global rate of production would largely increase. *Taylorism* refers to the efficient division of labour in industry. Simultaneously, Henry Ford, the founder of Ford Motor Company, designed in a car factory a system that would improve the efficiency of car production: workers were arranged in lines, and given one simple, repetitive task, such as rotating a bolt. This production architecture is still active in current industries and is referred to as *Fordism*. Crucially, the division of labour permits the same factory to realise different aims, which opens up many new production opportunities. These parallel developments of scientific management marked a shift in work organisation that pushed the efficiency and capacity of systems of productions [Witzel and Warner, 2015]. Hierarchical organisation of production is one example of a hierarchical solution for efficiency and increased productivity. This chapter explains that hierarchisation of control is efficient and biologically realistic, and I present a new hierarchical model for flexibility in spatial navigation. The model combines different actor-critic networks as presented in chapter 4 using mechanisms of meta-tuning of the parameters, inspired by Doya [2002], defined according to a goal prediction error of my own devising.

In a spatial navigation context, flexibility is crucial, and as situations are diverse and resources can be limited, efficiency goes hand in hand with flexibility. The actor-critic approach described in chapter 4 requires many trials to adjust to changes in goal locations (see figure 2.11d). The lack of flexibility of the actor-critic agent resides in the fact that the connection weights between the place cell network and the actor and critic networks (figure 4.1), which define the policy and value function, have to be fully relearned to adjust to a new goal location. This process is additionally slowed down by the interaction with the first goal location.

The flexibility shown by animals, in spatial navigation but also more generally in behavioural sciences, indicates that animals and humans use sophisticated generalisation methods that enable them to perform well in tasks in which the rules change regularly. In spatial navigation, the discovery of goal place cells suggests that rodents' brains are capable of generalisation as they have a similar representation that encompasses all goal locations [Hok et al., 2007, McKenzie et al., 2013]. The generalisation of goal locations within an abstract category in which all goal locations could fit is consistent with existing ideas in AI literature that use hierarchical control, which generalises over action sets, to generate more flexible behaviours [Bouchacourt et al., 2019, Botvinick et al., 2009].

Hierarchical Reinforcement Learning is an extension of RL that investigates how a division of tasks into subtasks leads agents to be more flexible [Al-Emran, 2015]. Many spatial navigation tasks can be subdivided into smaller tasks, which are transferable between situations: the motor patterns necessary to reach a goal location can be transferred to reach another goal location. Therefore, hierarchisation of control in spatial navigation seems necessary for the flexibility required in every day navigation behaviours.

Additionally, hierarchisation of control comes with the possibility to adjust the parameters which regulate the production of the behaviours according to information coming from higher levels of the hierarchy. For example, if a good trajectory has been found to reach a location after having tried other possibilities in a well-known environment, it can be efficient to stick to it and stop searching for it. However, if the environment changes occasionally, for example, if new routes are added, it can be good to explore from time to time, to verify whether the current trajectory is still the best one. This process is known as meta-learning [Schweighofer and Doya, 2003]. Section 4.3.1 described how the careful choice of parameters determines the performance of an RL agent. In meta-RL, parameters can be dynamically tuned in an adaptive manner [Schweighofer and Doya, 2003].

In this chapter, I present a novel hierarchical approach for solving the watermaze DMP task. The agent can select a strategy that corresponds to a particular goal according to information from the higher level of the hierarchy and can adjust its parameters according to meta-computations. This hierarchical approach separates trajectory control via a temporal-difference error from goal selection via a goal prediction error and may account for flexible, trial-specific, navigation to familiar goal locations, as required in certain arm-maze place memory tasks. However, it does not capture one-trial learning of new goal locations, as observed in open-fields, including watermaze and virtual, DMP tasks.

I first present the general framework of hierarchical reinforcement learning (HRL) and meta-learning to introduce my model and explain how my approach compares and integrates to the frameworks of HRL and meta-RL. The hierarchical agent adjusts to changing goal locations. I discuss its biological implementation and realism to account for one trial place learning in the watermaze DMP task and other related spatial navigation tasks.

6.2 Hierarchisation of control

In this section, I show that HRL enables fast learning and is biologically realistic. I introduce a hierarchical representation of the watermaze DMP task. I show that a hierarchical representation along with meta-computations enable an agent to adjust its current behaviour as a trial progresses. In particular, the degree of exploration of the agent should be adjusted depending on the stationarity of the environment. I then define a confidence level that evaluates the suitability of the policy for the current goal location, which modification of the degree of exploration of the agent depending on the current performance.

6.2.1 Hierarchisation enables generalisation over action chains

Hierarchisation enables one to decompose tasks into transferable subtasks. The lack of flexibility of TD methods is part of a wider problem in RL known as the problem of scaling [Botvinick et al., 2009]: in general, RL approaches do not deal well with large state and action space. Learning how to map a state space to a value and/or an action by having to perform every action in every state multiple times can be computationally infeasible when the number of actions and states increases and is intuitively inefficient. Many tasks in real life and artificial set-ups can be cast in hierarchical terms. To cope with the scaling problem, humans seem to have naturally dissected tasks into hierarchies, *e.g.*, baking a cake (finding a recipe, shopping for the ingredients, mixing them step by step, baking), or writing a PhD thesis (write the abstract, the title of each chapter, create bullet points for each chapter, create figure captions).

Hierarchical organisation of tasks has also been used in AI as it enables agents to learn faster, and is biologically realistic. HRL is an extension of RL that considers a hierarchical representation of tasks. The diverse approaches within HRL include feudal reinforcement learning [Dayan and Hinton, 1993], in which “managers” or “sub-managers” define subgoals for their workers from which they give reward or punishment, and themselves see their subgoals defined by their own hierarchy and receive rewards or punishments depending on the accomplishment of them. Dayan and Hinton [1993] show that such a hierarchical agent is faster at learning to navigate to a goal location. Another influential approach involves using temporal abstraction, in which the basic RL framework is extended to include temporally extended abstract actions [Barto and Mahadevan, 2003], for an example related to spatial navigation, an extended abstract action could be “go to the door”. The methods applied in AI have been adapted to investigate how brains organise task control [Botvinick et al., 2009], showing that hierarchical models fit very well with experimental data [Bouchacourt et al., 2019].

In this section, I provide a brief description of RL extensions that have been proposed to address the problem of scaling, both in the state and action space. I discuss what could be useful hierarchies to consider in spatial navigation, and I propose a hierarchical representation of the DMP task in the Morris watermaze.

6.2.1.1 HRL as a solution to the scaling problem

The scaling problem occurs for all important features of an RL model: the state space, when discretised, as originally implemented in RL algorithms [Sutton and Barto, 2018], can become extremely large if one wants a representation of the space that appears continuous. Equally, the action space scales up if one wants a smoother control of the behaviour. TD methods use experience to learn value and policy estimates. Using a discrete representation, learning requires going through every state of the network and taking every action, which in practice is unfeasible when the number of possible states and actions is too large. Thankfully, continuous representations enable generalisation.

In the state space, generalisation is possible by using overlapping representations, such that an experience in a certain state can be used to inform the value and policy in neighbouring or similar states. Figure 6.1 presents different solution to generalisation in the state space. For example, coarse coding uses a regular overlapping feature coverage of the state space. The generalisation from a state to another depends on the number of features whose receptive fields overlap. For example, in figure 6.1a, the states s and s' have one overlapping feature. In the case of coarse coding, the representation is not continuous over the space: features have finite receptive fields. Generalisation using

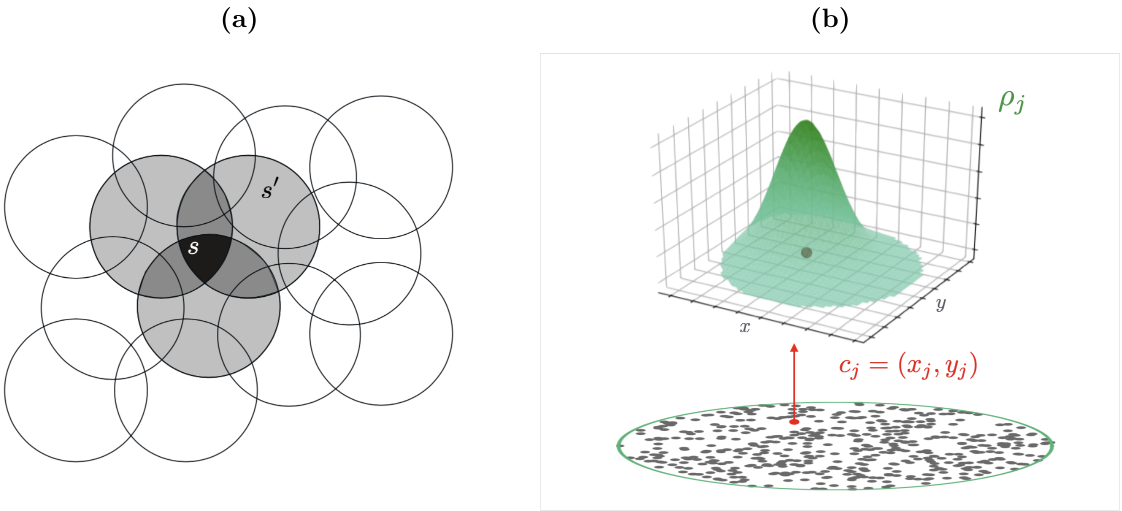


Figure 6.1: Solutions for generalisation in the state space. (a) Coarse coding represents similar states according to the number of features (here circular feature marked as circles) whose receptive fields overlap. For example, here, s and s' have one overlapping feature. Figure adapted from Sutton and Barto [2018] (b) Radial basis functions represent the space using overlapping functions which have Gaussian activity profile, for example, the place cell network used in chapter 4 whose activity ρ_j is given by equation (4.1) gives rise to a Gaussian activity profile, and whose centres c_j are uniformly scattered around the maze. Using this network, every location within the maze has a unique encoding, and the representation is continuous, which is advantageous for differentiation.

a continuous representation is possible using radial basis functions (see figure 6.1b) [Sutton and Barto, 2018]. Section 4.3.1 explained that generalisation in the actor-critic architecture can be done through the use of units mimicking place cells. Using units with Gaussian activity profiles (equation (4.1)), and whose centres are uniformly scattered around the space, is an example of representation using radial basis functions. In particular, I described how the degree of generalisation can be tuned by the width of the Gaussian in section 4.3.1.1.

In the action space, generalisation is also possible. For example, in chapter 5, I have used an action space within a recurrent network. The recurrent connections (equation (5.12)) between action cells enable generalisation across actions that code for similar directions. In that case, all the actions are of similar scale: choosing a direction involves a small step in that direction. In the examples in which similar chains of actions could be used for different purposes, this method is limited, as the actor network is tuned to reach one goal location only.

In HRL, the RL framework is extended to enable generalisation of the action space at abstract timescales and generate flexibility. HRL originated from the concept of temporal abstraction [Sutton et al., 1999]. In temporal abstraction, actions are decomposed into sets of sub-routines that can be flexibly used and adjusted according to the task requirement. Instead of having only the usual primitive actions, such as going North, East, etc., in a spatial navigation context, and in the other cardinal directions, as considered in section 4.2, abstract actions can specify a whole policy by themselves. One widely used example to illustrate the concept is the action "making a hot drink": the generalisation "making a hot drink" can apply to making a tea or a coffee or any other hot drink. Moreover, it can be divided into smaller steps, such as boiling the water or grasping a spoon to add sugar. These subroutines of shorter duration that are part of the abstract action "making a hot drink" enable generalisation: if you know how to add sugar to your tea, you can also add sugar to your coffee. This example illustrates how hierarchisation and abstraction of actions and task representation lead to more flexibility: if I know how to make a coffee, I can learn much faster how to make a tea. In the following section, I provide a quantifiable example of HRL in spatial navigation.

6.2.1.2 The options framework

In this section, I present the options framework and provide an example of its efficiency in spatial navigation. The options framework is a particular implementation of HRL [Sutton et al., 1999]. In the options framework, the action space, containing the single steps actions traditionally used in RL approaches and in the previous models presented in this thesis (*e.g*, in equation (4.3) in section 4.2), is augmented to incorporate temporally extended actions that are referred to as "options". An option is defined by an initiation set that contains the states in which the option can be selected, a termination set that consists of the states in which the option terminates, and an option-specific policy, similar to the policy defining the probability of selecting actions used in chapters 4 and 5 (equations (4.3) and (5.22)), but conditional on the selection of the option, and that can also contain other options. When the option terminates, it leads to the computation of an option error, which is defined as the difference between the value of the state where the option terminated and the value of the state where the option was initiated, plus the eventual rewards that have been accumulated during the execution of the option, referred to as pseudo-rewards. It is therefore similar to the error (3.5); with the only difference that it can compare the value of distant states that can be reached through the selection of an option. Figure 6.2a illustrates the framework: at time 2, the agent selects an option that remains active until timestep 5. During this time, primitive actions are selected using the option's policy, and the option-specific policy and value function are updated at every timestep based on an option-specific prediction error. The primitive actions are the same as those used in the actor-critic architecture in chapter 4 (*i.e.*, going North, East, and so on). The option-specific prediction error is similar to the reward prediction error (3.5), except that it takes into account pseudo rewards that are given only when this option is selected (the yellow star in figure 6.2a).

Botvinick et al. [2009] proposed an illustrative example of the options framework for spatial navigation by introducing subgoals in a multi-room environment, in which the options agent is much faster to reach a goal than a traditional actor-critic agent. The task is to reach a goal location within a multi-room environment (see figure 6.2b). In every room, two options are possible, each of them leads to one of the doors of the room where a pseudo reward is given. At every state, the agent can choose between a primitive action or an option. Before the main reward (given at the goal location) is given, the options value functions and policies are pretrained, which consists of learning the optimal chain of primitive actions through TD learning using the pseudo reward associated with that option. This generates an almost optimal trajectory to the door corresponding to the selected option. After pretraining, the task starts and the agent is trained to reach the goal location. Figure 6.2c shows the performance of two agents across repeated exposure to the task: an option agent and a traditional actor-critic agent using only primitive actions. The agent using only primitive actions takes much longer to reach good performance than the option agent.

The example of navigating in a multi-room environment illustrates that hierarchical

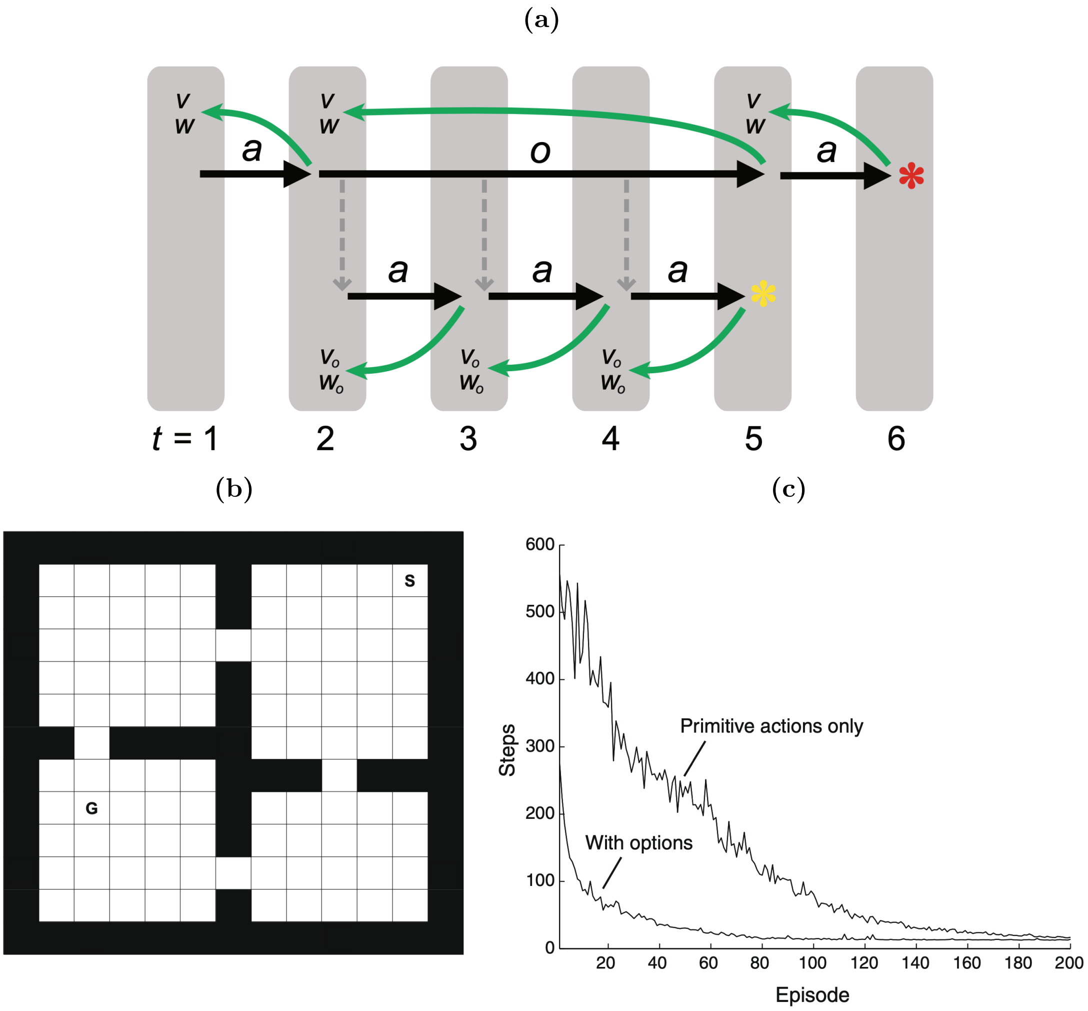


Figure 6.2: a) The options framework schematic: an option (top layer, "o") can be decomposed in sub-actions "a", also called "primitive" actions. At timestep 1, the agent chooses a primitive action, for example, moving North, South, East or West. After the completion of this primitive action, the value (V) and action strength (W) at the previous state are updated (backward arrows from timestep 2 to timestep 1). At timestep 2, the agent selects an option. This option persists until timestep 5, and involves the selection of primitive actions according to the option's policy (bottom layer). The option-value (V_o) and option-policy (W_o) of the states visited during the option are updated at every timestep, and depending on the pseudo reward given at the end of the option (yellow asterisk). At timestep 5, the option terminates and the value of the state it was chosen in, along with the probability of choosing that option in that state, is updated (large backward arrow from timestep 5 to timestep 2). The weight update follows a standard TD error update as described in 3.2.2.1. Afterward, one last primitive action is selected, which leads to the final goal, where a reward is received (red asterisk). The option framework involves a hierarchical representation of a task. In the room example (b) [Botvinick et al., 2009], the agent has to find a goal in a multi-room environment. In every state, the agent can select between primitive actions or two options, each of them terminating at one of the two doors of the room, where a pseudo reward is given. (c) Performance of the hierarchical agent ("with options") in the room problem, compared with the non-hierarchical agent ("Primitive actions only"). In the model proposed by [Botvinick et al., 2009], the agent is pre-trained to associate the correct primitive actions for each option. Once the goal is presented, the agent only needs to learn to select the correct options to reach the appropriate room, and to select the correct actions for the last few steps to the goal, therefore reducing the action space necessary to reach the goal. As a result, the option agent is much faster than the primitive agent to learn to reach the goal. Figure adapted from Botvinick et al. [2009] with permission.

organisation of action space leads to less training time to reach optimal performance. This provides a clear illustration of how hierarchies can improve efficiency through generalisation. In this example, the hierarchies of the task are already known by the agent, which is pre-trained to reach the sub-goals. In reality, humans and animals have to find efficient sub-goals by themselves. The doors of the room, which are the termination sets of the options, are locations common to many trajectories: any trajectory that starts in one room and ends in the other passes through the door. These are referred to as bottleneck states in a graph and are important to identify for efficient navigation. Bottleneck states are natural subgoals common to many goal locations. In the next section, I discuss experimental results suggesting that animals are efficient at finding the right hierarchies in spatial navigation.

6.2.1.3 Hierarchies in spatial navigation and in neural recordings

In this section, I show that animals and humans naturally extract hierarchies of situations and that this can also be seen in neural activities.

Animals and humans are very efficient at identifying bottleneck states. For example, going through the door of our home, passing by the end of our street, or reaching our car, are all common subgoals to many trajectories. Experiments suggest that humans navigating in an abstract graph, in which states correspond to visual stimuli, can infer the structure of the graph simply from presentations of sequences of neighbouring stimuli within this graph [Garvert et al., 2017]. Recent research suggests that humans' abstract reasoning capabilities rely on similar mechanisms to the one provided by the cognitive map, which provides a very efficient organisation of knowledge [Mark et al., 2020, Behrens et al., 2018].

In spatial navigation, hippocampal activities during a task evolve to efficiently represent the relevant dimensions of the task. McKenzie et al. [2014] studied spatial associative learning in rodents in four different dimensions: object, location of the object, global context, and reward associated with the task. They associated different objects at different positions in different contexts to different rewards so that the rule of the task (i.e., which object rats should optimally pursue) depends on the context. They recorded dorsal hippocampal CA1 and CA3 activities using 'hyperdrive' (a set-up that enables multi-site recordings on the rats). They found that most of the correlation of the activities of the targeted cells was explained by the context: similar contexts generated correlated activity profiles. Events that occurred in opposing contexts showed anti-correlated patterns of activity. They additionally showed that they could find a successive hierarchy within neural representations that was consistent with the task. The second separatory factor in terms of correlation of activity was the position of the object: within the same context, events in which the objects are in a similar position generated correlated activity profiles, objects in different positions generated anti-correlated activity profiles. In other terms, the overlap in the ensemble pattern decreases between events with different rewards or objects and in similar contexts and positions, events with different positions and in similar contexts, events with different positions and in different contexts. Their results show that representations of events within the hippocampus are hierarchical, and that the hierarchy defining the activity of neuronal ensembles is task-dependent. Indeed, the particular organisation of representations for this task seems efficient: in this particular example, the context defines which object is rewarded or not, therefore defines the rule. Thus, identifying the context as a discriminatory factor leads to better performance at the task [McKenzie et al., 2014].

This example suggests that efficient hierarchies for a spatial navigation task can be extracted from neural representations. Hence, considering a hierarchical organisation of the watermaze task seems a biologically realistic approach to enable the agent more flexibility towards changes in goal locations on the DMP task. In the next section, I present a hierarchical organisation of the watermaze DMP task.

6.2.1.4 Hierarchical representation of the watermaze task

The flexibility shown by rats in the watermaze DMP task suggests a hierarchical control. Rats tend to navigate to the previous goal location on trial 1 with a new goal location (Steele and Morris [1999], Pearce et al. [1998], and the unpublished observations from experiments run in Dr. Bast laboratory), and then find out that this remembered goal location is not the current goal. On the following trial, they take a direct trajectory to the new goal location (figure 2.13a) [Bast et al., 2009]. Linking back to the options framework, this flexibility could be achieved by defining temporally extended actions whose termination sets are defined by goal locations. When a new goal location is encountered, it can trigger the implementation of decisions of shorter timescales which allows the realisation of the movement to reach the new goal location. In the actor-critic architectures presented in chapter 4 and 5, the critic controls the selection of the direction at every timestep, finely chosen to mimic the generation of a smooth trajectory. In the terms used in section 6.2.1.2 related to the options framework, the critic and actor would define an option-policy and an option-critic over primitive actions, in which the option would correspond to reaching one of the goals.

To enable the agent to be more flexible towards changing goal locations, I thus take inspiration from the options framework and propose a hierarchical model to solve the watermaze DMP task. The hierarchical representation contains of a set of goals, and the agent is trained to reach each of these goals using the actor-critic model presented in chapter 4. Figure 6.3 illustrates the hierarchy: each different, pre-set, goal location j is associated with the actor and critic network with the connection weights (W^j) and (Z^j). I call the set of actor and critic weights that define the policy and value function corresponding to a particular goal location a strategy. The policy linked to a strategy sits at an intermediary level of control: it controls the choice of the direction at each timestep, it does not control the choice or retrieval process of the goal that is being pursued. I have added a level of control that determines the goal to be pursued, defined by the selection of a strategy. Initially, a strategy j is selected at random. From this point, the agent then follows the policy determined by the weights (W_t^j) and receives feedback on its actions based on the critic weights (Z_t^j).

At every timestep, the agent monitors the performance of its current strategy via computation of a “goal prediction error”:

$$\delta_G = \delta_t - \delta_t^j. \tag{6.1}$$

The goal prediction error is the comparison between the observed error δ_t and the predicted error δ_t^j. The observed error δ_t refers to the error computed from the current critic as in equation (3.5)

$$\delta_t = r_t + \gamma C(p_{t+1}, W_t^j) - C(p_t, W_t^j). \tag{6.2}$$

Here, $C(p_t, W_t^j)$ refers to the critic activity (4.2), and the reward term r_t refers to the reward received or currently given by the environment (*i.e.*, $r_t = 1$ at the current goal location). In equation (6.1), the predicted error δ_t^j is computed from the strategy j:

$$\delta_t^j = r_t^j + \gamma C(p_{t+1}, W_t^j) - C(p_t, W_t^j), \tag{6.3}$$

where r_t^j refers to the reward currently predicted by the strategy j (*i.e.*, $\delta_t^j = 1$ at the goal location j), and $C(p_t, W_t^j)$ refers to the critic activity associated with strategy j. In the next section, I present how the goal prediction error is used to shape the behaviour of the agent as the trial progresses.

6.2.2 Shaping the exploration level according to the confidence of the agent in its current strategy

In this section, I show that meta-learning enables the dynamic adjustment of the ongoing behaviour according to the current performance, and I describe that animals and humans tend to adjust their degree of exploration depending on the degree of uncertainty of a task. I then propose that the degree of exploration of the agent in the watermaze could depend on the confidence in their current strategy.

In section 4.4, I presented an improvement to the actor-critic architecture in which an estimated goal location is stored and compared to an estimated location of the agent to form a goal-directed displacement. This enables flexibility to changing goal

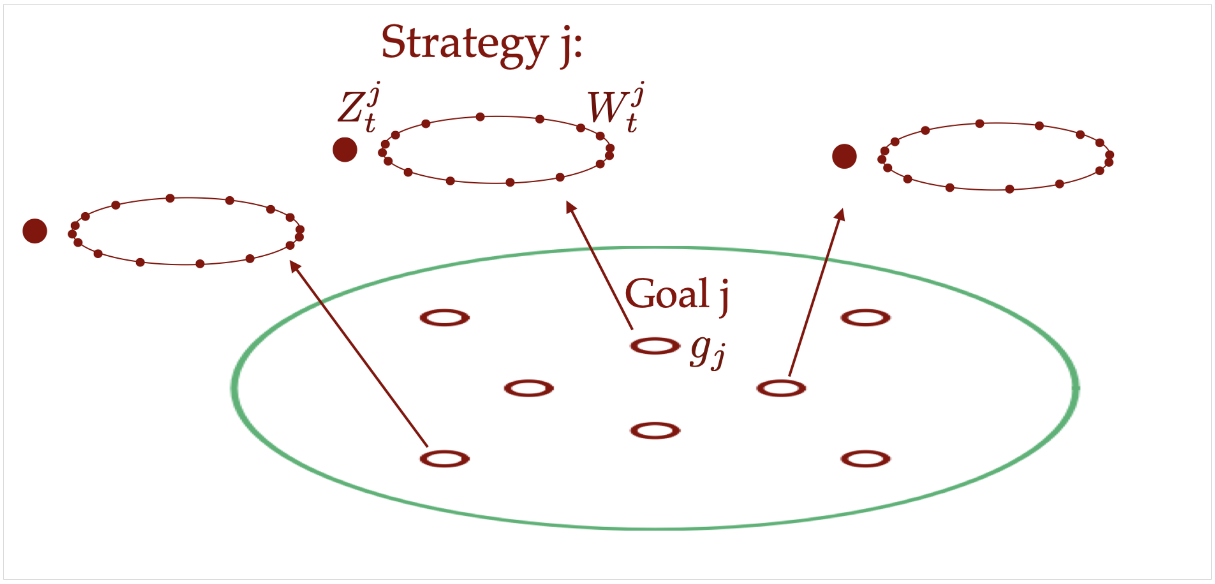


Figure 6.3: Schematic of the hierarchisation of the DMP task: every goal g_j, $j \in 1 \cdots N_G$, is associated with critic weights Z^j and actor weights W^j which have been trained using the actor-critic algorithm presented in chapter 4. The set of actor and critic weights associated with a particular goal is called a “strategy”. Similar to options in the options framework, a strategy can start at any of the start locations and terminate at any of the goal locations, in which it can either be pursued further if it has been successful, or it can be replaced by another strategy. Contrary to an option, the selection of a strategy is not trained through TD learning mechanisms as primitive actions are but is selected according to a goal prediction error (6.1).

locations, but, as the coordinate action is not available during the first trial, instead, a random exploration is enforced (see section 4.4.4), the model does not describe how the policy adjusts to an incorrect prediction of the goal location. To successfully escape the watermaze on the first trial to a new goal location in the DMP task, rats need to adjust their behaviour to this error in prediction and start exploring the maze in order to discover the current platform location. This suggests that rats have a way to adjust their ongoing policy to unpredicted changes as the trial progresses. This can be effected by adjusting parameters that shape the exploration/exploitation trade-off. In the following section, I briefly review the important parameters of an actor-critic architecture. I then introduce meta-learning, which enables online adjustment of these parameters.

6.2.2.1 Important parameters in an actor-critic architecture

All algorithms, including the actor-critic architecture, discussed in chapter 4, rely on a careful choice of parameter values to guarantee good performance. In addition to the choice of representation of the states, discussed in section 4.3.1, the important parameters in a classic actor-critic architecture are the discount factor (discussed in section 4.3.1.1), the learning rates (α in equation (3.7) refers to a general learning rate, χ_C in equation (4.8) and χ_A in equation (4.10) refers to the rate at which the value function and policy estimates evolve from experience), and the inverse temperature β (in equation (4.4)).

The learning rate shapes the speed at which the system adapts to experience: with a low learning rate, a single experience, characterised by a single update using equation (3.7), has little impact on the global estimate of the value function or the policy, computed over many steps or even trials (see section 4.3.1). The agent takes longer to train, but the behaviour is more steady (*i.e*, subject to slower fluctuations) and robust (*i.e*, less sensitive to outlier experience) than with a high learning rate. With a high learning rate, the system adjusts very quickly to any change in feedback. In a volatile environment, in which for example the best option might change quite often, it is useful to have a higher learning rate to adapt quickly to changes. However, in less volatile environments, a lower learning rate makes the behaviour more robust to outliers. Experiments show that animals and humans can adapt their learning rates to the volatility of the environment [Behrens et al., 2007, Constantinople et al., 2019].

The inverse temperature β regulates the degree of exploration of the agent. In equation (4.4), one can see that β scales the difference in probabilities from the activities of the action cells: $\beta > 1$ sharpens the selection by stretching the difference in activities, $\beta < 1$ decreases them. Figure 6.4 shows the behaviours of the agent trained using the model in section 4.2 for different values of β. For low values of β, one can see that the mean probability over the actions (computed by taking the mean of the probability given in equation (4.4), over the course of the trial for every action cell) within the trial is almost uniform across the action space, leading the agent to be highly exploratory, as can be seen in the random trajectory (figure 6.4a). For increasing values of β, the mean of the probability over the actions becomes progressively sharper, and the generated trajectories become more direct to the goal (figures 6.4b and 6.4c). And for high values of β, the mean probability distribution throughout the trial is concentrated on the best action, and the trajectory is direct to the goal (figure 6.4d). This regime is known as exploitation.

When the environment is stable and the policy has been learned, it is more efficient to have a low degree of exploration and stick to the learned policy. However, in highly volatile or unknown environments, a certain degree of exploration can be useful, either to discover the best actions in this environment or to adjust to changes. Therefore, within the execution of a task, it is efficient to be able to regulate and adjust the degree of exploration on the fly, as changes are being observed. In the following section, I present meta-learning, which enables an agent to adjust parameters as the trial progresses.

6.2.2.2 Meta-learning: adapting the parameters to the degree of stationarity of the task

Meta-learning, which enables the dynamic regulation of parameters, allows the agent to adapt to the non-stationary nature of tasks [Doya, 2002]. One of the first simple implementations of the meta-learning algorithm has been used for an agent to adjust to the volatility of a bandit task [Schweighofer and Doya, 2003]. A bandit task requires the agent to choose between different actions associated with different probability distributions of rewards. Schweighofer and Doya [2003] use the difference between long term rewards and instantaneous rewards to update the model parameters, and showed that their agent could adjust both its learning rate and exploration parameter according to the current requirement of the task. Successful meta-learning requires agents to be able to track information at multiple timescales [Doya, 2002, Schweighofer and Doya, 2003, Wang et al., 2018], and experiments show that animals are able to learn optimal strategies in tasks that require learning over multiple timescales [Iigaya et al., 2019].

Experimental results show that humans and animals adjust their exploration rate according to the stability of the task. In a bandit task, humans and animals tend to be more exploratory when the optimal strategy is less clear [Swanson et al., 2020]. Figure

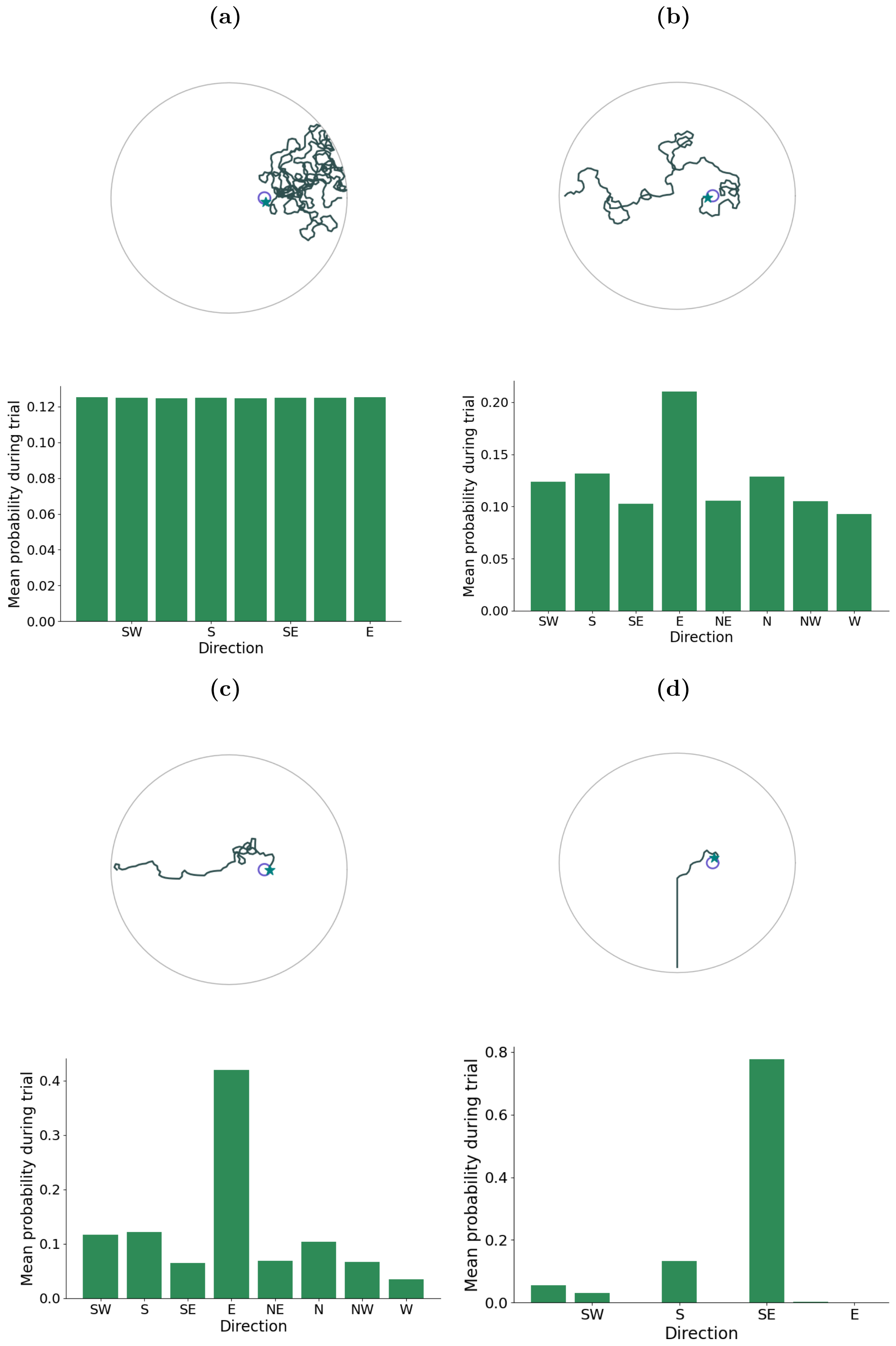


Figure 6.4: Effect of the inverse temperature parameter β on the trajectory (top) and on the mean policy (bottom, histogram plots). The trajectories were generated from implementing the trajectory defined by the policy given in equation (4.4). The histogram plots were computed by computing the mean of the probability of every action cell given in equation (4.4) over the whole course of the trial. a) For a very low value (here, $\beta = 0.01$), the behaviour is extremely exploratory: the trajectory is random and the mean probability distribution across the trial is uniform over the possible directions. b) and c) as the inverse temperature increases ($\beta = 0.2$ and $\beta = 0.6$), the behaviour becomes less random, and d) for very high values of inverse temperature ($\beta = 2$), the trajectory is direct to the goal and the mean probability distribution is concentrated around the best direction to choose.

6.5 shows the inverse temperature fitted to rats' choices on a two armed bandit task, in which rats learn to choose between one highly - and one less - rewarded arm. In the experiment, Swanson et al. [2020] changed the ratio between the reward probabilities of the two arms. Figure 6.5 shows the degrees of exploration of the animals in different stages of the task associated with different ratios. When one is rewarded 100% of the time and the other never, the degree of exploration is lower compared to stages in which the two options are less drastically different, for example, when the best arm is rewarded 70% of the time and the other 30%. In visual discrimination tasks, in which animals have to determine whether dots are moving right or left on a screen, sensory stimuli at the boundary involve response probabilities at random [Gold and Ding, 2013]. In both visual discrimination tasks and the example of the bandit arm, the experimenter can tune the level of confidence that the participants have in their decision by changing either the contrast or strength of the stimuli in the case of the visual discrimination task or by changing the ratio between reward probabilities in the bandit task. In both cases, this controllable manipulation leads to the observation of meta-learning adaptation in animals which can adjust their exploration parameter to the level of uncertainty in the task.

In the following section, I propose that the agent could adjust its degree of exploration according to its confidence in its current policy.

6.2.2.3 The degree of exploration shaped by the confidence in the current strategy

I extend the actor-critic architecture from chapter 4 by adding an online regulation of the inverse temperature β depending on the confidence that the agent has in its current strategy. I propose that the goal-prediction error (6.1) can be used as an input to compute a confidence level σ_t of the agent in its current strategy. The confidence level evolves according to:

$$\frac{\mathrm{d}\sigma_t}{\mathrm{d}t} = -\frac{\sigma_t}{\tau_\sigma} + \mathrm{D}\delta_G, \tag{6.4}$$

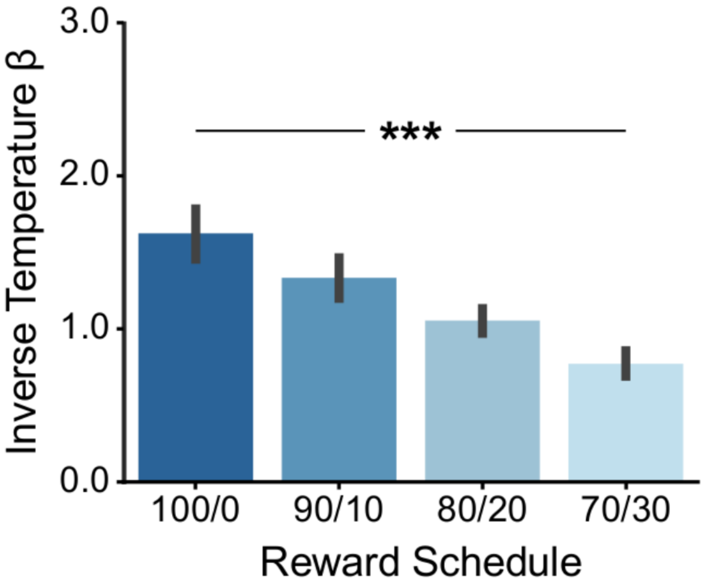


Figure 6.5: Adaptation of the inverse temperature β as a function of the reward difference in a 2 arms bandit task: when the difference in rewards probabilities between the two arms is high, for example when one arm is always rewarded and another never (100/0), the inverse temperature β is high, showing that rats behaviour mostly sticks to the same choice. β reduces when the reward difference reduces, even though the best option remains the same, showing that rats adapt their behaviour to the uncertainty of the situation. Adapted from Swanson et al. [2020].

where δ_G refers to the goal prediction error (6.1). In equation (6.4), D = 0 if a negative goal prediction error has already been received as an input during the execution of a trial, and D = 1 otherwise. D reflects the fact that, once the agent starts mistrusting its strategy, bumping into the goal predicted by its current strategy should not induce any more changes in the confidence level. The goal prediction error input captures the fact that a negative goal prediction error should reduce the confidence of the agent in its strategy, and a positive prediction error should lead to an increase in confidence. The temporal evolution of the confidence, regulated by the parameter τ_σ, is set quite high, so that the confidence level evolves quite slowly after a goal prediction error.

The confidence level influences how the agent's actual behaviour respects the defined policy. The temperature parameter β used in equation (4.4) is defined as a function of the confidence level according to:

$$\beta_t = f(\sigma_t), \tag{6.5}$$

where f is a sigmoid function given by:

$$f(\sigma) = \frac{\rho}{1 + \exp\left(-\omega(\sigma - h)\right)}. \tag{6.6}$$

Here, ρ defines the gain factor, h the inflection value (point of highest slope in the sigmoid). The parameter ω defines the steepness of the function around the cut threshold, which shapes how fast the exploitation parameter evolves when the confidence parameter becomes higher or lower than the cut threshold. Figure 6.6 shows the inverse temperature as a function of the confidence. When the confidence level is very low, the inverse temperature is close to 0, and when the confidence level increases, the inverse temperature reaches ρ.

In the following section, I present the full hierarchical model and its performance in the watermaze DMP task.

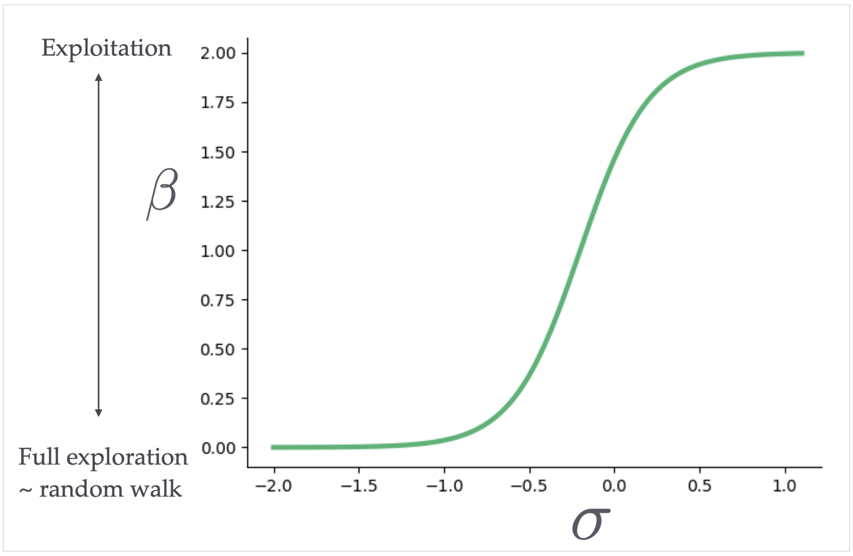


Figure 6.6: Influence of the confidence parameter σ on the exploration/exploitation parameter β. The figure was generated from equation (6.6), with $\rho = 2$, $w = 8$ and $h = -0.2$. The maximum of the function, ρ, was chosen to guarantee the agent to access both exploitation and exploration modes, from the observations made in figure 6.4. When the confidence level σ of the agent is low, the exploration parameter β is also very low, which enables the agent to explore its environment. Contrarily, a confident agent will exploit more its current policy, characterised by a high value of the exploitation parameter β.

6.3 A biologically-inspired hierarchical model for goal selection in the watermaze

I have defined in section 6.2.1.4 a hierarchical representation of the watermaze DMP task. In section 6.2.2.3, I have shown that in this hierarchical representation, the computation of a goal prediction error (6.1) can lead to the online regulation of the degree of exploration of the agent (6.5) via a confidence parameter (6.4). Figure 6.7 presents a schematic of the novel hierarchical model. I have implemented the hierarchical model using parameters given in appendix C. In this section, I show that the agent can adapt its degree of exploration as the trial progresses and that it can flexibly adjust to changes in goal locations.

6.3.1 The agent adjusts its degree of exploration and adapts to changes in goal location

In this section, I show that the agent can adjust its degree of exploration as the trial progresses and that it can adapt to changes in goal locations.

The degree of exploration and the resulting behaviour of the agent shifts when the agent receives a negative goal prediction error. Figure 6.8 shows an example simulation of an agent on the first and second trial of a fixed goal location. Figure 6.8a displays the confidence level (left axis) and the exploration level (right axis) of the agent. The agent starts following the strategy that is associated with the goal shown as a grey circle in figure 6.8. Its confidence level and exploitation degree are high, therefore the agent takes a direct path to the predicted goal location (figure 6.8b). At this moment, the agent receives a negative goal prediction error that drops its confidence level, and therefore, inverse temperature. The agent then explores randomly its environment (figure 6.8c) until it finds the goal (figure 6.8d). It can then select a strategy that maximises the goal prediction error, therefore leading to an increase in confidence and exploitation. On the subsequent trial, the agent takes a direct path to the goal (figure 6.8e).

The agent flexibly adjusts to changes in goal locations. Figure 6.9 shows the mean

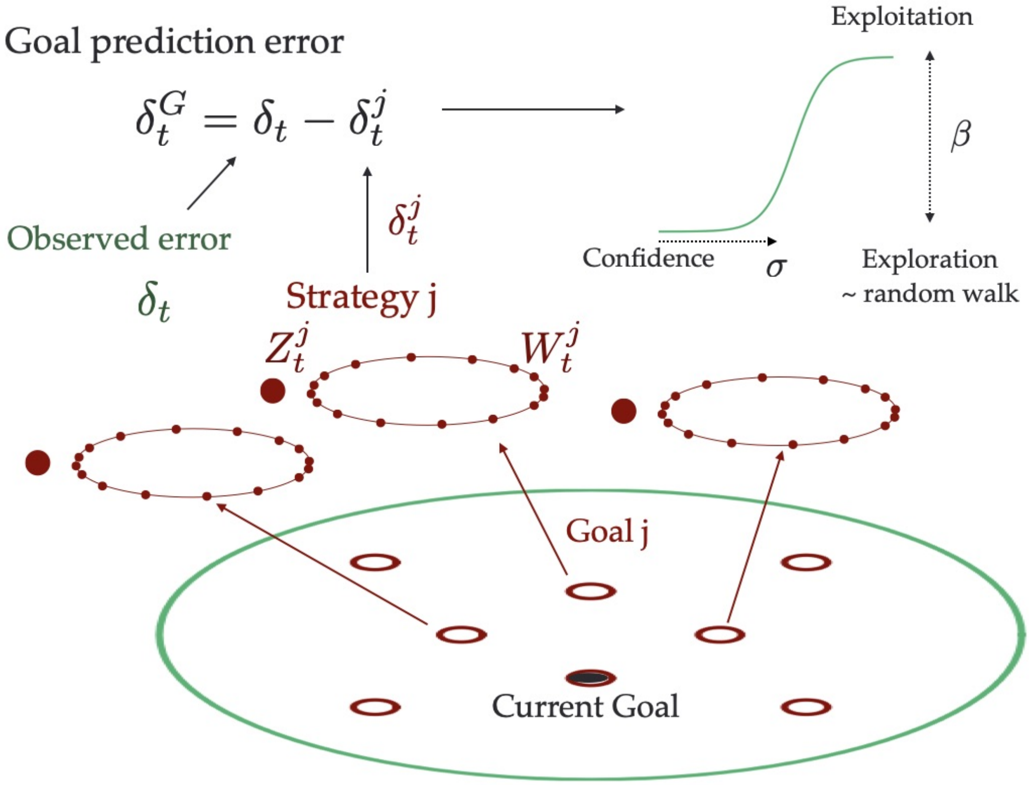


Figure 6.7: Hierarchical RL model.The agent has learned the critic and action connection weights Z_t^j, W_t^j for each goal j from the algorithm in 4, that form the strategy j. A goal prediction error δ_t^G (6.1) is used to compute a confidence parameter σ (6.4), which measures how good the current strategy is in reaching the current goal location. The confidence level shapes the degree of exploitation of the current strategy β through a sigmoid function of confidence (6.5). When the confidence level is very high, the strategy chosen is closely followed, as shown by a high inverse temperature parameter β. On the contrary, a low confidence level leads to more exploration of the environment.

latencies of the agents, computed for 20 independent agents, to different goal locations. Every 4 trials, the goal location is changed. The agent is able to adapt to changing goal locations, as marked by a reduction in latency between trial 1 and 2 of every new goal location. As a result of the confidence parameter having a slow time constant, the latencies tend to increase slightly between the second, third and fourth trial. The flexibility of the hierarchical agent towards changes in goal location was also shown by the coordinate-based navigating agent presented in 4.4. However, while the previous approach did not consider how the agent would adjust when facing an error in goal prediction, the hierarchical model here not only captures the flexibility of rodents from changing goal locations but also the behavioural adaptation that follows a goal prediction error.

In the next section, I relate the architecture of the model to existing results from the literature. I show that the hierarchical agent is partly biologically realistic.

6.3.2 Prefontal areas compute goal prediction errors and confidence level

In this section, I discuss the potential neural basis underlying the computations proposed in the hierarchical model. The behaviour of the hierarchical agent (figure 6.7) partly mimics the behaviour of rats in a watermaze DMP task: it can flexibly adjust to changes in goal location (figure 6.9), and is able to adjust its level of exploration from an error of prediction of the goal (figure 6.6). Moreover, the proposed hierarchy is consistent with existing literature on goal-directed control. I propose that the goal prediction error (6.1) is computed by prefrontal areas, motivated by dopaminergic signals. Prefrontal areas, in particular the Anterior Cingulate Cortex (ACC), are involved in assessing the level of confidence of the agent (6.4). Finally, I discuss the neural basis of exploration/exploitation trade-off. I discuss experimental evidence suggesting that this trade-off could be modulated by neurotransmitters noradrenaline and dopamine.

The hierarchy of control could be held within prefrontal and striatal control loops, with the striatum involved in the implementation of the strategy, and prefrontal areas regulating the strategy and its selection. Prefrontal brain areas have been proposed to carry

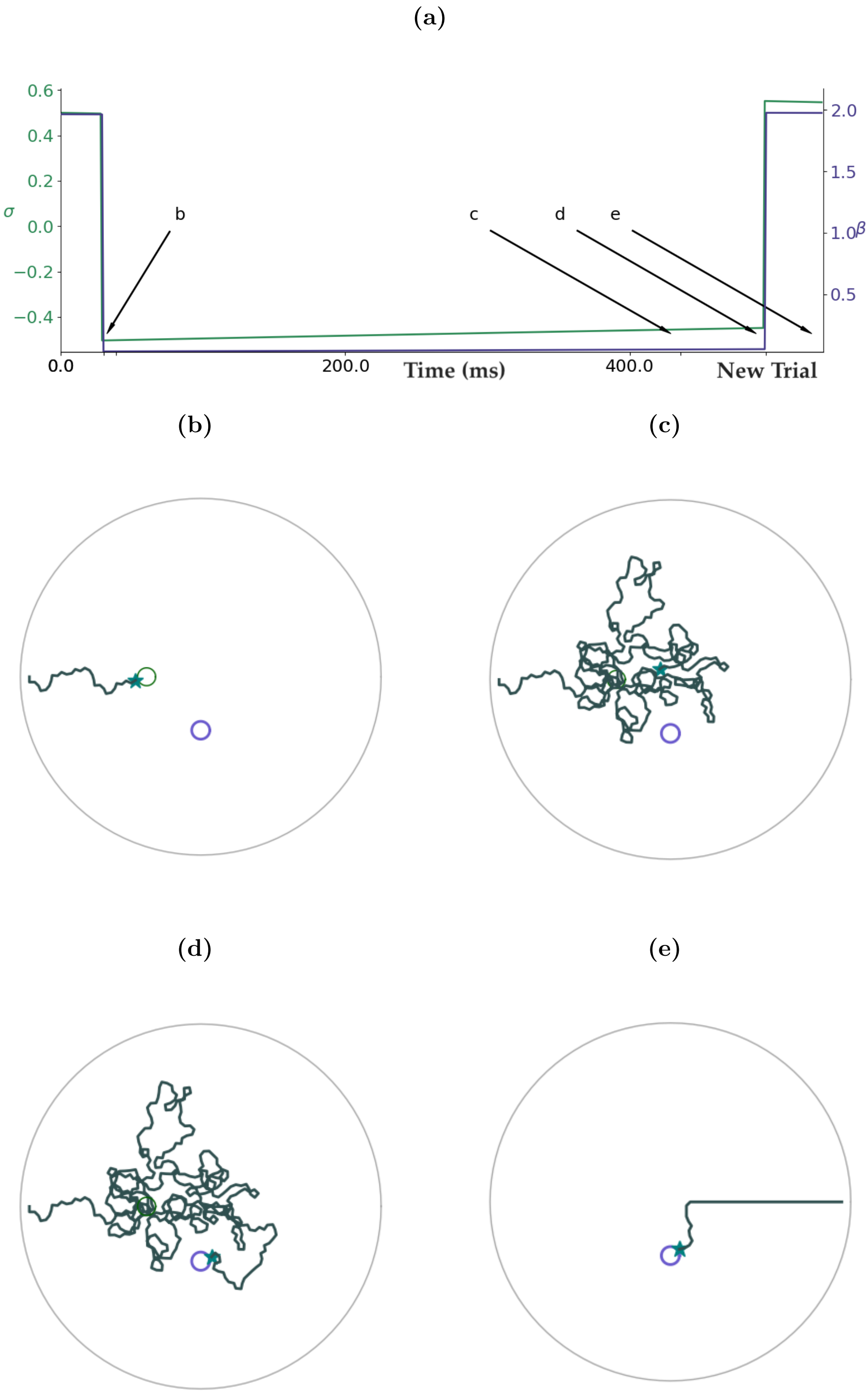


Figure 6.8: Progress of the confidence level σ (left axis) and the inverse temperature parameter β (right axis) (a), and the trajectories of the agent (b-e) during the first and second trial of a new goal location. The agent originally follows the strategy that leads to the goal shown as a green circle using a direct path. When it reaches its original goal, it receives a negative goal prediction error that decreases its level of confidence (panel (b), time point marked in panel (a)), after which it starts exploring its environment ((c), time point marked in panel (a)). This exploration can lead the agent to find the current goal ((d), time point marked in panel (a)) (otherwise, the agent is placed at the goal location, as in real experiments [Bast et al., 2009]), at which point it then receives a positive goal prediction error, leading to a sudden spike in confidence. On the following trial, it can then follow the strategy that best predicts the current goal location and reach it using a direct trajectory ((e), time point marked in panel (a)). This exploration can lead the agent to find the current goal ((d), time point marked in panel (a)).

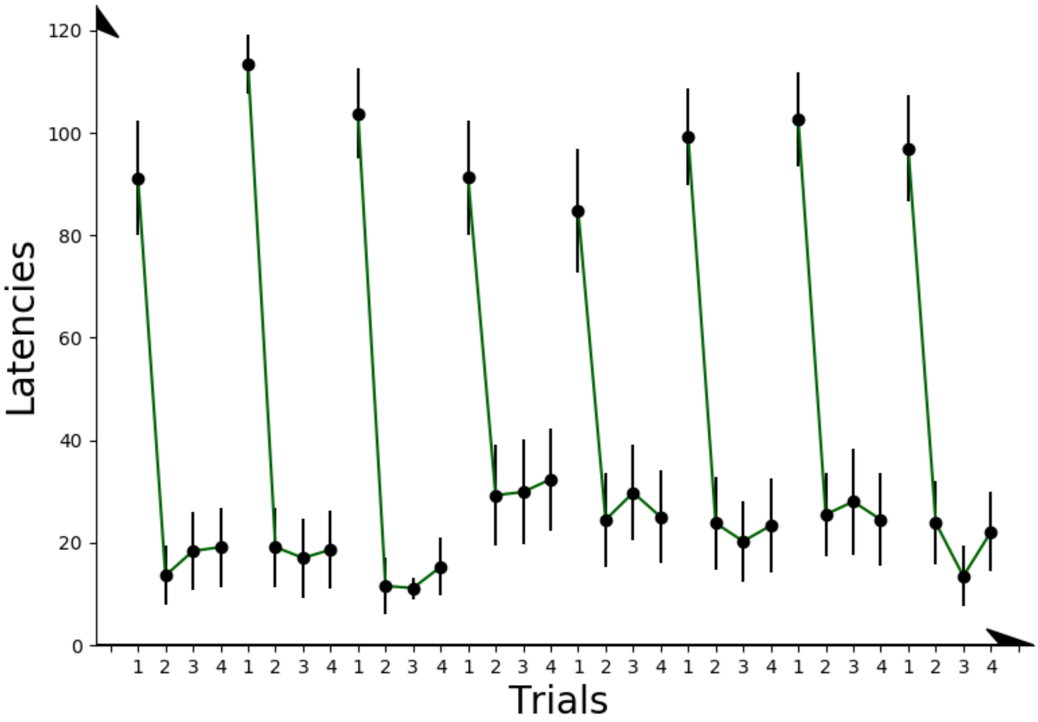


Figure 6.9: Latencies of the agent to reach the goal, for 8 different goal locations. The hierarchical agent is able to adapt to changing goal locations, as seen in the reduction of latencies to reach the goal between the first and second and subsequent trials to new goal locations. For parameters and simulation details see appendix C.

out meta-learning computations, integrating information over multiple trials to perform computations related to a rule or a specific task [Wang et al., 2018], and communicate to other brains areas in order to select an appropriate mapping between inputs, internal states, and outputs to perform the task [Miller and Cohen, 2001]. Neurons in prefrontal areas carry goal information [Poucet and Hok, 2017, Hok et al., 2005]. A synthesis on the neural basis of goal-directed behaviours suggests a hierarchically organised control of the behaviour with the higher levels of control, which regulates the lower level subroutines, mediated via frontal cortex [Rusu and Pennartz, 2020]. In particular, the authors propose that the prefrontal-ventral striatum circuits occupy the highest positions in the hierarchies of loops of control [Rusu and Pennartz, 2020]. In the approach presented here, the selection of the strategy, that I hypothesise could be performed by frontal areas, rules the lower level of control performed by the actor and critic associated with this strategy, which seems to be held by the striatum (section 4.3.2). Therefore, the hierarchy proposed in the hierarchical model (figure 6.7) is consistent with that proposed in Rusu and Pennartz [2020]. The activity of the medial prefrontal cortex (mPFC) of monkeys has also been correlated to adjustment in subsequent action selection following prediction errors [Matsumoto et al., 2007]. Moreover, prefrontal population activity dynamic correlates with the adoption of new behavioural strategies [Maggi et al., 2018]. This is consistent with the hierarchical model, in which the goal prediction error is used for the selection of an adapted strategy.

I propose that prefrontal areas compute the goal prediction (6.1) errors and the confidence level (6.4). Prefrontal areas, including the ACC, seem important for decisions under uncertainty [Rushworth and Behrens, 2008]. The ACC is a frontal area located right before prefrontal areas in humans, and within prefrontal cortex in rodents. An extensive literature exists about the ACC being involved in representation of uncertainty and its link with behavioural flexibility [Khamassi et al., 2013, Behrens et al., 2007, Amiez et al., 2005], which relates to the computation of the confidence level (6.1) in the hierarchical model considered here. In humans, ACC BOLD (Blood Oxygen Level Dependent) responses correlate with the volatility and uncertainty of a task [Behrens et al., 2007].

Oliveira et al. [2007] show that functional MRI responses in the ACC increase when subjects make errors and conclude that the ACC is involved in performance monitoring through computing prediction errors. In primates, ACC neuronal activity is modulated by reward prediction errors [Amiez et al., 2005]. More precisely, the authors show that ACC responses to prediction errors were more important when this prediction error indicated the necessity to shift response. They conclude that ACC activities can reflect goal prediction errors, and additionally signal crucial events that interrupt potentially rewarded actions [Amiez et al., 2005]. The extensive literature on humans and primates, and a recent study on rats showed that ACC is necessary to shift behaviour in a task that requires flexibility [Brockett et al., 2020]. Additionally, Kolling et al. [2016] found that dorsal ACC activities carry information about the history of recent reward integrated simultaneously over multiple time scales. In the model discussed in this chapter, the confidence level (6.4) is modulated by the goal prediction error, and this information is retained over multiple trials thanks to its slow time decay constant. This is consistent with experimental results involving the ACC in integrating goal prediction errors and linked with behavioural flexibility.

The goal prediction error (6.1) could be computed from dopaminergic inputs. Prefrontal dopaminergic activity affects flexibility towards changing rules [Goto and Grace, 2008, Ellwood et al., 2017], and frontal dopamine concentration increases during reversal learning and rule changes [van der Meulen et al., 2007]. In particular, blockade of dopaminergic receptors within the prefrontal cortex in a task in which the reward value changes impairs behavioural flexibility [Winter et al., 2009]. A recent experimental result suggests that communication from the ventral tegmental area (VTA, one of the areas in which the dopamine is produced) to the ACC is stronger when rats committed errors that were followed by behavioural adaptation Elston et al. [2019]. This is consistent with the use of the temporal difference error (6.3) to compute the goal prediction error, based on my previous discussion (section 4.3.2.1) implicating the dopaminergic system in the computation of reward prediction error. Therefore, the computation of the goal prediction errors and the confidence level could be performed by prefrontal areas based on reward prediction errors carried by dopaminergic signal.

These results suggest that prefrontal cortical areas, including the ACC, could be the locus for the computation of the goal prediction error (6.1) and their integration in the confidence parameter given in equation (6.4). Experimental results hint towards the ACC supporting the integration of the goal prediction errors, computed by other prefrontal areas, into the confidence level. Both seem to respond to prediction errors and to be involved in flexibility in spatial navigation, and they both receive dopaminergic projections from the VTA, which fits with the definition of the goal prediction error (6.1). However, it is unclear how roles are distributed between the two. Kolling et al. [2016] suggest that the interaction between ACC and other prefrontal areas leads to behavioural changes.

In the model, the level of confidence shapes the degree of exploration of the animal (β). Early papers on meta-learning suggest that this parameter could be regulated by the neurotransmitter noradrenaline [Doya, 2002, Ishii et al., 2002]. The involvement of noradrenaline in regulating the exploration/exploitation trade-off has also been confirmed in a pharmacological study in humans [Jepma et al., 2010]. However, more recent approaches also suggest that the tonic level of dopamine could shape the level of exploration/exploitation trade-off [Humphries et al., 2012]. The link between higher level of activity in frontal areas in response to prediction errors and the modulation of exploration/exploitation trade-off needs to be investigated further. In the watermaze, rats explore to escape the water, but it is more likely that this exploration is the result of the selection of a default policy than the rats sticking to their original policy that has proven ineffectual. However, my model captures the fact that meta-errors can be used to adjust the behaviour of the agent as the trial progresses.

The hierarchical model presented here is therefore partly biologically realistic, both in terms of behaviours and neural implementation. In the following section, I expose its limitations to account for a biologically realistic approach to navigation in the watermaze DMP task, and discuss other spatial navigation tasks for which the hierarchical model could provide a biologically realistic account of the behaviours shown by the rats.

6.3.3 A plausible model for delayed non-matching to place tasks in radial arm mazes

I have presented the hierarchical approach (figure 6.7) to highlight how separating the computation of the choice of the goal from the computation of the choice of the actions to reach it enables flexibility in spatial navigation. As discussed in section 6.3.2, the hierarchical model and the computations it implements are consistent with the involvement of prefrontal areas and the dopaminergic system in behavioural flexibility. However, the hierarchical model has several features that limit its use to provide a neuropsychologically plausible explanation of the computations underlying DMP performance in the watermaze and related open-field environments. In this section, I discuss the limitations of the hierarchical model to explain the behaviour of the rats in the watermaze DMP task. I also discuss other spatial navigation tasks, in radial arm maze environments, for which the hierarchical model could provide a biologically realistic explanation. I first present the main limitations of this model in addressing the watermaze DMP task.

6.3.3.1 Limitations of the hierarchical model is the watermaze DMP task

First, the requirement of pre-training is not consistent with the experimental procedure of the DMP task. The agent has to learn beforehand the connections between place cells and action and critic cells that lead to successful navigation towards every possible goal location of the maze. This would involve pretraining with the possible goal locations, whereas the agent would fail to learn a completely new goal location within 1 trial (*i.e.*, return to a location that contained the goal for the very first time). Hence, the model can be considered as a model of one-shot recall, rather than one-shot learning. This cannot account for the one-trial place learning performance shown by rats and human participants on DMP tasks towards new goal locations, rather than familiar ones [Bast et al., 2009, Buckley and Bast, 2018]. In the hierarchical RL architecture presented in section 6.3, I consider familiar goal locations (*i.e.*, although the goal location changes every four

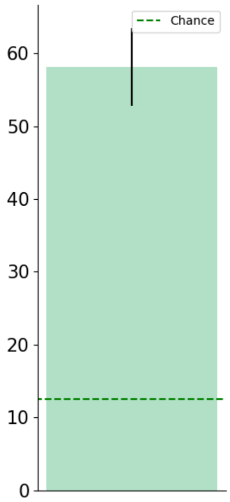


Figure 6.10: Search preference for an area around the supposed goal location on a trial in which the goal has been removed. The search preference has been computed using the same method as described in figures 2.13b and 4.9b: on the second trial, the goal location is not available to the agent. The time that the agent spends navigating in an area centred around the goal location is compared to the time spent navigating in the total of 8 areas of similar surface covering the whole environment (figure 2.13b). A negative goal prediction error induces more exploration (see figure 6.4) similar to a random walk over the 8 possible directions when the value of the inverse temperature β is low (4.4). Figure 6.4a shows the mean probability of action selection over an exploratory trial and a typical trajectory. A random walk is not broadly exploratory: the trajectory tends to loop over itself, which is reflected here by a high search preference in an area centred around the goal location, where the random walk starts.

trials, the goal locations are always chosen from a set of 8 locations where the agent has learned to navigate to the goal during a pretraining period). This contrasts with the most commonly used watermaze DMP procedure where the goal locations are novel (*e.g.*, Steele and Morris [1999], Bast et al. [2009]).

Second, the exploration based on a random walk is not biologically realistic. If the agent does not find the goal where its current strategy leads, it starts exploring the maze until it finds the current goal location and then selects a strategy that estimated its location. On probe trials, removing the goal location leads the agent to start exploring randomly. The agent shows a marked search preference for the area around the goal location (figure 6.10) due to the nature of the exploration. When the exploration parameter $\beta = 0$, the action selection (4.4) is random (figure 6.4a). This causes trajectories to be quite concentrated in space, therefore the search preference computed might simply be due to the inefficient nature of random walk exploration search [Thrun, 1992]. The inefficiency of purely random action selection occurs because there is no memory of action selection or constraint over sequential action selection. The agent can end up selecting serial actions that cancel each other, therefore spending more time in a very localised area of the space [Pardo et al., 2018]. The search preference shown by rats in open-field DMP tasks in the watermaze [Bast et al., 2009], virtual maze [Buckley and Bast, 2018] and in a dry-land arena [Bast et al., 2005] may be more a measure of persistence of the memory of the goal, and the trace of a very localised search due to the imprecision of the memory of the goal location, as place information relies on distal cues in the watermaze. One can wonder what would the search preference look like on a DMP task variant that uses familiar goal locations [Whishaw, 1985], and in which rats are pretrained to switch between them. As the memory of the goal location might be more precise, the rats may start exploring the maze earlier, therefore leading to a reduced marked search preference compared to non-pretrained DMP tasks.

Third, hippocampal plasticity is required in open-field DMP tasks [Steele and Morris, 1999, Bast et al., 2005]. The current approach suggests that the adaptation necessary during trial 1 gives rise to the selection of a set of actor and critic weights that lead to the goal through the computation of a goal prediction error. However, the model does not explain how the computation of the goal prediction error would be linked to hippocampal mechanisms. It is possible that a positive prediction error would make the current event (being in the right goal location) salient enough to influence hippocampal activities to incorporate the new goal information [Blumenfeld et al., 2006, Gershman et al., 2014b], for example. In a spatial navigation task in which rats had to remember reward locations chosen according to different rules, McKenzie et al. [2014] have shown that hippocampal representations are hierarchical depending on the task requirement: if the context determined the reward location, the context would be the most discriminatory factor within hippocampal representations. Hok et al. [2013] found that prefrontal lesions decreased variability of hippocampal place cell firing and hypothesised that this was linked to flexibility mechanisms and rule-based object associations within hippocampal firing patterns

[Navawongse and Eichenbaum, 2013]. This finding shows that the prefrontal cortex can modulate hippocampal place cell activity. If the goal prediction error is coded by the prefrontal cortex, these findings imply that the goal prediction error could act on hippocampal representations to incorporate new task requirements (*e.g.*, information about the new goal location) and modify expectations.

Fourth, the neural basis proposed to hold parts of the computations proposed in the hierarchical model is not consistent with recent experimental results suggesting that prefrontal areas are not required for flexibility in the watermaze. A lesion study [Jo et al., 2007], as well as an inactivation study [McGarrity et al., 2015], in rats, indicate that prefrontal areas are not required for successful one-shot learning of new goal locations, or the expression of such learning, in the watermaze DMP task, and frontal areas were also not among the brain areas where electroencephalogram (EEG) oscillations were associated with virtual DMP performance in a recent study in human participants [Bauer et al., 2020]. This contrasts with the hierarchical RL model, which implicates "meta-learning" processes that may be associated with the prefrontal cortex.

Therefore, the current hierarchical model contains four main limitations that need to be addressed to be consistent with the results of experimental investigations in the watermaze DMP task. First, the model should be adapted for the agent to be able to adjust to new, never encountered before, goal locations. Second, the exploration strategy should be adapted to be more efficient. Third, hippocampal plasticity should be incorporated to represent new goal locations. And finally, the hierarchy proposed to be performed within prefronto-striatal loops isn't consistent with experimental results showing that prefrontal areas are not required for flexibility in the watermaze DMP task. In section 6.4.2, I present possible extensions of the model which would enable one to address each of these points.

6.3.3.2 The hierarchical model could explain behavioural flexibility in radial arm maze tasks

In this section, I propose that the hierarchical model could account for one-trial place learning in a task of the watermaze that requires pre-training to goal locations and radial arm maze navigation tasks, in which flexibility has been shown to depend on prefrontal areas.

The hierarchical RL model may account for one-trial place learning performance on a DMP task variant when the changing goal locations are familiar goal locations (*i.e.*, always chosen from a limited number of locations) [Whishaw, 1985]. Indeed, the variant of the task using familiar, pre-trained, goal locations may be solved by a hierarchical RL mechanism. Therefore, prefrontal contributions may become more important, a hypothesis that remains to be tested. The model motivates further experiments on this particular variant of the DMP task [Whishaw, 1985].

Other spatial navigation tasks require pre-training to goal locations. Indeed, pre-trained, familiar, goal locations are also a feature of delayed-non matching-to-place (DNMP) tasks in the radial maze (*e.g.*, Lee and Kesner [2002], Floresco et al. [1997]), where rats are first pre-trained to learn that food can be found in any of 8 arms (*i.e.*, these are familiar goal locations); after this, the rats are required to use a "non-matching-to-place" rule to chose between several open arms during daily test trials, based on whether they found food in the arms during a daily study or sample trial: arms that contain food during the sample trial do not contain food during the test trial and vice versa. Figure 6.11a presents a DNMP task: in a delayed spatial-win-shift task, used in Seamans et al. [1995] and Floresco et al. [1997], rats are placed in the middle of the maze in which four arms are blocked. In the four remaining arms, rats can find food rewards. After a variable delay, all arms are available, the previously blocked arms are now baited, and rats need to remember which arms they have previously visited to achieve optimal task performance. Errors are measured by counting the number of entries to non-baited arms, and rats are trained until this number is equal to, or less than, one for three consecutive days. Figure 6.12a describes another task using radial arm mazes which also requires pretraining: the random foraging task, used by Seamans et al. [1995]. This task is perhaps more similar to the DMP task in the watermaze. Rats are placed at the center of the radial arm maze, and are free to visit every arm. Four arms only are baited with food rewards. After a first visit and a variable delay, rats are placed again at the center of the maze. Performance is measured by counting the number of visit errors (*i.e.*, either a visit to an unbaited arm, or a re-visit to an already visited arm), as the optimal strategy for the rats to maximise their reward is to visit only the visited arms during the first visit. Every day, the four rewarded locations change. Rats are trained until the number of visits reaches a maximum of five (that is, one error only allowed, as there are four rewarded arms).

The hierarchical RL model may account for one-trial place learning performance in radial arm mazes tasks. Interestingly, the prefrontal cortex, the anterior cingulate cortex, and hippocampo-prefrontal interactions are required for one-trial place learning performance on radial maze (see figures 6.11b,6.11c and 6.12c, Seamans et al. [1995], Floresco et al. [1997]) and T-maze DNMP [Spellman et al., 2015] tasks, which involve daily changing familiar goal locations and, hence, may be supported by hierarchical RL mechanisms similar to my agent. Moreover, on the T-maze DNMP task, Spellman et al. [2015] found that hippocampal projections to mPFC are especially important during encoding of the reward-place association, but less so during retrieval and expression of this association. This is partly in line with the behaviour of the model, as the goal prediction error is important to select the appropriate strategy when the agent finds the correct goal location during the sample trial. However, hippocampal-prefrontal interactions are not (yet) considered in the model.

Although the hierarchical RL approach may be limited in accounting for key features of performance on DMP tasks using novel goal locations, it may have more potential in accounting for flexible trial-dependent behaviour displayed by rats on random foraging or spatial-win-shift tasks in the radial arm maze, which involve trial-dependent choices between familiar goal locations. DNMP performance in radial maze tasks requires NMDA receptors, including in the hippocampus, during pretraining. After pretaining,

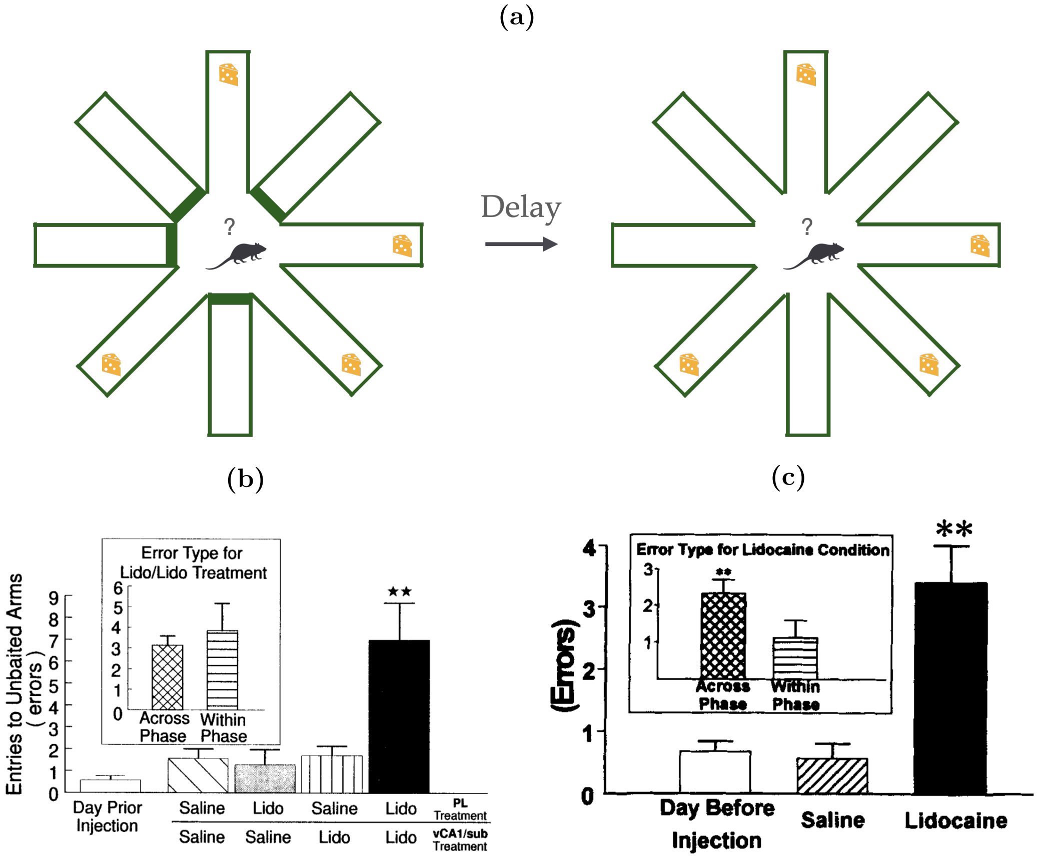


Figure 6.11: Hippocampal-Prefrontal connexion is required for flexibility in a delayed-spatial-win-shift task: (a) In a Delayed Spatial Win Shift task, rats are placed at the center of a 8-arm radial maze in which four arms are blocked and four arms contain food pellets, and can explore until they have discovered the food pellets. After a variable delay [Seamans et al., 1995, Floresco et al., 1997], the blocked arms are freed up and baited, and rats have to remember where they previously found the food pellets and visit the other arms in order to be optimal. Performance is measured by counting the number of entries to a non-baited arms, and rats are trained until this number is equal to or less than one for three consecutive days. After training, Floresco et al. [1997] impaired prelimbic cortex (PL, part of the frontal areas in rats) and CA1 area of the Hippocampus simultaneously and not, and found that simultaneous disconnection between CA1 and frontal areas impairs performance (b), adapted from [Floresco et al., 1997]. (c) effect of ACC lesion on the same task, adapted from [Seamans et al., 1995] with permission, showing that ACC is required for successful retrieval in the Spatial-Win-Shift task.

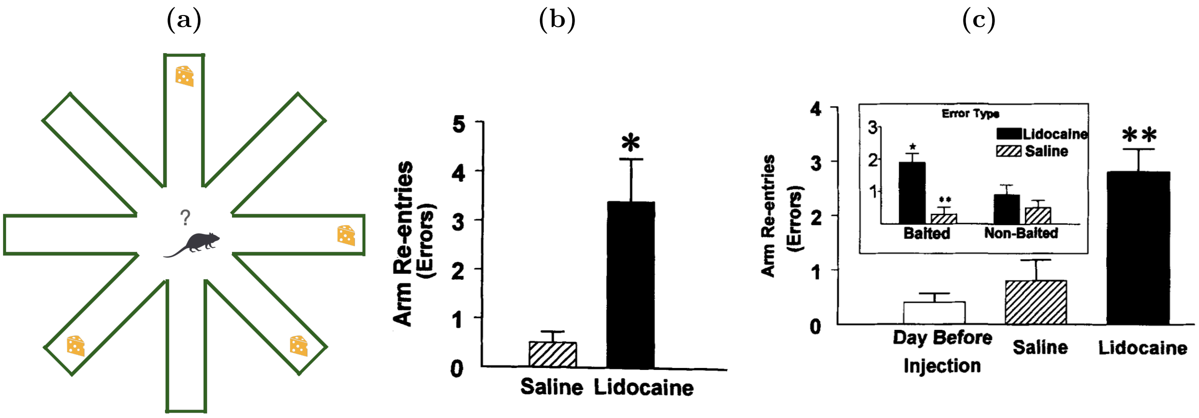


Figure 6.12: Prefrontal areas are required for flexibility in a random foraging task in a radial arm maze. (a) In a random foraging task in a radial arm maze, rats are placed at the center of a 8-arms maze and have to discover food pellets at the end of four arms. After a variable delay, rats are placed again in the same maze and performance is measured by counting the number of visit error, i.e., a visit to an unbaited arm or a re-visit to an already visited baited arm. Every day, the four baited arms change. Rats are trained until the total number of visits reaches five arms maximum. (b) Performance after training, PL lesioned rats cannot adjust to new baited arms anymore, showing much higher number of errors than control animals (Saline injection). Adapted from Seamans et al. [1995] with permission. (c) Effect of Anterior Cingulate Lesions on the task [Seamans et al., 1995]

and contrary to the watermaze [Steele and Morris, 1999] and event arena DMP tasks [Bast et al., 2005], rats can acquire and maintain trial-specific place information independent of hippocampal NMDA receptor mediated plasticity, even though the hippocampus is still required [Lee and Kesner, 2002, Shapiro and O'Connor, 1992, Caramanos and Shapiro, 1994]. The hierarchical RL architecture may account for this phase of acquisition of arm-reward association during pretraining to the eight possible goal locations, via the formation of actor and critic weights of every strategy. However, the plasticity considered is more consistent with changes in hippocampal-striatal connections, and the model does not address the role of plasticity within the hippocampus during the pretraining phase. Moreover, the hierarchical RL model also fits with the requirement of the prefrontal cortex for flexible spatial behaviour on arm-maze tasks, as described in the previous section.

The previous section exposed that one limitation of the hierarchical approach is the inefficiency of the exploration after a negative goal prediction error. I have explained that the search preference (figure 6.10) seems to be a measure of the persistence of the goal memory in tasks in which there is uncertainty over the locations, due to the distal nature of cues around the maze. In a radial arm maze task, the uncertainty over goal locations is almost none, as the goals are always at the end of arms. Therefore, the behaviour of rats in the radial arm maze random foraging task or win-shift paradigms may be explained by my model, as it would capture the change in behaviour caused by a negative goal prediction error. However, the random exploration should be adapted to be more efficient in searching different arms.

To test if a hierarchical RL architecture can reproduce behaviour on random foraging or spatial-win-shift arm-maze task, my implementation of a hierarchical RL model outlined above (see Fig. 6.7) would need to be adapted to the arm-maze environments, to the random foraging or spatial-win-shift rule and to an error measure of performance that is typically used in arm maze tasks (see figures 6.11b, 6.11c, 6.12c, and 6.12c, Seamans et al. [1995], Floresco et al. [1997]). In section 6.4.2, I give more details on how to adapt the model to these tasks.

6.4 Conclusion

6.4.1 Summary

In this chapter, I have proposed a novel architecture that combines ideas from HRL and meta-learning and uses the essential mechanisms that shape the behaviour of the actor-critic argent presented in chapter 4. I have described HRL as an extension to RL in which generalisation over actions enables more flexibility. I have applied this principle to the watermaze DMP task by separating the selection of the goal location to the selection of the actions that lead to the goal using a hierarchical architecture, in which the agent pre-learns a set of actor and critic weights, referred to as ‘strategy’, associated with different goal locations. I have described that meta-learning enables one to adjust behaviours according to changes in the task or in the environment at different timescales. Animals and humans adjust their behaviours to the statistics of the task or environments quite efficiently, and in particular, their degree of exploration, which seems related to the confidence or the level of certainty that they have in their current policy. I have thus proposed that the computation of a hierarchical error, namely the goal prediction error (6.1), can indicate the goodness of a strategy for the current goal, and therefore could serve to shape the degree of confidence that the agent has in its strategy. According to its degree of confidence, the agent explores or exploits more its current policy.

I have found that my hierarchical agent can adjust to changes in goal location. Moreover, it is partly biologically realistic. The hierarchical control, through the goal prediction error which shapes the behaviour of the agent, and in particular the changes in strategy, fits with experimental results from an extensive literature about the implication of prefrontal areas, including anterior cingulate cortex, on flexibility in spatial navigation. Their activity correlates with goal prediction errors, and corresponds to changes in behavioural strategies [Maggi et al., 2018], and, in particular, ACC responses to goal prediction errors are consistent with the confidence parameter proposed in the hierarchical model [Amiez et al., 2005]. Furthermore, they both receive input from one of the main areas of production of dopamine, the VTA, which as described in section 4.3.2, has been linked to reward prediction errors, which further fits with my model in which the goal prediction error is computed from the original TD error (4.6).

The current implementation, which requires pretraining to goal locations, does not represent faithfully the DMP task in which rats show one-shot learning of new, never encountered before goal locations [Bast et al., 2009, Steele and Morris, 1999]. However, I have outlined how the same architecture could be adapted to account for flexibility in radial arm maze tasks in which rewarded locations change every day too. In particular, experimental results for this task suggest that frontal areas, including the ACC and the mPFC, are required for flexibility in those tasks, which would be captured by my model. The hierarchical model predicts a reduced search preference for an area centred around the goal location.

6.4.2 Discussion: adaptation of the model to the radial arm mazes tasks

In this section, I briefly discuss how to proceed to adapt the model to the random foraging task (figure 6.12) and the delayed spatial-win-shift task (figure 6.11a).

To adapt the hierarchical model to both radial arm maze tasks mentioned, the first step would require one to adapt the environment, from a circular open-field to a radial arm maze as seen in figures 6.12 and 6.11a. The parameters of the actor-critic architecture should be adapted according to the difference of spatial scale between the watermaze and the radial arm mazes. To reduce interference between arms, a smaller scale of place cell representations (4.1) should be used.

The goal prediction error could be used to form a probability distribution over the potential strategies, which I will refer to as meta-policy in this section, linked with a "shifting" parameter that learns the particular rule of the task (shift or not). The visit to the arms during the first (training) trial would enable one to shape the meta-policy so that it becomes concentrated on the available visited goals. In the random foraging task, rats should visit the arms previously visited, and in the delayed spatial-win-shift task, rats should visit the arms not previously visited. Therefore, I propose that learning the task could involve training a shifting parameter that defines the probability to shift or maintain the meta-policy between the training trial and the test trial depending on the rule of the task. The shifting probability can be incorporated within the meta-policy to either reverse or preserve the meta-policy between the training trial and the testing trial to fit the requirement of the task. It can be trained based on a TD error computed from the difference between the rewards obtained during the test trial and the ones predicted by the meta-policy shaped by the switching parameter. In the spatial-win-shift paradigm, rewards are maximised when the meta-policy during the test trial is opposite to the one during the training trial, as arms that have been visited should not be visited again during the test trial (figure 6.11a). In the random foraging task, rewards are maximised when the meta-policy between the training trial and the test trial is maintained, as the same goal locations should be visited during the test trial (figure 6.12a).

Additionally, the exploration due to a goal prediction error should be adapted so that it is not random but directed towards the other arms. One solution would be to explore according to the meta-policy and not to a random walk so that the agent can navigate directly towards other goal locations after a negative goal prediction error.

To conclude, the extension to the radial arm maze delayed win shift and random foraging task would require training a switching probability using a meta-TD error, and incorporating a way to avoid revisiting of the same arm within one trial. The latter can be achieved using the hierarchical model presented in this chapter.

In this chapter, I have shown that hierarchical control is effective, in particular linked with meta-modulation of the behaviours, it can be mapped to biology, and I have incorporated these elements into a novel model for flexibility in navigation in the Morris watermaze DMP task.

Chapter 7

Predictive representations in attractor networks for rapid path planning

7.1 Introduction

Flexibility involves the ability to adapt plans when facing changes in the environment. By planning ahead, one can adapt the mechanisms learned to reach previous goals to novel situations. As the number of possible routes to investigate can be very high to reach a certain goal location, it can be crucial in order to react on time and manage resources to be able to plan efficiently by selecting the most likely scenarios.

In this chapter, I will present a model that enables the generation of preplayed trajectories to any goal location taking the shortest path, using an efficient representation of space [Corneil and Gerstner, 2015]. In two previous chapters, I have shown that minimal additions to a model-free architecture enable an agent to be flexible upon changing goal locations. One, in chapter 4, uses a map-like representation of locations to perform vector-based navigation to any goal location. The other, in chapter 6, uses hierarchical computations to switch between strategies depending on the current goal location. Both approaches overfit the particular task for which they are built: one uses Cartesian coordinates, which proves useful in an open field such as the watermaze, but which does not present a compelling representation for a general approach to spatial navigation. The latter only adjusts to already visited goal locations. The behavioural flexibility of rats in the watermaze discussed in section 2.5.3 suggests that rats use a trade-off between model-free and model-based control to perform the task, as discussed in section 3.3.2. The SR, presented in section 3.2.4, lies somewhere along this spectrum (figure 3.5).

The flexibility shown by both model-based and SR methods relies on search mechanisms that can be compared with planning (discussed in section 3.2.3). By searching through the tree of possible future options, a model-based or SR agent learns the value function of its current location and of the ones ahead and can choose a policy that maximises it before executing the sequence. While a model-based representation enables an agent to explore possible future trajectories based on the step-by-step transition probability matrix, the SR contains information about state occupancy on a longer timescale, as it represents how many times an agent can expect to visit the states in the future. As the number of states to consider in spatial navigation scales up in order to obtain a smooth representation of the environment, planning using an SR representation remains efficient.

In a spatial navigation context, planning is also hypothesised to underlie preplay activity patterns observed within hippocampal place cells when a rodent begins a trial in spatial navigation experiments [Foster, 2017] (discussed in section 2.3.2). Indeed, the generation of goal-directed trajectories has been observed in preplay activity patterns, suggesting that these are neural correlates of model-based mechanisms. Moreover, in addition to their investment in planning procedures, place cells and grid cells have firing profiles that adjust to the boundaries of the environment [Alvernhe et al., 2011], and this observation has been linked to place units obtained from the SR [Stachenfeld et al., 2017]. Therefore, planning using SR-based representation seems biologically plausible and might be involved in improving flexibility.

In this chapter, I present an approach from Corneil and Gerstner [2015] of path planning, which I implement in the environment originally proposed by Corneil and Gerstner [2015] and which I also adapt to planning in a circular open field environment mimicking the watermaze. I additionally propose a study of different values of relevant parameters of the model to investigate the relationship between precision, computational cost, and timescale of prediction. The approach uses results from graph theory to obtain an SR-based representation of the 2D space which links time and space, and in which computations are efficient. Section 7.2 introduces how to obtain such a representation and illustrates the relation between time and spatial representations in this framework. It is crucial that an efficient representation for prediction carries both information of space and time. This representation is used to define connections within a recurrent network, of which units are points of the space (section 7.3) and mimic place cells in the hippocampus. Section 7.4 describes how to use this network to generate dynamics that resemble those observed within place cell activities in the hippocampus. In the SR-based network, a bump of activity can be formed, which, when represented in its original 2D environment, stabilises around any location of the space. The bump of activity can be moved from one attractor to another in the neural activity space. When represented in its 2D equivalent, the bump forms around a starting location and smoothly moves towards any goal location, avoiding obstructions in the environment, or taking a direct trajectory when adapted to a watermaze-like environment. I show that the SR carries information about multi-timescale state occupancy, which can also be interpreted as diffusion across states at different timescales, which proves useful to reduce the dimension of state representation for efficient prediction. The model may explain preplay trajectories observed within hippocampal place cells [Foster, 2017].

7.2 An efficient representation of the environment for prediction using the SR

In this section, I will present a new representation of the space using the SR in which spatial representations also incorporate information about future state occupancy, which proves useful for computations related to predictions. I use graph theory to obtain an efficient representation of states, by interpreting the states in 2D space as nodes in a graph. I derive a high dimensional representation of the space from the SR, which carries information related to state occupancy at different timescales. In this high dimensional space, the value function is simply expressed as a dot product with the goal location. I start by forming a discretisation of the environment.

Two different spatial environments are considered. The first environment is a square maze, originally used by Corneil and Gerstner [2015], which contains walls (figure 7.1a). Second, I adapt the approach by Corneil and Gerstner [2015] to the watermaze, an open circular environment (figure 7.1b). In this section, I will refer to "the environment" as a general term for any of the two mazes. The environment is discretised as shown in figure 7.1. The two dimensional maze is represented by a set S of N states $s_j \in S$ covering the whole area, indexed from 1 to N. Figure 7.1 shows the location of each state across both environment. In figure 7.1a, the states are uniformly defined on a grid. For simplicity, they are given integer coordinates $(i, j) \in [1, \ldots, l] \times [1, \ldots, L]$. In figure 7.1b, the states are defined from a "sunflower" distribution, where the polar coordinate (r_j, θ_j) of state j is defined according to

$$r_j = R\sqrt{\frac{j - \frac{1}{2}}{N - \frac{(b+1)}{2}}}, \quad \theta_j = \frac{2\pi j}{\phi^2} \tag{7.1}$$

where R is the radius of the maze, $b = \left[\alpha\sqrt{N}\right]$, where [.] refers to the floor function (the greatest integer below the number), α indicates how much one cares about the evenness of the boundary and is set to a value of 2, and $\phi = \frac{\sqrt{5}+1}{2}$ is the golden ratio.

For each pair of states $(s_i, s_j) \in S^2$ indexed $i, j \in [1, \ldots, N]$, I define an affinity metric using a Gaussian kernel:

$$a(s_i, s_j) = \exp\left(-\frac{d(s_j, s_i)^2}{2\sigma^2}\right), \tag{7.2}$$

where $d(s_j, s_i) = d(s_i, s_j)$, so that also $a(s_i, s_j) = a(s_j, s_i)$, is the length of the shortest traversable path between s_i and s_j, respecting walls and obstacles, and σ describes the width of the kernel. The affinity metric defines how connected the network is. As the kernel is strictly positive, every state is connected to all other states of the network. If σ

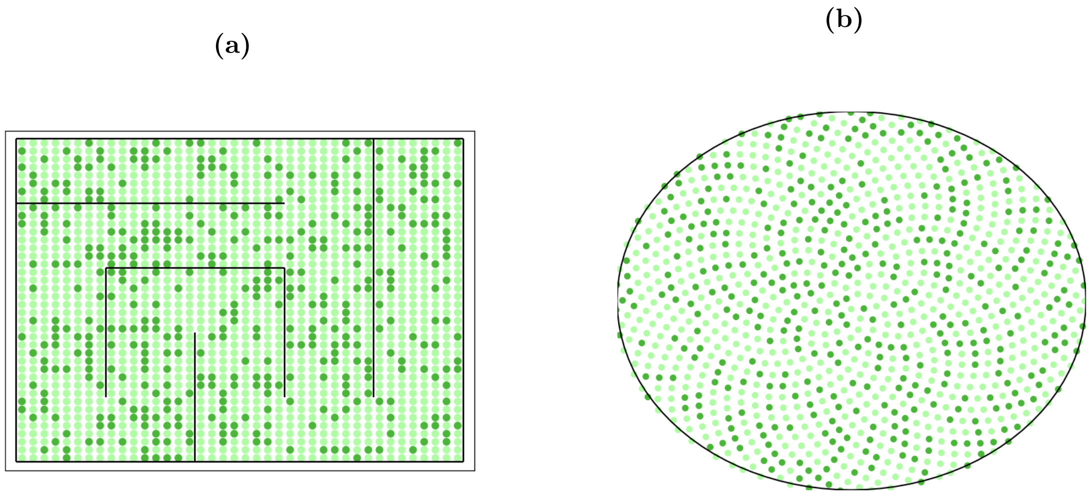


Figure 7.1: Maze shape, state centres (light green) and place cell centres (dark green) for the environments used in (a) Corneil and Gerstner [2015] and b) for the watermaze. In (a), states coordinates are defined from a fine grid with integer coordinates (i, j) for simplicity, with $i \in [1, \ldots, l]$, and $j \in [1, \ldots, L]$. In (b), they are defined using a sunflower distribution given by equation (7.1). Dark green dots mark the locations of the place cell centres, randomly selected among the states. Dark lines show walls and maze limits.

is high, a state in the network is strongly connected to many other states. In contrast, if σ is low, a state of the network is strongly connected to few states and weakly connected to all others. The affinity metric in equation (7.2) is then normalised to compute the probability of a transition from state s_i to s_j:

$$p(s_i, s_j) = \frac{a(s_i, s_j)}{\sum_{s_l \in S} a(s_i, s_l)}. \tag{7.3}$$

As $\sum_j p(s_i, s_j) = 1$, the matrix P defined by $P(i, j) = p(s_i, s_j)$ for every $i, j \in [1, \ldots, N]$ defines a transition probability matrix for the Markov chain formed by the states of the maze. In the next section, I will show that this matrix can be seen as the transition matrix over a random walk in the graph of states S with the affinity metric $A \in \mathbb{R}^{N \times N}$, defined by $A(i, j) = a(s_i, s_j)$.

7.2.1 The transition probability matrix and its eigendecomposition

In this section, I demonstrate that the transition probability matrix (7.3) defines transitions both in the 2D space and in a graph constructed from the affinity metric (7.2), and I use graph theory and linear algebra to show that P is diagonalisable.

In the state space S can be extended in a set (S, E, A), in which E is a set of vertices between the nodes of the graph defined from the state space S, weighted by the symmetric and strictly positive adjacency matrix A (7.2), forms a fully-connected graph. Its graph Laplacian, a matrix which describes diffusion phenomena within graphs, is given by:

$$L = D - A, \tag{7.4}$$

where D is the diagonal matrix defined by $D(s, s) = \sum_{s'} a(s, s')$, also called degree matrix. The normalised graph Laplacian is given by:

$$\mathcal{L} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}, \tag{7.5}$$

where I is the identity matrix. Applied to the example here, this becomes

$$D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = P, \tag{7.6}$$

where P is given in equation (7.3). The transition matrix, given by

$$T = D^{-1} A, \tag{7.7}$$

defines a random walk on the graph (S, A) [Mahadevan, 2009]. As $A > 0$, one gets

$$T = P. \tag{7.8}$$

As P is strictly positive, the Perron-Frobenius Theorem [Pillai et al., 2005] ensures that there is an unique invariant probability distribution on states, denoted by z. It can be shown that P defines a reversible Markov chain. In the particular example considered here, equation (7.3) leads to:

$$\begin{aligned} p(s_i, s_j) &= \frac{a(s_i, s_j)}{\sum_{s_l \in S} a(s_i, s_l)} \\ &= \frac{a(s_i, s_j)}{\sum_{s_l \in S} a(s_i, s_l)} \times \frac{\sum_{s_l \in S} a(s_j, s_l)}{\sum_{s_l \in S} a(s_j, s_l)} \\ &= \frac{\sum_{s_l \in S} a(s_j, s_l)}{\sum_{s_l \in S} a(s_i, s_l)} p(s_j, s_i), \end{aligned} \tag{7.9}$$

for all $(s_i, s_j) \in S \times S$. This condition guarantees that P defines a reversible Markov chain with the additional information (equation (7.9)) that the invariant probability distribution z is linked to the affinity metric:

$$z(s_i) \propto \sum_{s_l \in S} a(s_i, s_l), \tag{7.10}$$

for every $s_i \in S$. As discussed in section 7.2, the kernel A (7.2) defines the strength of connection between nodes of the graph or states in the network. In equation (7.10), the sum term on the right hand side can be interpreted as an "average reachable distance" between the node s_i and all other states in the network: a high value indicates that s_i has many close neighbours in the network, a low value suggests a more isolated state within the network. In a spatial navigation context, this measures the degree of accessibility of the states, and it seems intuitive that the stationary distribution, which determines the probability of state visit in infinite time (see section 7.2.2), is proportional to how accessible the states are. From Coifman and Lafon [2006], one can show that P has real eigenvalues and orthonormal eigenvectors with respect to the same inner product defined by the invariant distribution z. Therefore, there exists a set of vectors $\varphi_1, \ldots, \varphi_N$ and a set of scalars $\lambda_1, \ldots, \lambda_N$ such that for every $i \in [1, \ldots, N]$:

$$P\varphi_i = \lambda_i \varphi_i. \tag{7.11}$$

Moreover, the set of eigenvalues $\lambda_1, ..., \lambda_N$ satisfies $1 = |\lambda_1| \geq |\ \lambda_1| \geq ... \geq |\lambda_N|$ [Coifman and Lafon, 2006] (see appendix D.1 for demonstration).

In this section, I have shown that the Markov chain formed by the states space S equipped with the transition matrix P given by equation (7.3) can be described by a random walk on an undirected graph whose nodes are defined by the states in S and whose weight matrix is given by the affinity metric in equation (7.2). Therefore, the description of motion within Cartesian space is analogous to the description of movement within the graph. In the next section, I show that the eigenvalues and their associated eigenvectors are linked to different timescales of description of the state occupancy (where an agent can expect to be), both within the graph and the Cartesian plane.

7.2.2 Link between the eigenvalues, space and time

In this section, I investigate the behaviour of the set of eigenvalues when considering iterations of the matrix P (7.3), which describe diffusion within the graph at different timescales. When looking at the graph representing a 2D environment in a spatial navigation context, it is interesting to investigate how the state occupancy evolves with time. The transition probabilities given by the matrix P (7.3) describe its evolution: it provides the probabilities of an agent moving from one state to another. Therefore, the evolution of the state occupancy within the graph is given by a multiplication with the matrix P. The evolution of the state occupancy with time is given by repeating the multiplication with the matrix P. Therefore, the evolution of the state occupancy after time t is given by a multiplication with P^t. As P can be diagonalisable, the multiplication by P^t is fully described by the power of its eigenvalues $\lambda_1^t, ..., \lambda_N^t$. Considering the ordering $1 = |\lambda_1| \geq |\ \lambda_2| \geq ... \geq |\lambda_N|$, the only eigenvalue that does not decrease at every iteration is $1 = |\lambda_1|$. Convergence is a general property of Markov chains. Indeed, the distribution of a Markov chain converges to the stationary distribution z [Baxter and Rosenthal, 1995]. That means that the probability of occupancy of a state s converges to $z(s)$ when t goes to infinity. One can prove that the convergence to the stationary distribution is geometric [Baxter and Rosenthal, 1995].

The time of convergence of the Markov chain to the stationary distribution depends on the affinity metric A (7.2). Figure 7.2 shows the eigenvalue distribution at different times for two values of the width σ of the kernel A. For a larger value of σ, the non-trivial eigenvalues (less than 1) decay faster than for a lower value of σ. Therefore, for a larger kernel (larger σ), the speed of convergence is faster, as can be seen by comparing the dotted and plain curves evolution as t grows. Intuitively, the interpretation of this is that a larger value of σ involves a stronger connectivity of a state to other states of the network due to a wider affinity metric A. Therefore, it is faster to travel through the graph, as nodes are closer to each other.

As the eigenvectors of P are orthonormal, they form a basis in the state space. The eigenvectors can be seen both as a basis of the state and as a description of the state prediction occupancy at different timescales. Figure 7.3 shows the first five eigenvectors plotted over the state space, and more are given in Appendix D.2. The eigenvectors are ordered in decreasing magnitude of their associated eigenvalue. Figure 7.3 shows that the eigenvectors of the transition matrix behave like a Fourier decomposition of the physical environment: the first eigenvector (associated to the larger eigenvalue, $\lambda_1 = 1$) is uniform throughout the environment, the second increases linearly from one end to another, and as the value of the eigenvalue decreases, the spatial frequency carried by the associated eigenvector increases. The first eigenvector is proportional to the stationary distribution. This result can be interpreted by the following: in infinite time, the agent can expect to have visited all states an equal number of times, or the probability of state occupancy of the agent is uniform across the states. To guarantee the stability of the system, the value of the non-null eigenvectors over the space should be compensatory: areas of positive value should coexist with areas of negative value, which explains their "periodic" nature. The eigenvector represents a state occupancy at a timescale that depends on the eigenvalue (smaller eigenvalues are faster to converge, section 7.2.2). Intuitively, faster diffusion mechanisms can be described by higher spatial frequencies, which decay quickly with repeated iteration of the transition.

In this section, I have linked the spatial scale of the network connectivity and the timescale of evolution of the state occupancy that I will show is useful for prediction. A new basis for the state space is defined by the eigenvectors of the transition probability

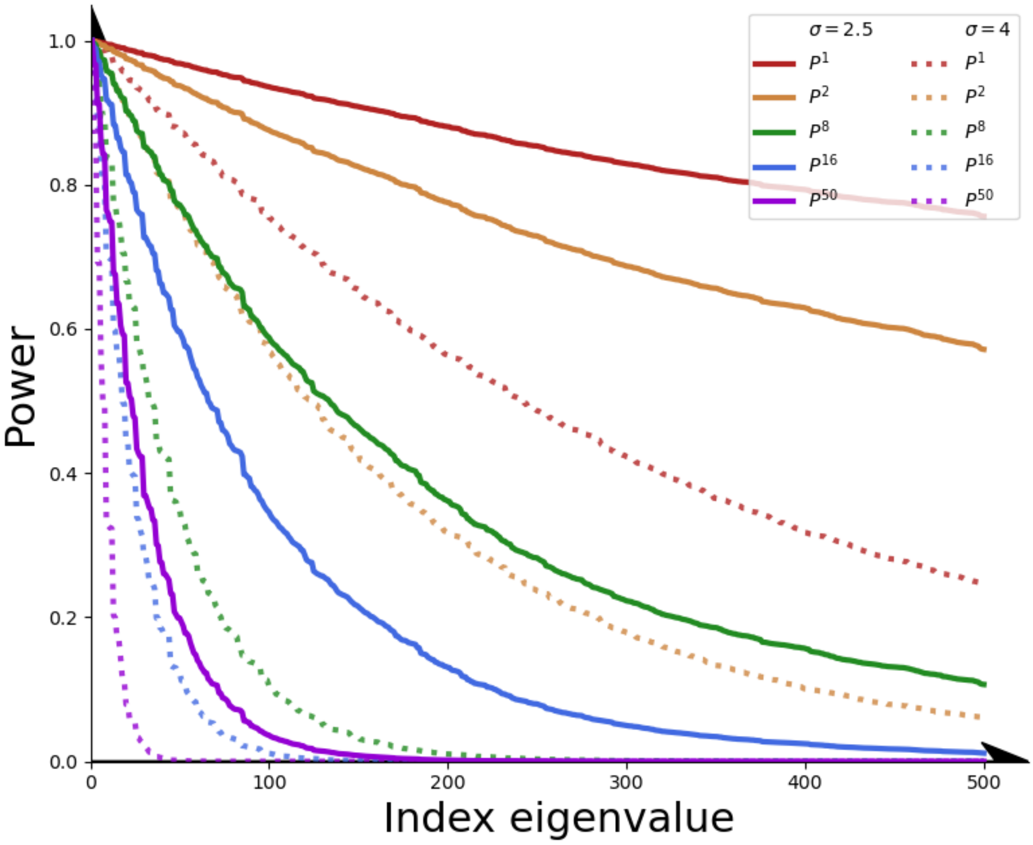


Figure 7.2: Eigenvalue spectrum for different powers P^t of P. On the x axis are shown the indices of the eigenvalue. As time increases, the "rank" of P^t decays (if one considers the rank the number of significant eigenvalues, *i.e.* higher than a threshold. A multiplication by P is equivalent to one displacement within the graph under the probability transition P. As time goes on, the power of the lower eigenvalues decrease, therefore, after infinite time, the state occupancy is given by the uniform distribution. For higher values of the width σ of the kernel (7.2), a state in the graph is more strongly connected to other points than for a lower value, resulting in a much faster dampening timescale on the spectrum of the eigenvalues, as can be seen by comparing the dashed to full lines on the graph. When the graph is more strongly connected, it is much faster to travel in it, and fewer iteration of P are required to significantly reduce its rank and for the state occupancy to be described by the uniform distribution.

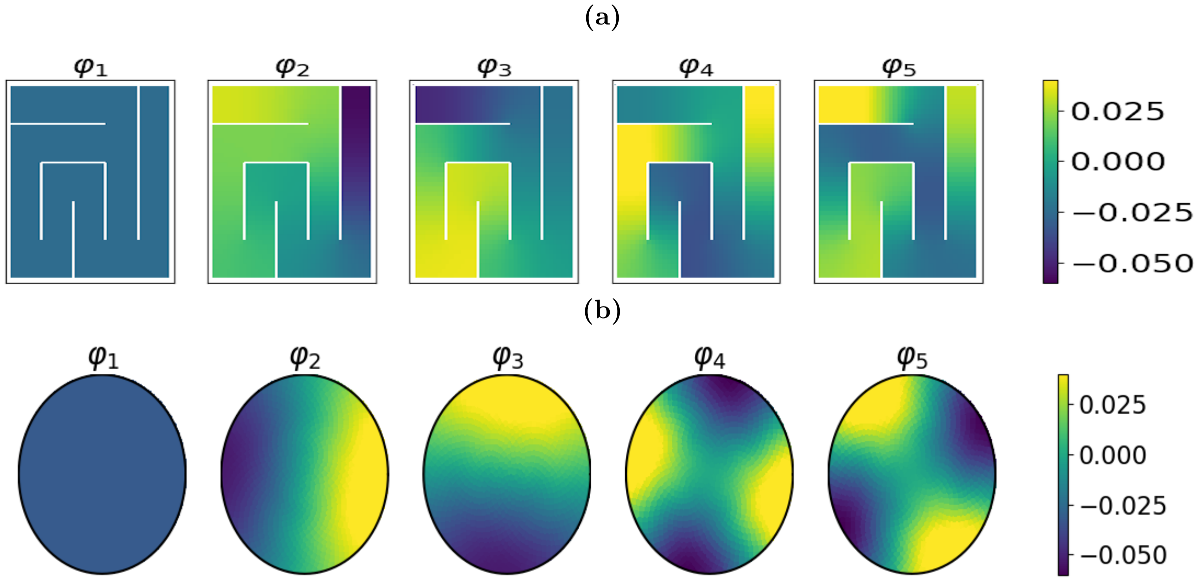


Figure 7.3: Eigenvectors of the transition matrix P, from the first (left) to the fifth (right) eigenvector, in (a) the square environment with walls proposed by Corneil and Gerstner [2015] and (b) the watermaze. With the decreasing value of the associated eigenvalue, the eigenvector progressively encodes a higher spatial frequency. The spatial frequency appears clearer in the maze containing walls. The fact that the second and third eigenvectors seem rotated can be intuitively understood as the eigenvectors reflect PCA of the chosen kernel (7.2). As the kernel is Gaussian, it is consistent that, after the uniform eigenvector, the two following eigenvectors which carry most of its variance are linear along orthogonal dimensions. More eigenvectors are provided in Appendix D.

matrix P. I showed that these provide a form of periodic decomposition of the state space, similar to a Fourier decomposition, and that they are associated to the states occupancy at different timescales. In the following section, this basis is used to obtain a very simple formulation of the value function (3.19).

7.2.3 The successor representation allows a simple expression of the value function

In this section, a simple expression of the value function is formulated using the eigenvectors basis (7.11). In the case where a single goal location at state $s_j \in S$ is considered, the reward function r over the state space in equation (3.20) is $r(s_i) = \delta_{i,j}$, for every $s_i \in S$, where δ denotes the Kronecker delta function. This corresponds to selecting a column in the SR equation (3.20), the column corresponding to the index of the goal location. By selecting the index of the vectorial expression of the value function (3.20),

$V_P = (\mathbb{1}_N - \gamma P_\pi)^{-1} r$, corresponding to state s_i, the value function of state s_i becomes:

$$\begin{aligned} v(s_i|s_j = g) &= z(s_j) \sum_{l=0}^{N-1} (1-\gamma\lambda_l)^{-1} \varphi_l(s_i)\varphi_l(s_j) \\ \text{or} \quad v(s_i|s_j = g) &= z(s_j)\langle \overline{s_i}, \overline{s_j}\rangle, \end{aligned} \tag{7.12}$$

where $\langle ., . \rangle$ denotes the scalar product, $z(s_j)$ refers to the steady-state occupancy of s_j given the transition matrix P. The steady-state occupancy z refers to the probability distribution over states which is invariant by action with P. Numerically, it is equivalent to the first eigenvector in figure 7.3. $\overline{s_i}$ refers to the i-th state s_i expressed from the new SR basis formed by the vectors:

$$\left(\sqrt{(1-\gamma\lambda_1)^{-1}}\varphi_1, \sqrt{(1-\gamma\lambda_2)^{-1}}\varphi_2, \ldots, \sqrt{(1-\gamma\lambda_N)^{-1}}\varphi_N\right). \tag{7.13}$$

In this section, I have adopted a graph-theoretic perspective to represent the 2D euclidean space. Using results from graph theory, a new representation of the points in the space based on successor coordinates has been defined. In this new basis, the formulation of the value function becomes a simple dot product. Here, a unique goal location is considered, and so the value function has an unique maximum. The simplicity of the formulation of the value function will facilitate the computation of a displacement in the direction of its gradient ascent. In the following section, the SR basis is used to define a recurrent network.

7.3 An SR-based recurrent network

In this section, a lower-dimensional subspace of the eigenspace (7.13) that allows a reduction of the dimension with minimal representational loss is considered. As the dimension has been reduced, the computations involved are less costly. The new state representation based on the SR defines the connection between nodes of a network that is used to form a bump of activity that moves from any start to any goal location within the network that can be interpreted as a gradient ascent of the value function.

7.3.1 A subspace of the eigenbasis for am efficient value gradient ascent procedure

The number of states N used to derive the SR representations is chosen to be large in order to obtain a high resolution of the space representation. As N also defines the dimension of the space defined by the SR basis (7.13), it can make computations in such a high dimensional space costly. Therefore, cutting the dimension of the space and keeping only the q first eigenvectors enables one to work in a space of lower dimension in which computation is more efficient. Section 7.2.2 described that the eigenvectors form a basis of the space which increases in spatial frequency as the index increases. Cutting the dimension of the space by only keeping the subspace defined by the q first eigenvectors (7.13) involves cutting off the highest spatial frequencies that have low representation weight (low associated eigenvalues), and therefore affects only minimally the precision of the representation (depending on the choice of q). Note that this dimensionality reduction approach is similar to dimensionality reduction using PCA [Coifman and Lafon, 2006]. In fact, the eigenvector decomposition of the SR corresponds to doing PCA in the space defined by the non-linear transformation of the points using the affinity metric (7.2) (see Ham et al. [2004] for more detail about the links between dimensionality reduction, PCA and kernel methods).

A subspace of the space defined by the successor coordinates in (7.13) is considered to construct $\hat{S} = \{\hat{s}_i, i \in [1, \dots, N]\}$ the set of states expressed in this subspace that represents the state space S using a set of successor coordinates. In this subspace, the coordinates of state i, for $i \in [1, \dots, N]$, are given by:

$$\begin{aligned} s_i \rightarrow \hat{s}_i &= \left(w_1 \varphi_1(s_i), \sqrt{(1-\gamma\lambda_2)^{-1}}\varphi_2(s_i), \dots, \sqrt{(1-\gamma\lambda_q)^{-1}}\varphi_q(s_i) \right) \\ &= \left(\xi_1(s_i), \xi_2(s_i), \dots, \xi_q(s_i)\right), \end{aligned} \tag{7.14}$$

where

$$\xi_l = \sqrt{(1-\gamma\lambda_l)^{-1}}\varphi_l \tag{7.15}$$

is a vector of the new basis $\{\varphi_i\}_{i\in[1,\dots,N]}$ (7.13), and q is a parameter that determines the dimension of the space. In (7.14), $\varphi_j(s_i)$ refers to the s_i coordinate of φ_j. Additionally, as φ_1 is constant throughout the environment, the coefficient $\sqrt{1-\gamma\lambda_1}$ is replaced by a free parameter w_1 that defines the weight of the first component compared to the other components and will influence the global activity of the network. The effect of the choice of both q and w_1 on the representation and on the dynamics within the network are described in section 7.5.

In the eigenbasis representation, the value function expression is the dot product between the current state and the goal state (7.12). In the subspace defined by the q vectors ξ_i in which states coordinates are expressed in equation (7.14), the value function (7.12) can be approximated by the dot product in this subspace:

$$v(s_i|s_j = g) = z(s_j)\langle\overline{s_i}, \overline{s_j}\rangle \approx z(s_j)\langle\hat{s}_i, \hat{s}_j\rangle. \tag{7.16}$$

In the new representation, the value landscape is particularly simple as the value of any point throughout the environment is given by a simple scalar product (equation (7.16)). The next section approximates a gradient ascent procedure towards a goal location in an SR-based attractor network.

7.3.2 Approximation of the gradient ascent of the value function in the SR coordinates

In this section, I formalise the gradient ascent procedure from any starting location of the space to any goal location of the space. Starting from a state $\hat{s}_0$ of the network, and in order to move to a goal position $\hat{s}_g$, one can consider a constrained gradient ascent process on the value landscape given by:

$$\hat{s}_{t+1} = \underset{\hat{s}\in\hat{S}}{\operatorname{argmin}}\left[(\hat{s} - (\hat{s}_t + \alpha\nabla v(\hat{s}_t)))^2\right]. \tag{7.17}$$

Following (7.16), the formulation of the gradient ascent procedure in the new coordinates can be simplified to:

$$\hat{s}_{t+1} = \underset{\hat{s} \in \hat{S}}{\text{argmin}} \left[(\hat{s} - (\hat{s}_t + \alpha \hat{s}_g))^2 \right], \tag{7.18}$$

where the steady-state occupancy $z(s_g)$ (7.12) has been absorbed in the parameter α. The gradient ascent procedure can be approximated by the step $\hat{s}_t + \alpha \hat{s}_g$ in the current state space S.

In the following, I will explain that one can approaximate the step $\hat{s}_t + \alpha \hat{s}_g$ using a recurrent network. The goal $\hat{s}_g$ will be an attractor state provided as input to the network. I first define the network based on the eigenbasis representation in the next section.

7.3.3 An SR-based network

The SR basis is used to define recurrent weights within an attractor network of units mimicking hippocampal place cells. First, the network is defined by $n = 500$ neurons sampled uniformly throughout the environment among the N states (see figure 7.1, with dark green dots indicating the neuron centres and in clear green the state centres). Each neuron's activity depends on its coordinates in the successor representation. For every $i \in [1, \cdot, n]$, the encoding vector e_i of the ith neuron in state s_i is defined from the SR coordinates (7.14) by:

$$e_i = \frac{\hat{s}_i}{||\hat{s}_i||}. \tag{7.19}$$

The encoding vector is used to shape the response properties of neurons in the network to any input within the space. The following section shows that the encoding vectors prove useful to determine response patterns consistent with those of hippocampal place cells, and lead to the formation of a bump of activity around a start location, which moves using efficient trajectories.

7.4 Network dynamics: a steady bump of activity moves towards any goal location

In this section, the activity of the network is defined based on the SR coordinates. The activity within the network can form a bump of activity representing any location, and move smoothly towards any goal location in the space. I show that this can be interpreted as a gradient ascent of the value function (7.18), achieved by providing a goal location as input to the network, which becomes an attractor state for the activity of the network. This requires two steps: First, decoding weights will be defined for the network to read its own activity and retrieve correctly the position of the bump. This defines a recurrent input given to the network that represents the current position of the bump in order to maintain a steady profile of activity around its current position. Following this, the input to the attractor network is changed to the goal location, so that it *drags* the activity of the network towards it. This causes the bump of activity to move towards the goal location.

7.4.1 Forming a steady activity profile representing a location in the space

In this section, I show how to create a steady profile of activity throughout the network defined in section 7.3.3 that represents a location as encoded within hippocampal place cells. The objective is to create a stable pattern over time. First, the dynamics of the responses of neurons of the network to an input is described. Then, recurrent weights are defined in order for the network's activity to reflect a location encoded by an input given to the network. The resulting recurrent input guarantees that the network's activity stabilises around this location.

7.4.1.1 Response of the network to an input: SR-based feedforward weights reflect place cell connectivity

An input $\hat{s}^{\text{in}}(t) = \sum_{k=1}^{m} \xi_k(s^{\text{in}}(t)) = \sum_{k=1}^{m} \hat{s}_k^{\text{in}}(t)$ is expressed in the successor coordinates (7.14). Typically, this input can be understood as representing a location in the space, and $m \geq q$, where q is the dimension of the space in which the network is embedded (see section 7.3.1), with $\hat{s}^{\text{in}}_k(t)$ being unique. $\hat{s}^{\text{in}}(t)$ can also represent a random collection of inputs (for example, a set of locations) expressed in the ξ-basis (7.14). m here is chosen heuristically, there should be enough coverage of the whole environment in order to facilitate encoding of every location within the network. Every neuron $i \in [1, \ldots, n]$ responds to the input $\hat{s}^{\text{in}}_k(t)$ through feedforward weights $w_{i,k}$

$$w^{\text{ff}}_{i,k} = e_i(k), \tag{7.20}$$

where e_i is the encoding vector for neuron i defined in equation (7.19), and k the kth coordinate of the input $\hat{s}^{\text{in}}(t)$ expressed in the basis (7.14). The activity a_i of neuron i in response to the m inputs $\hat{s}^{\text{in}}(t) = \sum_{k=1}^{m} \xi_k(s^{\text{in}}(t)) = \sum_{k=1}^{m} \hat{s}^{\text{in}}_k(t)$ evolves according to a first order differential equation given by:

$$\tau \frac{\mathrm{d}a_i(t)}{\mathrm{d}t} = -a_i(t) + g\left[\sum_{k=1}^{m} w^{\text{ff}}_{i,k}\hat{s}^{\text{in}}_k(t)\right]_+, \text{ for } i \in [1, \ldots, n], \tag{7.21}$$

where τ is the time constant of the neuronal activity, determining how fast the activity of the network fades without any input, and how fast will it take to converge to a steady input. $g > 0$ is a gain factor, and $\left[.\right]_+$ refers to a rectified linear (ReLu) function: it is zero for negative arguments, and linear in its argument otherwise. The choice of a ReLu activation function can be advantageous to facilitate the formation of a bump of activity because it does not cut off large positive values, and is partly biologically realistic as it does not cap the neurons' activity to an arbitrary value. Moreover, ReLu facilitates the computation of a gradient, which will also permit the dynamics of the network to approximate the gradient ascent of the value function (7.18).

The dynamics of the neural activity given by equation (7.21) in response to the inputs can then be expressed as a scalar product between the neuron encoding vector and the input:

$$\tau \frac{\mathrm{d}a_i(t)}{\mathrm{d}t} = -a_i(t) + g\left[\langle e_i, \hat{s}^{\text{in}}(t)\rangle\right]_+, \text{ for } i \in 1[\ldots n]. \tag{7.22}$$

where $\langle ., . \rangle$ refers to the dot product. A neuron is therefore maximally active when the input vector is aligned to its encoding vector. Figure 7.4 illustrates this phenomenon: it shows the input that a particular neuron receives (red circle) from the rest of the network. Neighbouring states in the attractor network, which have similar SR coordinates, strongly project to the neuron. In figure 7.4a, the input strength decays following the geometry of the environment (following the corridor of the maze) showing that the strength of the input is stronger when the input states are close to the neurons in "reachable space", as opposed to reflecting Euclidean distance. This reflects the choice of the distance used to compute the affinity metric (7.2), defined as the length of the shortest path between places. Changing the distance metric will change the result. For example, if the kernel chosen is not Gaussian but rather has elliptic levels, the input profile would be skewed slightly along these ellipses.

The profile of the input that a neuron receives is consistent with what one would expect from a place cell attractor network: active place cells influence positively the activity of cells that encode similar locations. This bears a resemblance to the preplay patterns discussed in section 2.3.2, in which the activation of a place cell when the animal is placed on its preferred position triggers the bump of activity to travel through the place cells corresponding to the future trajectory of the animal, suggesting that place cells have excitatory connections toward cells that encode similar positions.

7.4.1.2 Decoding the activity of the network to reflect locations given as inputs

In this section, equation (7.21) is used to initiate the network activity from inputs representing locations in the environment, from which decoding weights are built to enable the network to maintain its activity around a location.

The activity of the network in response to m inputs $(\hat{s}_k^{\text{in}})_{k\in[1,\ldots,m]}$ scattered throughout the environment and expressed in the SR coordinates given by (7.14) provides an encoding of the inputs locations. A recurrent input from the network to itself is defined in order to form a stable bump of activity around an input location. If $\hat{s}_{\text{rec}}$ denotes the recurrent

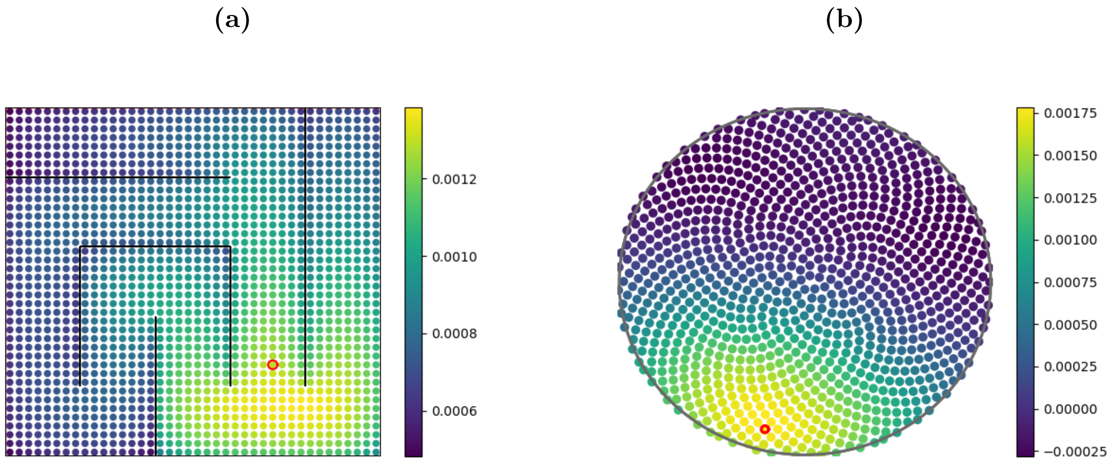


Figure 7.4: Strength of the input from all states to the neuron shown by a red circle, showing the right hand side of equation (7.22), the dot product $\langle e_i, \hat{s}^{\text{in}}(t)\rangle$, as if the inputs were located at the state locations, *i.e.* $\hat{s}^{\text{in}} = \hat{s}_j/||\hat{s}_j||$, with $j \in [1\ldots, N]$, and e_i the encoding vector of the unit represented as a red circle, given in equation (7.19) in (a) a square environment with walls proposed by Corneil and Gerstner [2015] and (b) the watermaze. In both environments, the dot product between the neuron and the input gives rise of a locally excitatory and progressively fading out weight profile when the input center gets further from the neuron's center. This seems consistent with a place cell recurrent network, as seen by the preplay patterns covering the future trajectory of the animal observed in recordings when an animal enters the environment, suggesting that place cells have an excitatory connection towards place cells that have a similar place preference.

signal decoded from the neural activities, decoding weights $(d_j)_{j\in[1,\ldots,n]}$, $d_j \in \mathbb{R}^q$ can be defined to express $\hat{s}_{\text{rec}}$ from the network activities $(a_j)_{j\in[1,\ldots,n]}$ according to:

$$\hat{s}_{\text{rec}}(t) = \sum_{j=1}^{n} d_j a_j(t). \tag{7.23}$$

For the network's activity to be able to stabilise around any location within the space, the matrix $D \in \mathbb{R}^{q\times n}$ of decoding weights $(d_j)_{j\in[1,\ldots,n]}$, $d_j \in \mathbb{R}^q$ that minimise the difference $||\hat{s}_{\text{rec}} - \hat{s}^{\text{in}}||$ for multiple inputs $\hat{s}^{\text{in}}$ scattered around the space [Eliasmith and Anderson, 2003] is defined according to:

$$D = \underset{D\in\mathbb{R}^{q\times n}}{\operatorname{argmin}} \left|\left| \hat{S}^{in} - D \times \bar{A} \right|\right| \tag{7.24}$$

where

$$\hat{S}^{in} = \begin{bmatrix} \hat{s}_1^{\text{in}} & \ldots & \hat{s}_m^{\text{in}} \end{bmatrix} \in \mathbb{R}^{q\times m}$$

encodes the inputs coordinates $\hat{s}_k^{in}$,

$$\bar{A} = \begin{bmatrix} \bar{a}_{1,1} & \dots & \bar{a}_{1,m} \\ & \dots & \\ \bar{a}_{n,1} & \dots & \bar{a}_{n,m} \end{bmatrix} \in \mathbb{R}^{n \times m}$$

contains the steady-state activities $(\bar{a}_{i,j})_{i=1\dots n,\ j=1\dots m}$ of neuron i in response to input j, and

$$D = \begin{bmatrix} d_1 & \dots & d_n \end{bmatrix} \in \mathbb{R}^{q \times n}$$

are the decoding weights. The decoding weight $d_i, i \ \in [1, \dots, n]$, can be understood as the sensitivity of neuron $i, i \in [1, \dots, n]$, to each dimension of the coordinates ξ (7.14).

The problem of finding the decoding weights as formulated in equation (7.23) is a linear regression problem. The best approximation of this solution is given by the Moore-Penrose pseudoinverse of the matrix of steady-state activities [Penrose, 1956], which generalises the inverse to any matrix, even non-square ones [Barata and Hussein, 2012]. Therefore, D is defined as the Moore-Penrose pseudoinverse of the matrix of steady neural activities in response to the example inputs. The decoding weights are vectors that enable one to read and retrieve the position encoded by the input given to the network from the activity of the network. In the next section, the decoding weights are used to define a recurrent term to stabilise the activity around a location in the space.

7.4.1.3 Using the decoded location as a recurrent signal to stabilise the activity profile of the network

The decoding weights given by (7.23) allow one to define recurrent weights $w_{i,j}^{\text{rec}}$ so that the network can maintain a memory of the past. The recurrent weights are determined by projecting the decoded location back on to the neuron encoding vectors:

$$w_{i,j}^{\text{rec}} = (1 - \varepsilon)\langle e_i, d_j \rangle, \text{ for } i, j \in [1, \dots, n], \tag{7.25}$$

such that the full recurrent input to neuron i can be defined as:

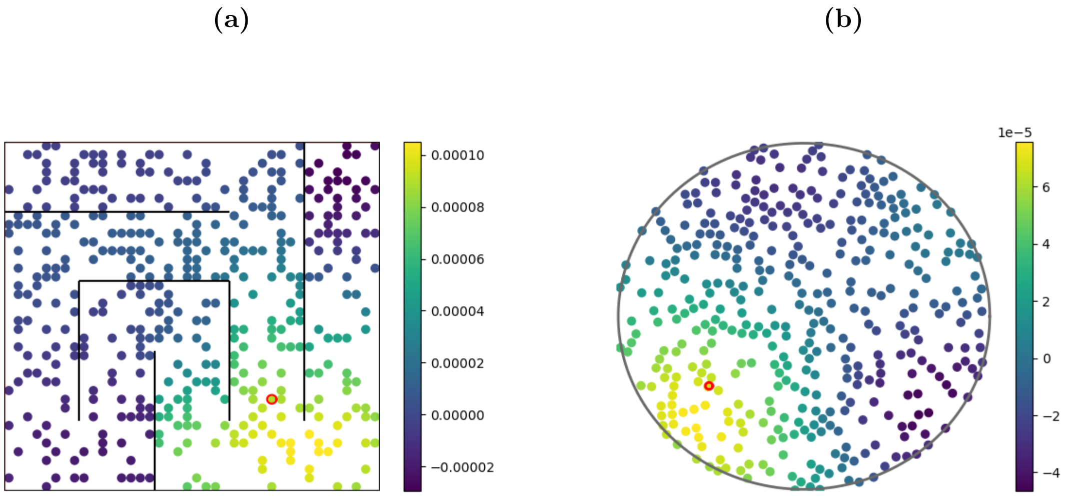


Figure 7.5: Recurrent weights $w_{i,j}^{\text{rec}}$ from all the neurons to the neuron at the center of the red circle (the colour of the points shows the strength of the weight), computed from equation (7.25) in which e_i is the encoding vector of the neuron denoted by the red circle, and d_j correspond to the decoding weights for every neuron within the space as represented by their locations for $j \in [1, \ldots, n]$, in (a) the square maze from Corneil and Gerstner [2015] and (b) the watermaze.

$$\sum_{j=1}^{n} w_{i,j}^{\text{rec}} a_j(t) = (1-\varepsilon)\ \langle e_i, \hat{s}_{\text{rec}}(t)\rangle. \tag{7.26}$$

In (7.25) and (7.26), the parameter $\varepsilon \ll 1$ determines the timescale on which the network self-sustained activity fades. Figure 7.5 shows the recurrent weight profile from any neuron, *i.e.* shows the value of equation (7.25) for $j \in [1, \ldots, n]$ to a particular neuron i shown as a red circle. The emerging recurrent weight profile is locally excitatory and distantly inhibitory. For the same reason as explained in section 7.4.1.1, it is consistent with a network of units mimicking place cells.

To summarise, the initialisation of the network consists in providing inputs to the network that represent locations in the space, building decoding weights that enable one to retrieve the original input from the activity of the network. The decoding weights are used to define a recurrent input given to the network that enables it to create and maintain a steady bump profile. After the initialisation period, the recurrent term, which reflects the network's bump current location, will compete with a goal input. The next section describes the full dynamic of the network once the recurrent (equation (7.25)) and feedforward (equation (7.20)) weights have been defined.

7.4.2 Moving to the goal

In this section, the full dynamics of the network is described. The dynamics is characterised by a competition between a recurrent term, maintaining the activity onto itself, and a feedforward term, *dragging* the activity towards the goal location. After having formed a bump of activity around a start location, the goal input generates the displacement of the bump to any goal location in the network. I illustrate that the bump of activity can smoothly move towards any goal location [Corneil and Gerstner, 2015].

7.4.2.1 Competition between a recurrent term and a goal input

The dynamics of the activity of the network is defined by the balance between two input terms, which trigger the displacement of the bump of activity across space. This displacement can be interpreted as an approximation of the gradient ascent of the value function. The full description of the dynamics of the activity of the ith neuron is:

$$\tau \frac{\mathrm{d}a_i(t)}{\mathrm{d}t} = -a_i(t) + g\left[\left\langle e_i \ , \ \big((1-\varepsilon)\hat{s}^{\mathrm{rec}}(t) + \alpha \hat{s}^{\mathrm{in}}(t)\big) \right\rangle\right]_+ , \tag{7.27}$$

where α corresponds to the input strength. Here, the balance between the parameters α and ε will shape the competition between the recurrent term s^{rec} (7.26) and the feedforward inputs s^{in} expressed in the SR coordinates (7.14). By considering $\hat{s}_{\mathrm{rec}} = \sum_{j=1}^{n} d_j a_j$, the recovered activity from decoding the activities from the network, and assuming that $\hat{s}_{\mathrm{rec}} \simeq \sum_{j=1}^{n} d_j g[\langle e_j \ , \ \hat{s}^{\mathrm{rec}} \rangle]_+$ and $\hat{s}^{\mathrm{in}} \simeq \sum_{j=1}^{n} d_j g[\langle e_j \ , \ \hat{s}^{\mathrm{in}} \rangle]_+$ (heuristically, this is equivalent to assuming that the network's recovery is perfect), one arrives at the update equation:

$$\begin{aligned}
\tau \frac{\mathrm{d}\hat{s}^{\text{rec}}}{\mathrm{d}t} &= \sum_{j=1}^{n} d_j \frac{\mathrm{d}a_j}{\mathrm{d}t} \\
&= \sum_{j=1}^{n} d_j \left(-a_j(t) + g \left[\left\langle e_j \ , \ \left((1-\varepsilon)\hat{s}^{\text{rec}}(t) + \alpha \hat{s}^{\text{in}}(t) \right) \right\rangle \right]_+ \right) \\
&= -\hat{s}^{\text{rec}} + \sum_{j=1}^{n} d_j g \left[\left\langle e_j \ , \ \left((1-\varepsilon)\hat{s}^{\text{rec}}(t) + \alpha \hat{s}^{\text{in}}(t) \right) \right\rangle \right]_+ , \\
\tau \frac{\mathrm{d}\hat{s}^{\text{rec}}}{\mathrm{d}t} &\simeq \alpha \hat{s}^{\text{in}} - \varepsilon \hat{s}^{\text{rec}}. \qquad (7.28)
\end{aligned}$$

Given an input $\hat{s}^{\text{in}}$, the network representation $\hat{s}^{\text{rec}}$ approximates the input location and reinforces it, allowing a stable bump of activity to form around the input location. Once the input is moved $\hat{s}^{\text{in}}$ to a new (goal) location, the recurrent input competes with the new input location. The recurrent term acts to maintain the network activity in its current location. The new input *drags* the bump of activity towards its location, therefore taking the direction of the gradient of the value function as written in equation (7.17). One can obtain a smooth displacement of the bump from one point to another. In the next section, I show example trajectories from the model.

7.4.2.2 The bump of activity can reach any goal location and takes the shortest path

To show that the bump of activity could realise a smooth displacement from a starting location to a goal location within the space, I simulated the displacement of the bump of activity from equation (7.27) in a square environment with walls, reproducing results from Corneil and Gerstner [2015], and I adapted it to the watermaze-like environment, both described in figure 7.1. The decoding weights (7.23) were computed from the steady-state activities from $m = 50$ inputs uniformly sampled from the environment.

When the input given to the network changes from the start location to the goal location, the bump of activity moves smoothly towards it. Figure 7.6 shows the evolution of the activity of the network with time. The activity is shown using a heatmap across the space. The network was initially stimulated from equation (7.26), forming a bump of activity around a starting location (grey circle in figure 7.6). After a short delay ($\approx 3\tau$), $\hat{s}_{in}$ is changed to a new input representing the goal location (red circle in figure 7.6). At this point, the input $\hat{s}_{in}$ and recovered coordinates from the network $\hat{s}_{rec}$ compete. As the approximation through the recovered location (7.28) suggests, the bump of activity starts moving towards the goal location. In the circular environment mimicking the watermaze, the bump of activity then follows a smooth trajectory to the goal location (Figure 7.6a). Moreover, the bump in a square environment with walls as used in Corneil and Gerstner [2015] moves smoothly from a start location to a goal location taking the shortest path (Figure 7.6b). The details of the implementations and choices of parameters are given in Appendix D.3. In particular, the complexity of adapting the model to a novel environment emerges when defining the graph of points and the parameters scaling the connection within the graph, in particular the value of σ in equation (7.2).

7.5 Influence of the dimensionality of representation and relative weight of the first eigenvector

In this section, I extend the work from Corneil and Gerstner [2015] to investigate the influence of the dimensionality of the network and the relative weight of the first eigenvector on the dynamics of the network. First, I show that the dimensionality q of the network influences the precision of the representation and the smoothness of the trajectory between start and goal location. Then, I discuss the effect of the choice of the relative weight of the first eigenvector w_1 in the definition of the new basis (7.14). I show that the choice of scale of the first eigenvector affects the global level of activity within the network, the precision of the bump of activity formed in the network, and its trajectory to a goal location.

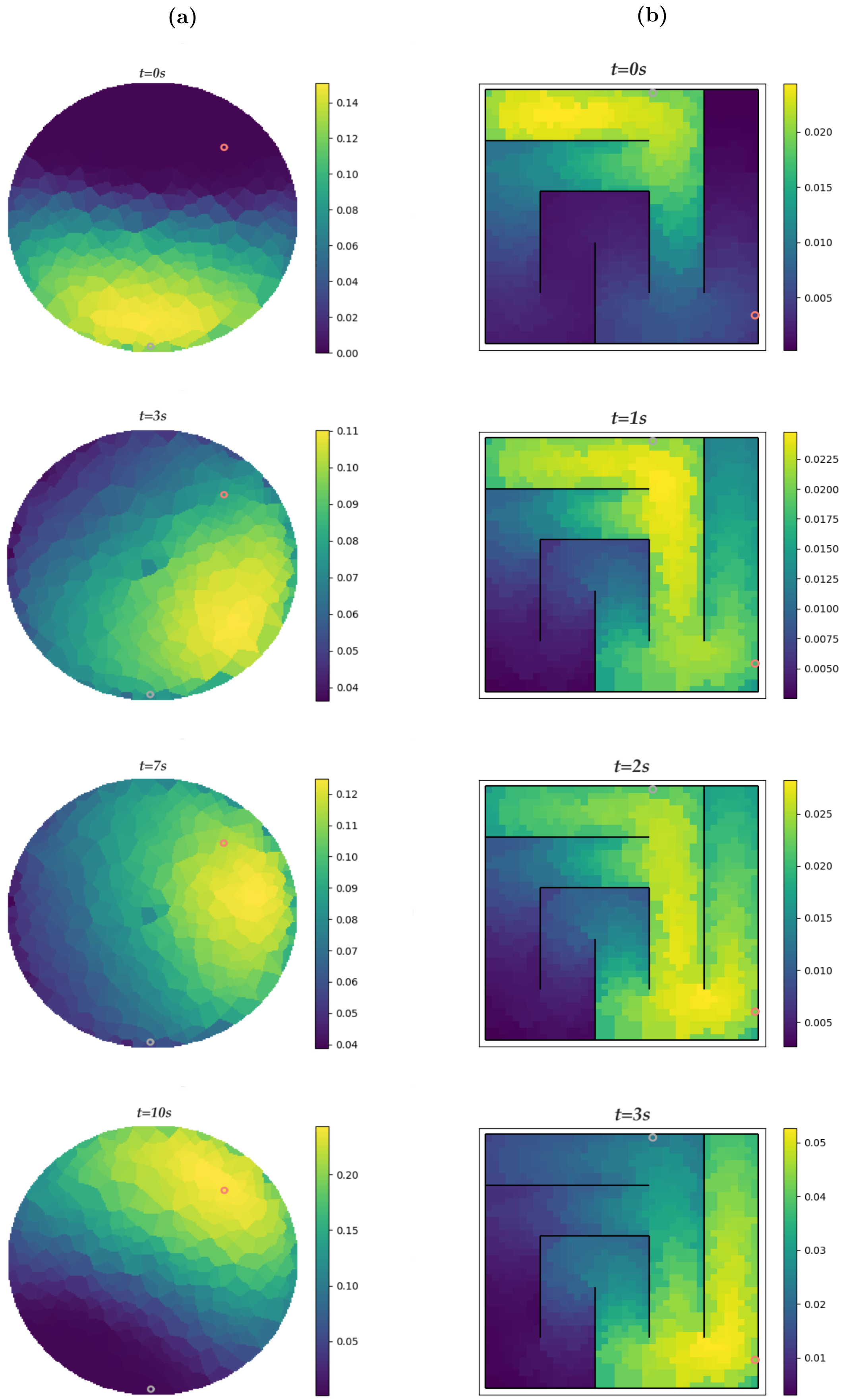


Figure 7.6: Displacement of the activity profile from equation (7.28) for $q = 5$ and a $w_1 = \sqrt{1.2 * (1 - \gamma\lambda_2)^{-1}}$. (a) In the watermaze environment, the bump of activity moves from the start location (grey circle) to the goal location (red circle). (b) In a square environment with walls used in [Corneil and Gerstner, 2015], the bump follows walls and reaches the goal using the shortest path.

7.5.1 Effect of the dimensionality q on the precision and reach of the trajectory

In this section, I show that the dimensionality q influences the precision of the spatial representation and the spatial reach of the displacement of the bump. Section 7.2.2 described how the eigenvectors provide a basis of the space which resembles a Fourier decomposition (figure 7.3). As the first eigenvectors represent lower spatial frequencies, choosing q to be small involves a representation of the states with low spatial resolution. This can be seen in figure 7.7, which shows the strength of the input received from all states of the network to a particular unit of the network (marked as a red circle). The strength of the input is given by the dot product $\langle e_i, \hat{s}^{\text{in}}(t) \rangle$ between the states, which represent the inputs, *i.e.* $\hat{s}^{\text{in}} = \hat{s}_j / ||\hat{s}_j||$, with $j \in [1, \ldots, N]$, and e_i, the encoding vector of the unit represented as a red circle, given in equation (7.19).

A low dimensionality generates very broad recurrent and feedforward weights, and a high value of q generates very localised recurrent and feedforward weights. Figure 7.7a shows the input strength for $q = 2$. The state space is then a 2 dimensional subspace generated by ξ_1, ξ_2 from the basis (7.15). As the shape of the eigenvectors in figure 7.3 suggests, the coordinates are changing with low frequency throughout the space. This can be seen in figure 7.7a, as the width of the input strength is very wide. Moreover, it is not localised around the unit's location. For $q = 40$, in figure 7.7b, the strength of the input is narrower and localised around the unit's centre. Figure 7.8 shows the value of the incoming recurrent weights $w_{i,j}^{\text{rec}}$ to the neuron i at the centre of the red circle. $w_{i,j}^{\text{rec}}$ is computed from equation (7.25) in which e_i is the encoding vector of the neuron denoted by the red circle, and d_j correspond to the decoding weights for every neuron within the space as represented by their locations for $j \in [1, \ldots, n]$. The strength of the input and the recurrent weights are similarly influenced by the value of q. For $q = 2$ (figure 7.8a), the recurrent weight profile is wider than for a high value of q (figure 7.8b). This shows that a unit in a low dimensional space is more responsive to activity of neurons and to any input further away than in a high dimensional space. For a low value of q, the spatial precision is very low. For $q = 40$, the spatial precision is higher, but a unit does not respond to inputs too far away.

The recurrent weights and strength of input profiles then affect the dynamics of the displacement of the bump of activity once the input changes to the goal state, shown figure 7.9. For a low value of q, in figure 7.9a, the bump of activity formed during the initialisation of the network activity is not centred around the start location. Moreover, when the input changes to represent the goal location, the bump only slightly adjusts. This is because, even though quite far apart in the maze, the start and goal locations have too similar encoding vectors, and the dimension of the network does not allow these inputs to be differentiated. Contrarily, for a high value of q, in figure 7.9b, the bump of activity formed within the network is very localised and centred around the start location. When the input changes to the goal location, the bump of activity does not take a direct path to the goal: it progressively fades out at the start location while emerging at the goal location. As the recurrent weights (figure 7.8b) are very localised, neurons active very far away from the starting location will not affect its activity. Similarly, the strength of the input (figure 7.7b) is very localised, therefore triggers an increase in activity only locally around the goal location. Therefore, the input at the goal location does not drag the network activity smoothly towards it, but only becomes the new centre of the activity.

To summarise, the high dimensional representation is more spatially precise but does not allow the generation of a smooth trajectory to a distant goal location. Moreover, the computations associated with higher dimensions are more costly: it requires more storage capacity and more computations. In contrast, predictions in a low dimensional space are easier, it requires less storage capacity, but are not spatially precise. Therefore the optimal q should be chosen as a trade-off between precision and computational efficiency.

7.5.2 Effect of the scale of the first eigenvector on the global activity level and the trajectory

In this section, I show that the relative scale of the first eigenvector in the representation influences the global level of activity within the network and the trajectory of the bump of activity. The weight associated to the first eigenvector, w_1, replaces the value $\sqrt{(1-\lambda_1)^{-1}}$

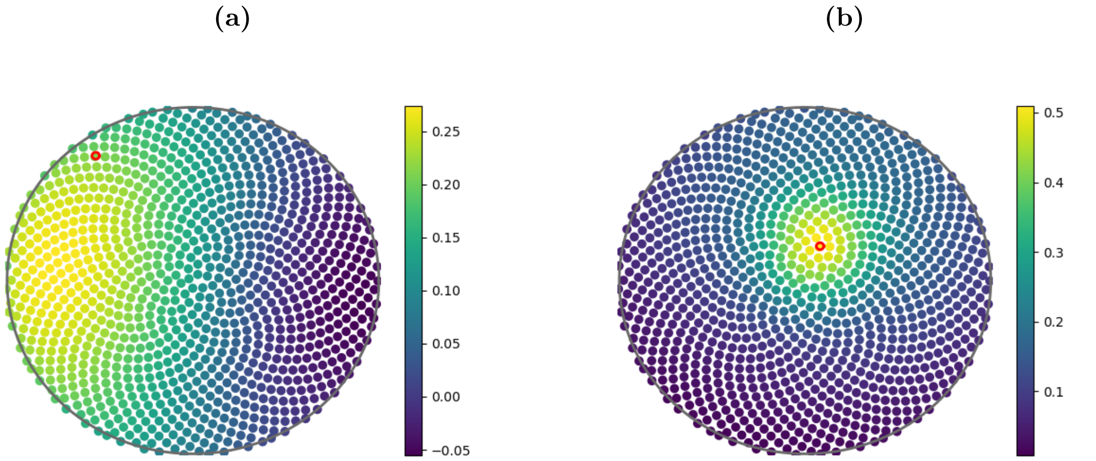


Figure 7.7: Effect of the dimension q on the strength of the input received by a unit of the network (shown as a red circle). As in figure 7.4, the plot shows the value of the dot product $\langle e_i, \hat{s}^{\text{in}}(t)\rangle$, the right hand side of equation (7.22), as if the inputs $\hat{s}^{\text{in}}$ were located at the state locations marked by dots, *i.e.* $\hat{s}^{\text{in}} = \hat{s}_j/||\hat{s}_j||$, with $j \in [1, \dots, N]$, and e_i the encoding vector of the unit represented as a red circle, given in equation (7.19). Yellow colours represent higher values and blue colours lower values. (a) For a low value of q, here $q = 2$, the strength of the input is not localised on the neuron anymore, and is widely spread across the environment. (b) For high value of q, here $q = 40$, the strength of the input is narrower and more localised around the unit's centre.

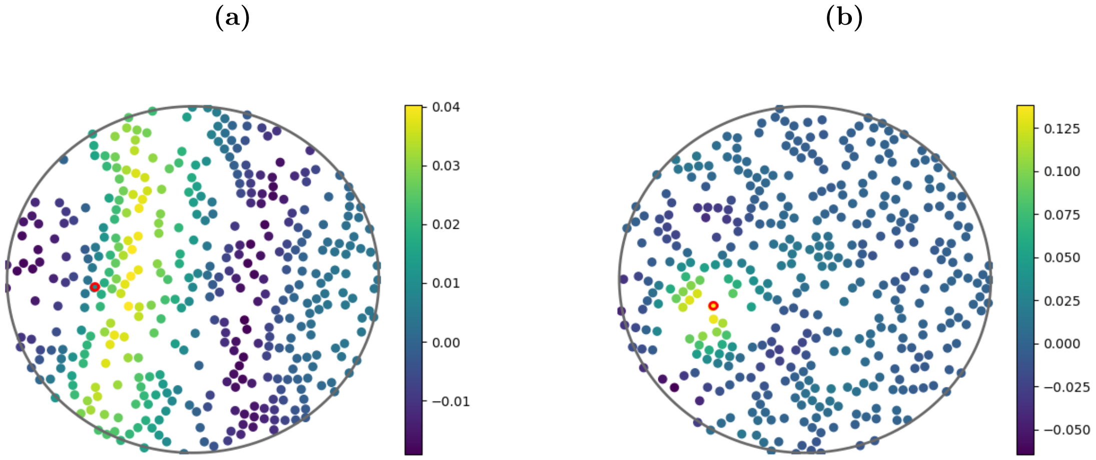


Figure 7.8: Effect of the dimension q on the recurrent weight profile. Value of the recurrent weights $w_{i,j}^{\text{rec}}$ from all the neurons j to the neuron at the center of the red circle, computed from equation (7.25) in which e_i is the encoding vector of the neuron denoted by the red circle, and d_j correspond to the decoding weights for every neuron within the space as represented by their locations for $j \in [1, \dots, n]$. Yellow colours represent higher values and blue colours lower values. (a) For a low value of q, here $q = 2$, units are equally sensitive to activity of other units within a wide band of resolution. (b) For a high value of q, here $q = 40$, the sensitivity of units is very spatially localised.

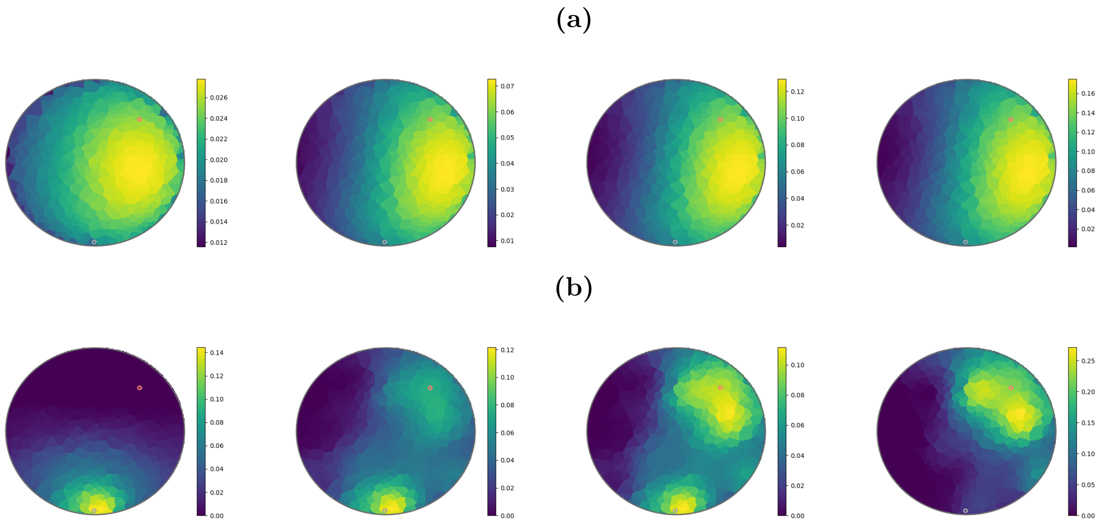


Figure 7.9: Effect of the dimension q on the width and displacement of the bump of activity. Yellow colours represent higher values and blue colours lower values. (a) For a low value of q, the bump of activity formed during the initialisation of the network activity around the start location is not centred around the start location. Moreover, when the input changes to represent now the goal location, the bump only slightly adjust, but stays in its main location. (b) For a high value of q, the bump "jumps" from start to goal location.

in the basis expressed in equation (7.13). The first eigenvector is uniform across the environment and therefore w_1 scales the degree of uniformity of the representation in the network. This parameter, via influencing both the feedforward (7.20) and recurrent weights (7.25), acts as an activity gain uniform across the network, because the first eigenvector is uniform across the environment. In figure 7.10, one can see that for a high value of w_1 (figure 7.10b), the strength of the input is slightly more uniform around the environment than for a low value of w_1 (figure 7.10a) and the global strength of the input scales with w_1. In the recurrent weight profile across the environment (figure 7.11), there is no obvious difference between the recurrent weight profiles for a low value of w_1 (figure 7.11a) and a high value of w_1 (figure 7.11b) in terms of spatial scale and baseline value. This comes from the definition of the recurrent weights (7.25), as the dot product between the decoded weights (7.24) and the units encoding vector (7.19). As the decoding weights are defined to minimise the difference between the location of an input and the decoded location of the input from the activity of the network, and because the first coordinate is uniform across the environment, it should not play a role in the separation of the locations in the linear readout. Therefore, the choice of w_1 affects the strength of the input, but not the recurrent term.

w_1 influences the trajectory of the bump of activity. Figure 7.12 shows the effect of the value of w_1 on the evolution of the bump of activity from a starting location to goal location across the environment. For a low value of w_1, the bump of activity is narrower and tends to jump from start to goal location when the input changes (figure 7.12a). For a high value of w_1, more weight is put on the uniform eigenvector of the basis, leading to a wider bump of activity. When the input changes, the bump of activity takes a smooth trajectory towards the goal. Because there is more weight on the uniform eigenvector, the trajectory of the bump is slightly more direct towards the goal location compared to an intermediate value of w_1 given in figure 7.6. Interestingly, for a low value of w_1, one can see that the global level of activity in the network reduces, and contrarily, for a high value of w_1, the global activity level of the network increases.

To summarise, the strength of the uniform eigenvector in the basis shapes the spatial precision of the bump of activity within the network, the trajectory of the bump between the start and goal location, and tunes the global activity level across the network.

In this section, I showed that the dimension of the space needs to be chosen depending on a trade-off between precision and computational efficiency. In a high dimensional space, the loss of information is minimal, but computations are more expensive. A lower dimensional space enables cheaper computation and efficient memory storage, but at the cost of lower spatial precision. The weight of the scaling of the uniform eigenvector w_1 should be chosen relative to the other values $\sqrt{1 - \gamma\lambda_j}$ for $j \in [2, \ldots, q]$ associated to the other eigenvectors in (7.14). w_1 affects the precision of the bump of activity, the trajectory of the bump, and the global level of activity within the network: the strength of the input received by the network is higher, and the bump is wider, therefore, the apparent trajectory in an open field is more direct, and the global activity is higher.

7.6 Conclusions

Extending a spatial bump attractor model adapted from Corneil and Gerstner [2015], generalised to environments with arbitrary obstacles and adapted to open field environments

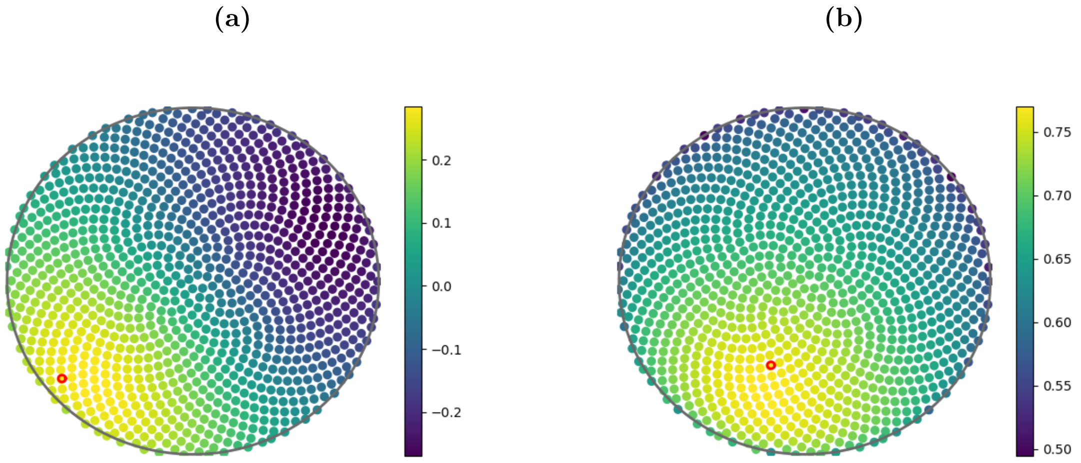


Figure 7.10: Effect of the weight w_1 of the first eigenvector on the strength of the input received by a neuron. The plot shows the value of the dot product $\langle e_i, \hat{s}^{\text{in}}(t) \rangle$, the right hand side of equation (7.22), as if the inputs $\hat{s}^{\text{in}}$ were located at the state locations marked by dots, *i.e.* $\hat{s}^{\text{in}} = \hat{s}_j/||\hat{s}_j||$, with $j = [1, \ldots, N]$, and e_i the encoding vector of the unit represented as a red circle, given in equation (7.19). Yellow colours represent higher values and blue colours lower values. (a) For a low value of w_1, the strength of the input is slightly more localised than for a high value of w_1 (b), in which the strength of the input is more spread around the environment. A major difference as well is that the global strength of input is much higher for higher value of w_1 then for lower values, which can be seen through comparing the values of the colour bars in (a) and (b).

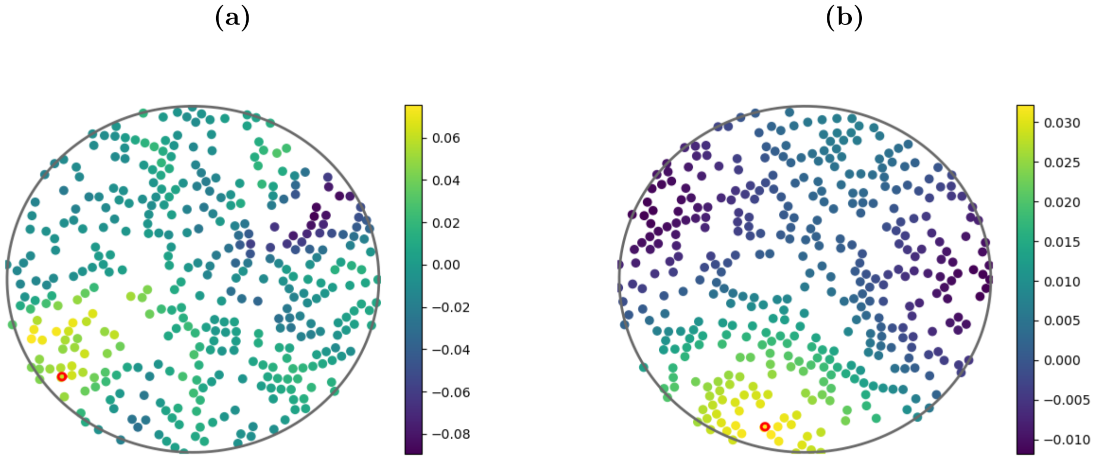
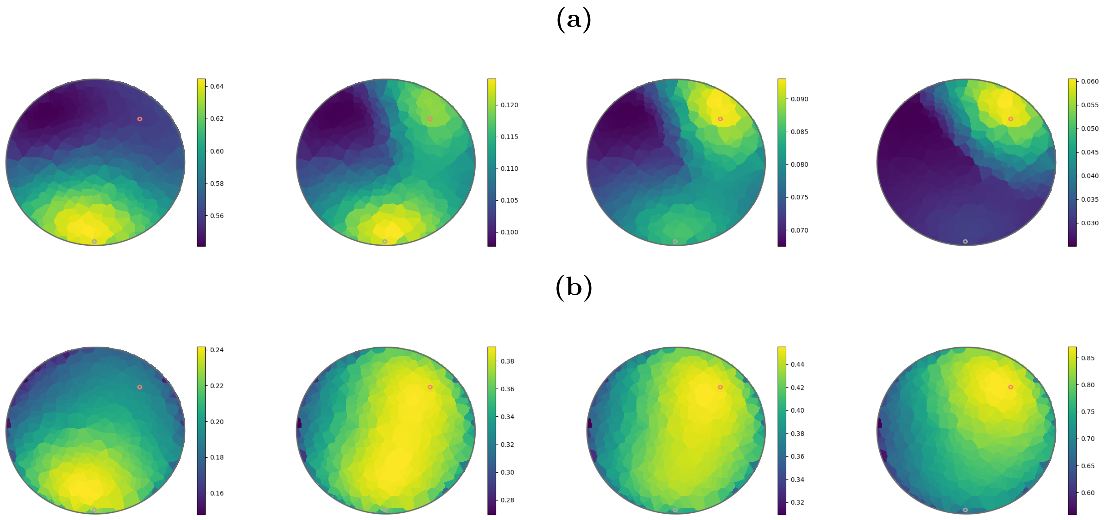


Figure 7.11: Effect of the weight w_1 of the first eigenvector on the recurrent weight profile.Value of the recurrent weights $w_{i,j}^{\text{rec}}$ from all the neurons to the neuron at the center of the red circle, to all the neurons within the environment, computed from equation (7.25) in which e_i is the encoding vector of the neuron denoted by the red circle, and d_j correspond to the decoding weights for every neuron within the space as represented by their locations for $j \in [1, \ldots, n]$. Yellow colours represent higher values and blue colours lower values. (a) for a low value of w_1, the spatial frequency of the recurrent weight profile is slightly higher than for higher values of w_1 (b).


(a)

(b)

Figure 7.12: Effect of the weight w_1 of the first eigenvector on the width and displacement of the bump of activity. Yellow colours represent higher values and blue colours lower values. (a) For a low value of w_1, the bump of activity is narrower, and tends to jump from start to goal location when the input changes. (b) For a high value of w_1, more weight is put on the uniform eigenvector of the basis, this leads to a wider bump of activity. When the input changes, the bump of activity takes a smooth trajectory towards the goal. Because there is more weight on the uniform eigenvector, the trajectory of the bump is more direct towards to goal location, compared to an intermediate value of w_1 given in figure 7.6.

(figure 7.1), I have shown that predictive representations as seen in place cell activity profiles can be used to efficiently plan trajectories towards any goal location. In section 7.2, the environments have been discretised into states. The states can be seen as locations in the 2D space, but also as nodes of a graph. A transition probability matrix between states was defined from an affinity kernel which depends on the shortest path between two states. Results from graph theory [Coifman and Lafon, 2006] have led to a new basis of the state space formed by the eigenvectors of the transition probability matrix. A recurrent network is defined in which units are points in the space generated by the eigenbasis.

The connectivity between units obtained mimics what is expected of the connectivity between hippocampal place cells [Káli and Dayan, 2000]: units which have similar "preferred" locations excite each other. Feedforward weights in the network are based on the similarity between the SR coordinates of an input and those of the units of the network (figure 7.4), and recurrent weights between units enable the network to maintain its own activity (figure 7.5).

In this recurrent network, goal locations provided as input to the network define attractor states of the network activity [Corneil and Gerstner, 2015]. The recurrent input guarantees the formation of a bump of activity when the start location is provided as input to the network, which, when the input switches to the goal location, takes a smooth trajectory towards the goal.

The SR representation offers an advantageous opportunity for prediction by making seemingly complex calculations over future spatial states computationally efficient. Indeed, in the new SR basis, the value function becomes a simple dot product. Therefore, the computation of the gradient of the value function in this space is simple and efficient. Remarkably, the displacement of the bump of activity from a start to goal location can be interpreted as a gradient ascent procedure of the value function [Corneil and Gerstner, 2015].

Previous work has linked place cell activity profiles to a representation of the environment based on its natural topology. For example, Gustafson and Daw [2011] link the firing profile of place cells and grid cells to the geodesic distance between locations, and Stachenfeld et al. [2017] illustrated that hippocampal place cells have predictive components similar to SR representations. By looking at the changes in activity profiles in response to the introduction of new boundaries or in response to changes in the geometry of the environment, Stachenfeld et al. [2017] showed that the adjustment of their place field observed in recordings from previous experimental works match the ones obtained from the SR. Here, SR place representations have activity profiles that adjust to the geometry of the environment, and the displacement of the activity within the network corresponds to trajectories to goal locations using the shortest path. Moreover, as described in section 2.3.2, preplay activities have been recorded in hippocampal place cells towards goal locations [Foster, 2017, Pfeiffer and Foster, 2013]. This model suggests that goal attractor states within an SR recurrent network could lead to the generation of such preplay trajectories [Corneil and Gerstner, 2015].

Further investigations of the model have exposed how memories in time and space could be stored in a low dimensional state space for efficient computations and memory storage. In this chapter, the units mimicking place cells have a firing profile that provides an encoding of spatial locations in time and space. The eigenvectors, assigned in the order of decreasing magnitude their associated eigenvalue, reflect diffusion across states at different timescales. Cutting the dimensionality of the state is equivalent to keeping the largest timescales of representation of state occupancy. In this chapter, I showed that cutting the dimensionality of the space in which the network is embedded impairs the spatial precision but increases the spatial reach of the trajectory. Therefore, the model predicts that a precise path to the goal location can only be generated to acute goals which are close in time and space. It predicts that the precision of a prediction is inversely proportional to the timescale at which the prediction is made.

This work relates to an approach originally proposed by Coifman and Lafon [2006], which used diffusion processes within a graph to find a meaningful geometric description of data sets. Coifman and Lafon [2006] used the eigendecomposition of the transition matrices within a graph to study the structure of the graph at different timescales. They used it to investigate the multiscale geometries within data sets that they used for dimensionality reduction and efficient storage. In the environments considered here, the eigenvectors also permit to compartmentalise the space into areas of similar future state occupancy, and provide an efficient representation of space and its travel statistics. This approach, when applied to more complex graphs which have bottleneck states for example, can lead to very fast identification of these bottleneck states [Machado et al., 2018] and more efficient navigation [Ramesh et al., 2019]. If subgoals can be seen as bottleneck states, SR representations could be useful to plan subgoals [Ramesh et al., 2019].

Previous work on the SR [Stachenfeld et al., 2014] already linked the normalised graph Laplacian to the SR in a spatial context. A recent study proposes that hippocampal place representations code Laplace transforms of functions of past spatio-temporal events [Howard and Hasselmo, 2020]. Laplace transforms represent a function in its frequency domain, and in the Laplace domain, integrals and differential equations become algebraic equations, phenomenon also observed here in the value function which becomes a simple dot product in the SR coordinates. The idea that brains form efficient representations of past experience is quite intuitive, and Laplace representations enable efficient representation of time evolution, and this chapter has made a link between efficient time representation and predictions.

In this chapter, I assumed that the topology of the environment as encoded in the transition matrix P is known. However, it is unknown how the brain would form such neural representation. In section 3.2.4, I discussed how the SR successfully trades off flexibility and computational cost. However, in this approach, the step of extracting the eigenvalues and eigenvectors is computationally expensive. Using generalised Hebbian Algorithm, or Sanger's rule, a generalisation of Oja's rule, could provide a biologically realistic approach to learning the eigenvectors of the transition probability matrix [Olshausen, 1998]. Previous work has suggested that place and grid cell profiles could be obtained using slow feature analysis [Franzius et al., 2007, Schönfeld and Wiskott, 2015]. Slow feature analysis consists in extracting the parts of a temporal signal that minimises the temporal variance of a stimulus. So, SR representations could emerge from highly processed visual information extracted from higher order visual areas [Stachenfeld et al., 2017, Corneil and Gerstner, 2015]. Other possibilities, such as learning the SR using TD methods, as briefly introduced in section 3.2.4, could be a possible extension to this approach for the model to be more self-contained. Although presented here in discrete state space as a matrix, the SR can be generalised to continuous space using successor features [Barreto et al., 2017].

The model highlights a link between precision of representation and spatial reach of the planning. One can assume that the further one can project in time, the more imprecise the prediction about the exact trajectory could be. But projection far in time enables one to generate trajectories to distal locations in one pass. Therefore, it is more computationally efficient, as only one planning phase is necessary to reach further locations. This is nicely captured in the model presented here, in which the cost of time projection is a loss of precision over the bump width.

As planning a trajectory in an open field, such as the watermaze, depends on distal visual cues, the precision is limited by the errors made on the estimation of the relative positions of the cues. In section 4.3.1.1, I have related the spatial scale of place cell representation to the generation of optimal trajectories. I showed that wider spatial activity profiles lead to faster learning, and smaller ones lead to more precise trajectories [Scleidorovich et al., 2020]. In this model, I showed that wider place activities have the potential to enable further planning, and smaller place activities to obtain a precise planned trajectory towards close goal locations. This offers a coherent picture of the link between planning, scale of representation, and performance.

Further experiments targeting the intermediate hippocampus during replay episode could enable to link scale of representation, replay and performance in the watermaze. As discussed in section 4.3.1.1, the intermediate hippocampus is necessary for one-shot learning in the Morris watermaze [Seaton, 2019]. Experiments targeting particularly the ventral, intermediate, or dorsal hippocampus while recording replay phenomena are lacking, but it could be interesting to look at the differences observed in replay patterns in different scales of tasks between the dorsal, in which the spatial representations are smaller, intermediate, and ventral, with wider spatial scales. In particular, it could be interesting to investigate whether, in the DMP task, replay phenomena are observed in the intermediate hippocampus, and whether these are linked with performance.

To conclude, the model presented here enables one to form a representation of the space which contains information about future visit expectancies and to generate a bump of activity within a network that can take smooth trajectories to goal locations. The spatial reach of the network depends on the precision of the bump, more precise trajectories have only a limited spatial reach. The model can be used to find subgoals and generate planning trajectories to the nearest subgoals. Interestingly, reducing the precision of the bump also involves reducing the dimension of the space in which the network is defined, therefore saving storage capacity and gaining computational efficiency. It is reasonable to assume that, in spatial navigation, being able to form very precise spatial trajectories to far away goal locations is perhaps not necessary if one is able to find subgoals and plan trajectories to the closest subgoals. It is perhaps more crucial to have an efficient representation of space that permits planning with minimal computations, in order to encompass the wholeness of spatial navigation situations that an animal goes through in its lifetime.

Chapter 8

Conclusions and future work

8.1 Recapitulation of the work

The goal of this thesis was to develop RL approaches for flexible hippocampal-dependent spatial navigation motivated by one-shot learning in the Morris watermaze.

After an overview of the thesis, I began in chapter 2 by providing background information regarding experimental results in behavioural sciences and spatial navigation. I discussed 'flexibility', the ability to adapt quickly to new situations. Flexible spatial learning in rodents and humans can be assessed in the Morris watermaze and its virtual equivalent by regular changes of goal locations [Morris, 1981, Steele and Morris, 1999, Buckley and Bast, 2018]. Hippocampal neural representations include diverse spatial correlates [Moser et al., 2008, Jeffery, 2018], and past studies have revealed that the hippocampus is critical for one-shot spatial learning in the Morris watermaze [Bast et al., 2009, Bast, 2007]. This raises many questions about the mechanisms that connect neural representations to the production of behaviours.

In chapter 3, I reviewed the fundamentals of RL, highlighting that the RL framework enables the investigation of computations underlying the generation of reward-based behaviours [Sutton and Barto, 2018]. RL can thus be used to link spatial representations with action selection for navigation. However, past RL approaches are either inefficient in terms of computational cost, or not flexible to changes in the environment, such as changes in goal locations. I discuss the trade-off between flexibility and efficiency of predominant RL methods, in which each system achieves a different balance depending on the purpose it is built to achieve.

I began chapter 4 by discussing an actor-critic agent, that can learn to navigate to a fixed goal location in a Morris watermaze task. The principal pitfall of a simple actor-critic architecture is exposed by its inability to reproduce flexibility demonstrated by animals and humans in watermaze tasks to changes in goal locations. A simple extension of the architecture, using a map of the space linked with a generalised goal-directed action, enables the agent to adapt to changes in goal locations [Foster et al., 2000]. Further than implementing the original approach proposed by Frémaux et al. [2013], I additionally discussed its biological realism in light of recent experimental findings and discussed the role of key parameters in shaping the learning of the agents. By assessing the search preference, a metric commonly used in watermaze tasks, I showed that the agent's trajectory is persistent around goal locations, supporting its behavioural realism. I showed how the computations involved in the production of the agent's behaviour are partly consistent with experimental findings assigning an actor and critic roles to the striatum, linked with the hippocampus as a basis for spatial representations, and using dopaminergic signals as reward prediction errors for learning. In particular, striatum, hippocampo-striatal projections, and dopamine are involved in rapid place learning in the watermaze. Moreover, the scale of spatial representations is linked with performance, an observation characterised by a trade-off between speed and precision. This result is consistent with previous computational results [Scleidorovich et al., 2020] and experimental evidence implicating different sub-regions of the hippocampus, differentiated by their spatial scales [Kjelstrup et al., 2008], for different purposes [Bast et al., 2009, Moser et al., 1995]. The use of a generalised goal-directed action is consistent with goal and goal-vector representations found in diverse areas of the brain, in particular in the hippocampus. However, dopamine acts on other plasticity mechanisms to those proposed in the model, and new goal information should be incorporated within hippocampal representations.

In chapter 5, I discussed how continuous action representations enable faster learning and smoother control of the direction. A mathematical framework that features continuous time representations is consistent with the diverse timescales of natural behaviours [Doya, 2000], therefore improving the biological realism of RL approaches. I adapted a spike-based approach for spatial navigation [Frémaux et al., 2013] to a continuous rate model for spatial navigation in the Morris watermaze. Although using continuous action and time representations in an actor-critic architecture for navigation in the watermaze generated smoother control of the trajectories, I did not observe faster learning when comparing its performance to that of an actor-critic architecture with discrete action and time representations. This could be due to the fact that the learning dynamics in the actor-critic method are slow and do not benefit from the improved temporal resolution of actions and time provided by continuous representations in the example of navigating in the watermaze. Further, I compared two RL approaches featuring continuous representations of states, actions and time: one featuring an encoding via rates, the other encoding via spikes. In the context of the watermaze task, I observed that their performances are comparable, but a continuous rate approach is computationally cheaper than a spiking rate implementation. Although fast and high dimensional information transmission in spiking networks seems beneficial, it has to my knowledge not been linked to advantages in the production of behaviour within an RL framework. Spike coding has not been related to flexibility in watermaze experiments, however, it could be part of general flexibility mechanisms useful for spatial navigation in its diverse forms by providing high dimensional and fast state-action categorisation.

Given that general representations and transfer of knowledge seem to be key points underlying efficiency, in chapter 6 I proposed a new model for flexibility in the watermaze which incorporates ideas from hierarchical reinforcement learning and meta-learning. In hierarchical learning, tasks are hierarchically organised, and actions are abstracted in order to be transferable to serve different purposes. Meta-learning features parameters update at multiple timescales to adjust to fluctuating changes in the task environment. By combining the two, I propose a hierarchical model that uses the long term suitability of the current policy for the current goal to shape the agent's behaviour using meta-learning mechanisms. I could show that the agent successfully adapts to changes in goal locations, and I discussed that the model is biologically realistic. The meta-computations performed for flexibility are consistent with experimental results involving prefrontal areas and their interaction with the striatum and the dopaminergic system in spatial navigation. My approach suggests that hierarchical representations of actions in the striatum could underlie flexibility in the watermaze.

Flexibility in spatial navigation requires efficient planning capabilities in order to adjust to new situations. Chapter 7 explores how predictive information, as found in the tuning of spatial representations, can shape goal-directed trajectories. I considered an artificial recurrent neural network in which units have predictive representations based on the SR and goals are attractor states [Corneil and Gerstner, 2015]. Here, the predictive representation reflects diffusion across space at multiple timescales. I confirmed the results of Corneil and Gerstner [2015] showing that the model produces an activity which, when transferred back to the equivalent 2D space mimicking the maze environment from which the network was defined, can settle around a particular location in space and move smoothly towards goal locations. The model provides a basis of representation in which computations are efficient and perfectly suited for prediction. Along with reproducing the results already found in a maze with arbitrary walls and obstacles [Corneil and Gerstner, 2015], I showed that the model also produces direct trajectories from any start location to any goal location in an open field environment mimicking the watermaze. Further studying this network, I have identified a link between the precision of spatial representation, the reach of the generated trajectories and the cost of computations: the model predicts that the further away the locations are, the less precise the planning can be, and that precise trajectories are computationally costly.

To conclude, I showed that flexibility in spatial navigation demands general representations of space, actions, and goals, which enable transfer between experiences; a hierarchical organisation of the task, which enables different levels of control to be performed simultaneously and facilitates transfer; and predictive representations, which permit efficient planning. In the next section, I describe ideas for future development.

8.2 Future development

The work presented here draws together a number of themes pertaining to computational mechanisms underlying flexibility in spatial navigation, and also suggests exciting opportunities for future development. I presented RL approaches that enable an agent to be more flexible in spatial navigation. However, all of them have shortcomings, and in this section I suggest extensions that address these. Additionally, based on the models' predictions, I suggest further experimental studies.

In chapter 4, section 4.4.3.2, I discussed how a uniform representation of locations throughout the space used to form a coordinate-action which computes a goal-directed vector displacement leads to direct trajectories to any goal location. However, this approach uses Cartesian coordinates which is consistent only in open field environments, but is not mappable to any environment. In the SR attractor network described in chapter 7, the activity also reaches any goal location, and in diverse environments, open fields or with arbitrary obstacles. Therefore, using SR representations to perform goal-directed displacement seems more biologically plausible.

The dynamics of the SR network can be used to inform a goal-directed action. As the bump of activity of the network takes the direction towards the input provided to the network, comparing the activities of the network on a short interval after the goal input has been presented would enable the formation of a goal-directed vector. Adapting population models of temporal differentiation, for example as suggested by Tripp and Eliasmith [2010], to differentiate the response of the network after a change of input, would enable the formation of a goal-directed vector. Using a population that would be sensitive to this difference, for example using bisynaptic inputs; in which the units of the 'differential network' receive information about the state of the SR network from two synapses delayed by a short time interval, could lead to the encoding of goal-directed vectors within the new population [Tripp and Eliasmith, 2010]. The cells considered in the 'differential network' should therefore be sensitive to both the location of the bump and a particular heading of the bump, which seems consistent with recordings of neurons in the subiculum [Cacucci et al., 2004] and in the entorhinal cortex [Sargolini et al., 2006] that encode the position and direction of the animal. Therefore, adapting the goal-directed navigation proposed in section 4.4.3.2 with the use of coordinates based on the SR presented in chapter 7 would enable one to perform goal-directed navigation based on SR coordinates in a biologically realistic scenario. Moreover, the discussion in section 7.6 on the link between the precision of the bump of activity, the timescale of prediction, and the dimension of representation, adapted to the population differentiation methods, could lead to predictions on the resolution and precision of the firing profiles of neurons sensitive to heading vectors.

In the hierarchical model presented in chapter 6, the selection of the strategy could be performed using a Bayesian framework [Bernardo and Smith, 2009]. In the presented approach, the selection of the strategy takes place at the start of a trial depending on the goal prediction errors (6.1). The selection of a strategy could be dynamically performed using a posterior probability distribution over the strategies given the goal prediction errors. One could then assess the degree of confidence of the agent directly from the posterior probability distribution. The actor network activity (4.3) used to compute the policy could be determined from the sum over strategies, weighted by their relative posterior probabilities. The search trajectories resulting from a negative goal prediction error would then lead the agent to explore other strategies rather than using a random walk. This could be consistent with search trajectories in rodents that tend to return to previous goal locations when facing a misprediction in goal location [Steele and Morris, 1999, Pearce et al., 1998].

Similar to the bayesian extension of the hierarchical architecture described above, the continuous actor network (5.12) considered in chapter 5 can be extended to incorporate different strategy inputs. As its resulting heading direction is more precise than using the discrete actor network originally used in the hierarchical agent, it would display a larger degree of sensitivity to disagreeing inputs, as would be the case when strategies conflict. The trajectories obtained from the model could be compared to rodents behaviours to investigate the relative weighting of strategies that drive their behaviours.

Animal search trajectories are efficient (figure 2.10). Such realistic search trajectories could be obtained by using the SR as a default search policy. The noise term in the continuous actor (5.12) network leading the exploration of the agent could be replaced by an SR-weighted policy, which influences exploration in the directions of most likely scenarios. The agent would then explore more of the areas of the space with higher predicted future state occupancy, leading to efficient search trajectories.

The different proposed approaches all contain different limitations and are not mutually exclusive but rather complementary. A merged model consisting of continuous and hierarchical action selection and SR-based representations of space could enable a robust scheme for flexibility. As all approaches rely on different mechanisms, one could investigate the interactions of these solutions in producing flexibility. Moreover, these mechanisms all reflect different candidates neural pathways: the hippocampus containing SR representations [Stachenfeld et al., 2017], action selection in the striatum [Kimchi and Laubach, 2009], and hierarchical control within prefrontal and striatal pathways [Rusu and Pennartz, 2020]. Understanding how the individual model's components balance out in producing trajectories could help to shed light on how the individual brain regions coordinate in the production of natural navigation behaviours.

In section 6.3.3.1, I highlighted that the principal limitation of the hierarchical model to account for flexibility in the watermaze DMP task is that it requires pre-training and cannot incorporate new goal locations to its strategy space. I propose to adapt computational approaches of memory updating [Gershman et al., 2014b, Blumenfeld et al., 2006] in order to form a new strategy when a novel goal location is discovered. Both studies use the strength of a novelty signal to form a new memory node, which could be adapted to create a new goal location in the context of spatial navigation, for example using a goal prediction error.

In this thesis, I have not performed a fit of the models to animals' behaviours. Lesions, or pharmacological manipulations, lead to the generation of specific behavioural deficits. Fitting models to behaviours of control and manipulated animals and comparing the two could shed light on the functional impairment resulting from those manipulations. For example, in a DMP task variant in which rats would be pretrained to familiar goal locations, fitting the hierarchical model to the behaviour of prefrontal lesioned and control rats could reveal an insensitivity of lesioned animals to goal prediction errors.

In addition to behavioural predictions, the work presented here also makes predictions about neural activities. The model analysed in chapter 7 leads to predictions about hippocampal preplay activities during spatial navigation. The model predicts a relationship between the spatial resolution of an SR place cell and the spatial reach of the planned trajectory. As the dorsoventral axis of the hippocampus is characterised by place cell activity profiles of increasing widths [Kjelstrup et al., 2008], the model predicts that cells along the dorsoventral axis are increasingly likely to participate in preplay events to locations of increasing distance.

Moving towards broader perspectives beyond spatial navigation, the approaches and discussions presented here can also be of benefit to other areas of research. A burgeoning area of great interest of mine, computational psychiatry, investigates the computations underlying certain psychiatric disorders such as depression or anxiety [Montague et al., 2012]. In section 6.2.2.2, I described that humans and animals tend to dynamically adjust the parameters regulating their behaviour based on long-term information about the current situation, which is beneficial to perform optimally in situations of fluctuating volatility. Patients suffering from anxiety and depression tend to have difficulties adjusting their behaviour to the long-term stability of the task [Browning et al., 2015]. Using a hierarchical approach, as described in chapter 6, would enable one to investigate and estimate more precisely the difference between controls and patients in that regard.

Moreover, SR-based computations are being increasingly investigated as experimental evidence suggests that they underlie more global cognitive capabilities, beyond spatial navigation [Behrens et al., 2018, Mark et al., 2020]. Depression and anxiety can greatly impair cognitive capabilities of patients [Castaneda et al., 2008]. For example, anxious patients tend to focus their thoughts on negative experiences, a phenomenon known as 'excessive worrying' [Lee et al., 2010]. A recent computational approach links the SR and value-based neuroeconomics models to explain the development of worry [Gagne and Dayan, 2020]. In general, the approaches presented in this thesis link neural computations to behaviours in the context of reward-based reinforcement and are very well suited to address questions related to cognition and motivation.

To conclude, the work presented in this thesis has identified key computational mechanisms underlying flexibility in spatial navigation, which all argue for efficient representations. It is my belief that the RL framework can be extended to embrace the complexity of a broad range of natural behaviours, including, and beyond, flexibility in spatial navigation.

Bibliography

Thomas Akam, Inês Rodrigues-Vaz, Ivo Marcelo, Xiangyu Zhang, Michael Pereira, Rodrigo Freire Oliveira, Peter Dayan, and Rui M Costa. Anterior cingulate cortex represents action-state predictions and causally mediates model-based reinforcement learning in a two-step decision task. *bioRxiv*, page 126292, 2020.

Mostafa Al-Emran. Hierarchical reinforcement learning: a survey. *International Journal of Computing and Digital Systems*, 4(02), 2015.

Majid Aljalal, Sutrisno Ibrahim, Ridha Djemal, and Wonsuk Ko. Comprehensive review on brain-controlled mobile robots and robotic arms based on electroencephalography signals. *Intelligent Service Robotics*, 13:539–563, 2020.

Alice Alvernhe, Etienne Save, and Bruno Poucet. Local remapping of place cell firing in the tolman detour task. *European Journal of Neuroscience*, 33(9):1696–1705, 2011.

R Ellen Ambrose, Brad E Pfeiffer, and David J Foster. Reverse replay of hippocampal place cells is uniquely modulated by changing reward. *Neuron*, 91(5):1124–1136, 2016.

Céline Amiez, Jean-Paul Joseph, and Emmanuel Procyk. Anterior cingulate error-related activity is modulated by predicted reward. *European Journal of Neuroscience*, 21(12): 3447–3452, 2005.

Dian Anggraini, Stefan Glasauer, and Klaus Wunderlich. Neural signatures of reinforcement learning correlate with strategy adoption during spatial navigation. *Scientific Reports*, 8(1):10110, 2018.

Lucy E. Annett, Anthony McGregor, and Trevor W. Robbins. The effects of ibotenic acid lesions of the nucleus accumbens on spatial learning and extinction in the rat. *Behavioural Brain Research*, 31(3):231–242, 1989.

Flavio Abreu Araujo, Mathieu Riou, Jacob Torrejon, Sumito Tsunegi, Damien Querlioz, Kay Yakushiji, Akio Fukushima, Hitoshi Kubota, Shinji Yuasa, Mark D Stiles, et al. Role of non-linear data processing on speech recognition task in the framework of reservoir computing. *Scientific reports*, 10(1):1–11, 2020.

Hisham E Atallah, Dan Lopez-Paniagua, Jerry W Rudy, and Randall C O'Reilly. Separate neural substrates for skill learning and performance in the ventral and dorsal striatum. *Nature Neuroscience*, 10(1):126–131, 2007.

Bernard W Balleine. The meaning of behavior: discriminating reflex and volition in the brain. *Neuron*, 104(1):47–62, 2019.

Andrea Banino, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, Martin J Chadwick, Thomas Degris, Joseph Modayil, et al. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705):429, 2018.

DM Bannerman, MA Good, SP Butcher, M Ramsay, and RGM Morris. Distinct components of spatial learning revealed by prior training and nmda receptor blockade. *Nature*, 378(6553):182–186, 1995.

João Carlos Alves Barata and Mahir Saleh Hussein. The moore–penrose pseudoinverse: A tutorial review of the theory. *Brazilian Journal of Physics*, 42(1-2):146–165, 2012.

André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 4055–4065, 2017.

Caswell Barry, Colin Lever, Robin Hayman, Tom Hartley, Stephen Burton, John O'Keefe, Kate Jeffery, and N Burgess. The boundary vector cell model of place cell firing and spatial memory. *Reviews in the Neurosciences*, 17(1-2):71–98, 2006.

Andrew G Barto and Sridhar Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 13(1):41–77, 2003.

Tobias Bast. Toward an integrative perspective on hippocampal function: from the rapid encoding of experience to adaptive behavior. *Reviews in the Neurosciences*, 18(3-4): 253–282, 2007.

Tobias Bast. The hippocampal learning-behavior translation and the functional significance of hippocampal dysfunction in schizophrenia. *Current Opinion in Neurobiology*, 21(3):492–501, 2011.

Tobias Bast, Bruno M da Silva, and Richard GM Morris. Distinct contributions of hippocampal NMDA and AMPA receptors to encoding and retrieval of one-trial place memory. *Journal of Neuroscience*, 25(25):5845–5856, 2005.

Tobias Bast, Iain A Wilson, Menno P Witter, and Richard GM Morris. From rapid place learning to behavioral performance: a key role for the intermediate hippocampus. *PLoS Biology*, 7(4):e1000089, 2009.

Markus Bauer, Matthew G Buckley, and Tobias Bast. Individual differences in theta-band oscillations in a spatial memory network revealed by eeg predict rapid place learning. *bioRxiv. Epub ahead of print 5 June.*, 2020.

John Robert Baxter and Jeffrey S Rosenthal. Rates of convergence for everywhere-positive markov chains. *Statistics & probability letters*, 22(4):333–338, 1995.

Timothy EJ Behrens, Mark W Woolrich, Mark E Walton, and Matthew FS Rushworth. Learning the value of information in an uncertain world. *Nature Neuroscience*, 10(9): 1214–1221, 2007.

Timothy EJ Behrens, Timothy H Muller, James CR Whittington, Shirley Mark, Alon B Baram, Kimberly L Stachenfeld, and Zeb Kurth-Nelson. What is a cognitive map? organizing knowledge for flexible behavior. *Neuron*, 100(2):490–509, 2018.

R Bellman. Dynamic programming princeton university press princeton. *New Jersey Google Scholar*, 1957.

José M Bernardo and Adrian FM Smith. *Bayesian theory*, volume 405. John Wiley & Sons, 2009.

William Bialek, Fred Rieke, RR De Ruyter Van Steveninck, and David Warland. Reading a neural code. *Science*, 252(5014):1854–1857, 1991.

Andrej Bicanski and Neil Burgess. Neuronal vector coding in spatial cognition. *Nature Reviews Neuroscience*, pages 1–18, 2020.

Calvin Bierley, Frances K McSweeney, and Renee Vannieuwkerk. Classical conditioning of preferences for stimuli. *Journal of Consumer Research*, 12(3):316–323, 1985.

Sonja Binder, Matthias Mölle, Michael Lippert, Ralf Bruder, Sonat Aksamaz, Frank Ohl, J Simon Wiegert, and Lisa Marshall. Monosynaptic hippocampal-prefrontal projections contribute to spatial memory consolidation in mice. *Journal of Neuroscience*, 39(35): 6978–6991, 2019.

Barak Blumenfeld, Son Preminger, Dov Sagi, and Misha Tsodyks. Dynamics of memory representations in networks with novelty-facilitated synaptic plasticity. *Neuron*, 52(2): 383–394, 2006.

Sander M Bohte. The evidence for neural information processing with precise spike-times: A survey. *Natural Computing*, 3(2):195–206, 2004.

Mathew Botvinick, Sam Ritter, Jane X Wang, Zeb Kurth-Nelson, Charles Blundell, and Demis Hassabis. Reinforcement learning, fast and slow. *Trends in Cognitive Sciences*, 2019.

Matthew M Botvinick, Yael Niv, and Andrew C Barto. Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. *Cognition*, 113(3): 262–280, 2009.

Flora Bouchacourt, Stefano Palminteri, Etienne Koechlin, and Srdjan Ostojic. Temporal chunking as a mechanism for unsupervised learning of task-sets. *bioRxiv*, page 713156, 2019.

Amanda A Braun, Devon L Graham, Tori L Schaefer, Charles V Vorhees, and Michael T Williams. Dorsal striatal dopamine depletion impairs both allocentric and egocentric navigation in rats. *Neurobiology of Learning and Memory*, 97(4):402–408, 2012.

Daniel A Braun, Carsten Mehring, and Daniel M Wolpert. Structure learning in action. *Behavioural Brain Research*, 206(2):157–165, 2010.

Romain Brette. Philosophy of the spike: rate-based vs. spike-based theories of the brain. *Frontiers in Systems Neuroscience*, 9:151, 2015.

Adam T Brockett, Stephen S Tennyson, Coreylyn A deBettencourt, Fatou Gaye, and Matthew R Roesch. Anterior cingulate cortex is necessary for adaptation of action plans. *Proceedings of the National Academy of Sciences*, 117(11):6196–6204, 2020.

Michael Browning, Timothy E Behrens, Gerhard Jocham, Jill X O'reilly, and Sonia J Bishop. Anxious individuals have difficulty learning the causal statistics of aversive environments. *Nature Neuroscience*, 18(4):590–596, 2015.

Matthew G Buckley and Tobias Bast. A new human delayed-matching-to-place test in a virtual environment reverse-translated from the rodent watermaze paradigm: Characterization of performance measures and sex differences. *Hippocampus*, 28(11):796–812, 2018.

György Buzsáki. Hippocampal sharp wave-ripple: A cognitive biomarker for episodic memory and planning. *Hippocampus*, 25(10):1073–1188, 2015.

Cameron M Bye and Robert J McDonald. A specific role of hippocampal nmda receptors and arc protein in rapid encoding of novel environmental representations and a more general long-term consolidation function. *Frontiers in Behavioral Neuroscience*, 13:8, 2019.

Francesca Cacucci, Colin Lever, Thomas J Wills, Neil Burgess, and John O'Keefe. Theta-modulated place-by-direction cells in the hippocampal formation in the rat. *Journal of Neuroscience*, 24(38):8265–8277, 2004.

Zografos Caramanos and Matthew L Shapiro. Spatial memory and n-methyl-d-aspartate receptor antagonists apv and mk-801: memory impairments depend on familiarity with the environment, drug dose, and training duration. *Behavioral Neuroscience*, 108(1): 30, 1994.

Margaret F Carr, Shantanu P Jadhav, and Loren M Frank. Hippocampal replay in the awake state: a potential substrate for memory consolidation and retrieval. *Nature Neuroscience*, 14(2):147, 2011.

Anu E Castaneda, Annamari Tuulio-Henriksson, Mauri Marttunen, Jaana Suvisaari, and Jouko Lönnqvist. A review on cognitive impairments in depressive and anxiety disorders with a focus on young adults. *Journal of Affective Disorders*, 106(1-2):1–27, 2008.

Nicolas Cazin, Martin Llofriu Alonso, Pablo Scleidorovich Chiodi, Tatiana Pelc, Bruce Harland, Alfredo Weitzenfeld, Jean-Marc Fellous, and Peter Ford Dominey. Reservoir computing model of prefrontal cortex creates novel combinations of previous navigation sequences from hippocampal place-cell replay with spatial reward propagation. *PLoS Computational Biology*, 15(7):e1006624, 2019.

Bruno Cessac, Hélène Paugam-Moisy, and Thierry Viéville. Overview of facts and issues about neural coding by spikes. *Journal of Physiology-Paris*, 104(1-2):5–18, 2010.

Janice Chen, Uri Hasson, and Christopher J Honey. Processing timescales as an organizing principle for primate cortex. *Neuron*, 88(2):244–246, 2015.

Fabian Chersi and Neil Burgess. The cognitive architecture of spatial navigation: hippocampal and striatal contributions. *Neuron*, 88(1):64–77, 2015.

Ami Citri and Robert C Malenka. Synaptic plasticity: multiple forms, functions, and mechanisms. *Neuropsychopharmacology*, 33(1):18–41, 2008.

Claudia Clopath, Lars Büsing, Eleni Vasilaki, and Wulfram Gerstner. Connectivity reflects coding: a model of voltage-based stdp with homeostasis. *Nature Neuroscience*, 13(3): 344, 2010.

Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

Christine M Constantinople, Alex T Piet, and Carlos D Brody. An analysis of decision under risk in rats. *Current Biology*, 29(12):2066–2074, 2019.

Laura H Corbit. Understanding the balance between goal-directed and habitual behavioral control. *Current Opinion in Behavioral Sciences*, 20:161–168, 2018.

Dane S Corneil and Wulfram Gerstner. Attractor network dynamics enable preplay and rapid path planning in maze–like environments. In *Advances in Neural Information Processing Systems*, pages 1684–1692, 2015.

Carolina Feher Da Silva and Todd A Hare. Humans are primarily model-based and not model-free learners in the two-stage task. *BioRxiv*, page 682922, 2019.

Teruko Danjo, Taro Toyoizumi, and Shigeyoshi Fujisawa. Spatial representations of self and other in the hippocampus. *Science*, 359(6372):213–218, 2018.

Nathaniel D Daw, Yael Niv, and Peter Dayan. Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nature Neuroscience*, 8(12):1704–1711, 2005.

Nathaniel D Daw, Samuel J Gershman, Ben Seymour, Peter Dayan, and Raymond J Dolan. Model-based influences on humans' choices and striatal prediction errors. *Neuron*, 69(6):1204–1215, 2011.

Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993.

Peter Dayan. Goal-directed control and its antipodes. *Neural Networks*, 22(3):213–219, 2009.

Peter Dayan and Geoffrey E Hinton. Feudal reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 271–278, 1993.

Livia de Hoz, Jane Knox, and Richard GM Morris. Longitudinal axis of the hippocampus: both septal and temporal poles of the hippocampus support water maze spatial learning depending on the training protocol. *Hippocampus*, 13(5):587–603, 2003.

Livia De Hoz, Edvard I Moser, and Richard GM Morris. Spatial learning with unilateral and bilateral hippocampal networks. *European Journal of Neuroscience*, 22(3):745–754, 2005.

Sophie Denève, Alireza Alemi, and Ralph Bourdoukan. The brain as an efficient and robust adaptive learner. *Neuron*, 94(5):969–977, 2017.

Bryan D Devan and Norman M White. Parallel information processing in the dorsal striatum: relation to hippocampal function. *Journal of Neuroscience*, 19(7):2789–2798, 1999.

Amir Dezfouli and Bernard W Balleine. Habits, action sequences and reinforcement learning. *European Journal of Neuroscience*, 35(7):1036–1051, 2012.

Amir Dezfouli and Bernard W Balleine. Learning the structure of the world: The adaptive nature of state-space and action representations in multi-stage decision-making. *PLoS Computational Biology*, 15(9):e1007334, 2019.

Ray J Dolan and Peter Dayan. Goals and habits in the brain. *Neuron*, 80(2):312–325, 2013.

Bradley B Doll, Katherine D Duncan, Dylan A Simon, Daphna Shohamy, and Nathaniel D Daw. Model-based choices involve prospective neural activity. *Nature Neuroscience*, 18 (5):767, 2015.

Laurent Dollé, Ricardo Chavarriaga, Agnès Guillot, and Mehdi Khamassi. Interactions of spatial strategies producing generalization gradient and blocking: A computational approach. *PLoS Computational Biology*, 14(4):e1006092, 2018.

Robert J Douglas. The hippocampus and behavior. *Psychological Bulletin*, 67(6):416, 1967.

Kenji Doya. Reinforcement learning in continuous time and space. *Neural Computation*, 12(1):219–245, 2000.

Kenji Doya. Metalearning and neuromodulation. *Neural Networks*, 15(4-6):495–506, 2002.

George Dragoi, Kenneth D Harris, and György Buzsáki. Place representation within hippocampal networks is modified by long-term potentiation. *Neuron*, 39(5):843–853, 2003.

David Dupret, Joseph O'neill, Barty Pleydell-Bouverie, and Jozsef Csicsvari. The reorganization and reactivation of hippocampal maps predict spatial memory performance. *Nature Neuroscience*, 13(8):995, 2010.

H Eichenbaum. Hippocampal representation in spatial learning. *Journal of Neuroscience*, 10:331–339, 1990.

Arne D Ekstrom, Michael J Kahana, Jeremy B Caplan, Tony A Fields, Eve A Isham, Ehren L Newman, and Itzhak Fried. Cellular networks underlying human spatial navigation. *Nature*, 425(6954):184–188, 2003.

Chris Eliasmith and Charles H Anderson. *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. MIT press, 2003.

Ian T Ellwood, Tosha Patel, Varun Wadia, Anthony T Lee, Alayna T Liptak, Kevin J Bender, and Vikaas S Sohal. Tonic or phasic stimulation of dopaminergic projections to prefrontal cortex causes mice to maintain or deviate from previously learned behavioral strategies. *Journal of Neuroscience*, 37(35):8315–8329, 2017.

Thomas W Elston, Eloise Croy, and David K Bilkey. Communication between the anterior cingulate cortex and ventral tegmental area during a cost-benefit reversal task. *Cell Reports*, 26(9):2353–2361, 2019.

Alexis Faure, Ulrike Haberland, Françoise Condé, and Nicole El Massioui. Lesion to the nigrostriatal dopamine system disrupts stimulus-response habit formation. *Journal of Neuroscience*, 25(11):2771–2780, 2005.

Michael Fauth, Florentin Wörgötter, and Christian Tetzlaff. The formation of multi-synaptic connections by the interaction of synaptic and structural plasticity and their functional consequences. *PLoS Computational Biology*, 11(1):e1004031, 2015.

Stan B Floresco, Jeremy K Seamans, and Anthony G Phillips. Selective roles for hippocampal, prefrontal cortical, and ventral striatal circuits in radial-arm maze tasks with or without a delay. *Journal of Neuroscience*, 17(5):1880–1890, 1997.

David J Foster. Replay comes of age. *Annual Review of Neuroscience*, 40:581–602, 2017.

David J Foster and Matthew A Wilson. Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature*, 440(7084):680–683, 2006.

DJ Foster, RGM Morris, and Peter Dayan. A model of hippocampally dependent navigation, using the temporal difference learning rule. *Hippocampus*, 10(1):1–16, 2000.

Mathias Franzius, Henning Sprekeler, and Laurenz Wiskott. Slowness and sparseness lead to place, head-direction, and spatial-view cells. *PLoS Computational Biology*, 3(8):e166, 2007.

Nicolas Frémaux, Henning Sprekeler, and Wulfram Gerstner. Reinforcement learning using a continuous time actor-critic framework with spiking neurons. *PLoS Computational Biology*, 9(4):e1003024, 2013.

Jonathan B Fritz, Stephen V David, Susanne Radtke-Schuller, Pingbo Yin, and Shihab A Shamma. Adaptive, behaviorally gated, persistent encoding of task-relevant auditory information in ferret frontal cortex. *Nature Neuroscience*, 13(8):1011, 2010.

Marianne Fyhn, Torkel Hafting, Alessandro Treves, May-Britt Moser, and Edvard I Moser. Hippocampal remapping and grid realignment in entorhinal cortex. *Nature*, 446(7132):190–194, 2007.

C Gagne and P Dayan. Avoidant behavior and pathological worry as optimal policies and optimal offline planning given individually exaggerated risk preferences. In *Bernstein Conference 2020*, 2020.

C Giovanni Galizia and Pierre-Marie Lledo. *Neurosciences-From Molecule to Behavior: a university textbook*. Springer, 2013.

Mona M Garvert, Raymond J Dolan, and Timothy EJ Behrens. A map of abstract relational knowledge in the human hippocampal–entorhinal cortex. *Elife*, 6:e17086, 2017.

Jeffrey L Gauthier and David W Tank. A dedicated population for reward coding in the hippocampus. *Neuron*, 99(1):179–193, 2018.

Tiago V Gehring, Gediminas Luksys, Carmen Sandi, and Eleni Vasilaki. Detailed classification of swimming paths in the Morris water maze: multiple strategies within one trial. *Scientific Reports*, 5:14562, 2015.

Thomas Gener, Lorena Perez-Mendez, and Maria V Sanchez-Vives. Tactile modulation of hippocampal place fields. *Hippocampus*, 23(12):1453–1462, 2013.

Charles R Gerfen and D James Surmeier. Modulation of striatal projection systems by dopamine. *Annual Review of Neuroscience*, 34:441–466, 2011.

Samuel J Gershman. Reinforcement learning and causal models. *The Oxford Handbook of Causal Reasoning*, page 295, 2017.

Samuel J Gershman. The successor representation: its computational logic and neural substrates. *Journal of Neuroscience*, 38(33):7193–7200, 2018.

Samuel J Gershman, Arthur B Markman, and A Ross Otto. Retrospective revaluation in sequential decision making: A tale of two systems. *Journal of Experimental Psychology: General*, 143(1):182, 2014a.

Samuel J Gershman, Angela Radulescu, Kenneth A Norman, and Yael Niv. Statistical computations underlying the dynamics of memory updating. *PLoS Computational Biology*, 10(11):e1003939, 2014b.

Wulfram Gerstner. Spiking neurons. Technical report, MIT-press, 1998.

Wulfram Gerstner and LF Abbott. Learning navigational maps through potentiation and modulation of hippocampal place cells. *Journal of Computational Neuroscience*, 4(1): 79–94, 1997.

Wulfram Gerstner, JL van Hemmen, and TJ Sejnowski. How can the brain be so fast. *JL van Hemmen & TJ Sejnowski (Eds.)*, 23:135–142, 2005.

Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

Narcis Ghisovan, Abdellatif Nemri, Svetlana Shumikhina, and Stephane Molotchnikoff. Synchrony between orientation-selective neurons is modulated during adaptation-induced plasticity in cat visual cortex. *BMC Neuroscience*, 9(1):1–17, 2008.

Paul W Glimcher. Understanding dopamine and reinforcement learning: the dopamine reward prediction error hypothesis. *Proceedings of the National Academy of Sciences*, 108(Supplement 3):15647–15654, 2011.

Joshua I Gold and Long Ding. How mechanisms of perceptual decision-making affect the psychometric function. *Progress in Neurobiology*, 103:98–114, 2013.

Yukiori Goto and Anthony A Grace. Dopamine modulation of hippocampal–prefrontal cortical interaction drives memory-guided behavior. *Cerebral Cortex*, 18(6):1407–1414, 2008.

HJ Groenewegen, E te Vermeulen-Van der Zee, A Te Kortschot, and MP Witter. Organization of the projections from the subiculum to the ventral striatum in the rat. A study using anterograde transport of Phaseolus vulgaris leucoagglutinin. *Neuroscience*, 23(1):103–120, 1987.

Anoopum S Gupta, Matthijs AA van der Meer, David S Touretzky, and A David Redish. Hippocampal replay is not a simple function of experience. *Neuron*, 65(5):695–705, 2010.

Nicholas J Gustafson and Nathaniel D Daw. Grid cells, place cells, and geodesic generalization for spatial reinforcement learning. *PLoS Computational Biology*, 7(10):e1002235, 2011.

Juergen Haag and Alexander Borst. Encoding of visual motion information and reliability in spiking and graded potential neurons. *Journal of Neuroscience*, 17(12):4809–4819, 1997.

Thomas Haferlach, Jan Wessnitzer, Michael Mangan, and Barbara Webb. Evolving a neural model of insect path integration. *Adaptive Behavior*, 15(3):273–287, 2007.

Torkel Hafting, Marianne Fyhn, Sturla Molden, May-Britt Moser, and Edvard I Moser. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–806, 2005.

Jihun Ham, Daniel D Lee, Sebastian Mika, and Bernhard Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the twenty-first international conference on Machine learning*, page 47, 2004.

Constance Hammond. *Cellular and molecular neurophysiology.* Academic Press, 2014.

Michael E Hasselmo. *How we remember: brain mechanisms of episodic memory.* MIT Press, 2011.

Linnea E Herzog, Leila May Pascual, Seneca J Scott, Elon R Mathieson, Donald B Katz, and Shantanu P Jadhav. Interaction of taste and place coding in the hippocampus. *Journal of Neuroscience*, 39(16):3057–3069, 2019.

Desmond J Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Review*, 43(3):525–546, 2001.

James R Hinman, G William Chapman, and Michael E Hasselmo. Neuronal representation of environmental boundaries in egocentric coordinates. *Nature communications*, 10(1): 1–8, 2019.

V Hok, E Save, PP Lenck-Santini, and B Poucet. Coding for spatial goals in the prelimbic/infralimbic area of the rat frontal cortex. *Proceedings of the National Academy of Sciences*, 102(12):4602–4607, 2005.

Vincent Hok, Pierre-Pascal Lenck-Santini, Sébastien Roux, Etienne Save, Robert U Muller, and Bruno Poucet. Goal-related activity in hippocampal place cells. *Journal of Neuroscience*, 27(3):472–482, 2007.

Vincent Hok, Ehsan Chah, Etienne Save, and Bruno Poucet. Prefrontal cortex focally modulates hippocampal place cell firing patterns. *Journal of Neuroscience*, 33(8):3443–3451, 2013.

Marc W Howard and Michael E Hasselmo. Cognitive computation using neural representations of time and space in the laplace domain. *arXiv preprint arXiv:2003.11668*, 2020.

Mark D Humphries and Tony J Prescott. The ventral basal ganglia, a selection mechanism at the crossroads of space, strategy, and reward. *Progress in Neurobiology*, 90(4):385–417, 2010.

Mark D Humphries, Mehdi Khamassi, and Kevin Gurney. Dopaminergic control of the exploration-exploitation trade-off via the basal ganglia. *Frontiers in Neuroscience*, 6:9, 2012.

Quentin J. M. Huys, Anthony Cruickshank, and Peggy Seriès. *Reward-Based Learning, Model-Based and Model-Free*, pages 1–10. Springer New York, New York, NY, 2013. ISBN 978-1-4614-7320-6.

Quentin JM Huys, Neir Eshel, Elizabeth O'Nions, Luke Sheridan, Peter Dayan, and Jonathan P Roiser. Bonsai trees in your head: how the pavlovian system sculpts goal-directed choices by pruning decision trees. *PLoS Computational Biology*, 8(3):e1002410, 2012.

Hideyoshi Igata, Yuji Ikegaya, and Takuya Sasaki. Prioritized experience replays on a hippocampal predictive map for learning. *bioRxiv*, 2020.

Kiyohito Iigaya, Yashar Ahmadian, Leo P Sugrue, Greg S Corrado, Yonatan Loewenstein, William T Newsome, and Stefano Fusi. Deviation from the matching law reflects an optimal strategy involving learning over multiple timescales. *Nature Communications*, 10(1):1–14, 2019.

Jennifer Inglis, Stephen J Martin, and Richard GM Morris. Upstairs/downstairs revisited: spatial pretraining-induced rescue of normal spatial learning during selective blockade of hippocampal n-methyl-d-aspartate receptors. *European Journal of Neuroscience*, 37 (5):718–727, 2013.

John TR Isaac, Katherine A Buchanan, Robert U Muller, and Jack R Mellor. Hippocampal place cell firing patterns can induce long-term synaptic plasticity in vitro. *Journal of Neuroscience*, 29(21):6840–6850, 2009.

Shin Ishii, Wako Yoshida, and Junichiro Yoshimoto. Control of exploitation–exploration meta-parameter in reinforcement learning. *Neural Networks*, 15(4-6):665–687, 2002.

Hiroshi T Ito, Sheng-Jia Zhang, Menno P Witter, Edvard I Moser, and May-Britt Moser. A prefrontal–thalamo–hippocampal circuit for goal-directed spatial navigation. *Nature*, 522(7554):50–55, 2015.

Eugene M Izhikevich. Solving the distal reward problem through linkage of stdp and dopamine signaling. *Cerebral Cortex*, 17(10):2443–2452, 2007.

Shantanu P Jadhav, Caleb Kemere, P Walter German, and Loren M Frank. Awake hippocampal sharp-wave ripples support spatial memory. *Science*, 336(6087):1454–1458, 2012.

Kate J Jeffery. Cognitive representations of spatial location. *Brain and Neuroscience Advances*, 2:2398212818810686, 2018.

Marieke Jepma, Erik T Te Beek, Eric-Jan Wagenmakers, Joop Van Gerven, and Sander Nieuwenhuis. The role of the noradrenergic system in the exploration-exploitation trade-off: a pharmacological study. *Frontiers in Human Neuroscience*, 4:170, 2010.

Yong Sang Jo, Eun Hye Park, Il Hwan Kim, Soon Kwon Park, Hyun Kim, Hyun Taek Kim, and June-Seek Choi. The medial prefrontal cortex is involved in spatial memory retrieval under partial-cue conditions. *Journal of Neuroscience*, 27(49):13567–13578, 2007.

Daphna Joel, Yael Niv, and Eytan Ruppin. Actor–critic models of the basal ganglia: New anatomical and computational perspectives. *Neural Networks*, 15(4-6):535–547, 2002.

Roland S Johansson and Ingvars Birznieks. First spikes in ensembles of human tactile afferents code complex spatial fingertip events. *Nature Neuroscience*, 7(2):170–177, 2004.

Adam Johnson and A David Redish. Neural ensembles in ca3 transiently encode paths forward of the animal at a decision point. *Journal of Neuroscience*, 27(45):12176–12189, 2007.

Szabolcs Káli and Peter Dayan. The involvement of recurrent connections in area ca3 in establishing the properties of place fields: a model. *Journal of Neuroscience*, 20(19): 7463–7477, 2000.

Richard Kempter, Wulfram Gerstner, and J Leo Van Hemmen. Spike-based compared to rate-based hebbian learning. *Advances in Neural Information Processing Systems*, 11: 125–131, 1999.

Mehdi Keramati, Amir Dezfouli, and Payam Piray. Speed/accuracy trade-off between the habitual and the goal-directed processes. *PLoS Computational Biology*, 7(5), 2011.

Mehdi Khamassi and Mark D Humphries. Integrating cortico-limbic-basal ganglia architectures for learning model-based and model-free navigation strategies. *Frontiers in Behavioral Neuroscience*, 6:79, 2012.

Mehdi Khamassi, Pierre Enel, Peter Ford Dominey, and Emmanuel Procyk. Medial prefrontal cortex and the adaptive regulation of reinforcement learning parameters. *Progress in Brain Research*, 202:441–464, 2013.

Eyal Yaacov Kimchi and Mark Laubach. Dynamic encoding of action selection by the medial striatum. *Journal of Neuroscience*, 29(10):3148–3159, 2009.

Kirsten Brun Kjelstrup, Trygve Solstad, Vegard Heimly Brun, Torkel Hafting, Stefan Leutgeb, Menno P Witter, Edvard I Moser, and May-Britt Moser. Finite scale of spatial representation in the hippocampus. *Science*, 321(5885):140–143, 2008.

Markus Knaden and Rüdiger Wehner. Ant navigation: resetting the path integrator. *Journal of Experimental Biology*, 209(1):26–31, 2006.

Nils Kolling, Marco K Wittmann, Tim EJ Behrens, Erie D Boorman, Rogier B Mars, and Matthew FS Rushworth. Value, search, persistence and model updating in anterior cingulate cortex. *Nature Neuroscience*, 19(10):1280–1285, 2016.

Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

George F Koob and Nora D Volkow. Neurocircuitry of addiction. *Neuropsychopharmacology*, 35(1):217–238, 2010.

Carien S Lansink, Pieter M Goltstein, Jan V Lankelma, Bruce L McNaughton, and Cyriel MA Pennartz. Hippocampus leads ventral striatum in replay of place-reward information. *PLoS Biology*, 7(8), 2009.

Brian Lau, Tiago Monteiro, and Joseph J Paton. The many worlds hypothesis of dopamine prediction error: implications of a parallel circuit architecture in the basal ganglia. *Current Opinion in Neurobiology*, 46:241–247, 2017.

Inah Lee and Raymond P Kesner. Differential contribution of nmda receptors in hippocampal subregions to spatial working memory. *Nature Neuroscience*, 5(2):162–168, 2002.

Jonathan K Lee, Susan M Orsillo, Lizabeth Roemer, and Laura B Allen. Distress and avoidance in generalized anxiety disorder: Exploring the relationships with intolerance of uncertainty and worry. *Cognitive Behaviour Therapy*, 39(2):126–136, 2010.

Tara A LeGates, Mark D Kvarta, Jessica R Tooley, T Chase Francis, Mary Kay Lobo, Meaghan C Creed, and Scott M Thompson. Reward behaviour is regulated by the strength of hippocampus–nucleus accumbens synapses. *Nature*, 564(7735):258–262, 2018.

Mimi Liljeholm, Simon Dunne, and John P O'Doherty. Differentiating neural systems mediating the acquisition vs. expression of goal-directed and habitual behavioral control. *European Journal of Neuroscience*, 41(10):1358–1371, 2015.

Michael Losh and Daniel Llamocca. A low-power spike-like neural network design. *Electronics*, 8(12):1479, 2019.

Marlos C Machado, Marc G Bellemare, and Michael Bowling. Count-based exploration with the successor representation. *arXiv preprint arXiv:1807.11622*, 2018.

Silvia Maggi, Adrien Peyrache, and Mark D Humphries. An ensemble code in medial prefrontal cortex links prior events to outcomes during learning. *Nature Communications*, 9(1):1–12, 2018.

Sridhar Mahadevan. *Learning representation and control in markov decision processes.* Now Publishers Inc, 2009.

Pedro Maldonado, Cecilia Babul, Wolf Singer, Eugenio Rodriguez, Denise Berger, and Sonja Grun. Synchronization of neuronal responses in primary visual cortex of monkeys viewing natural images. *Journal of Neurophysiology*, 100(3):1523–1532, 2008.

Shirley Mark, Rani Moran, Thomas Parr, Steve W Kennerley, and Timothy EJ Behrens. Transferring structural knowledge across cognitive maps in humans and models. *Nature Communications*, 11(1):1–12, 2020.

Elizabeth Marozzi and Kathryn J Jeffery. Place, space and memory cells. *Current Biology*, 22(22):R939–R942, 2012.

Madoka Matsumoto, Kenji Matsumoto, Hiroshi Abe, and Keiji Tanaka. Medial prefrontal cell activity signaling prediction errors of action values. *Nature Neuroscience*, 10(5): 647–656, 2007.

Marcelo G Mattar and Nathaniel D Daw. Prioritized memory access explains planning and hippocampal replay. *Nature Neuroscience*, 21(11):1609–1617, 2018.

Stephanie McGarrity, Sorley Somerled, Curtis Eaton, Rob Mason, Marie Pezze, and Tobias Bast. Medial prefrontal cortex is not required for, but can modulate, hippocampus-dependent behaviour based on rapid learning of changing goal locations on the watermaze delayed-matching-to-place test. In *British Neuroscience Association Abstract*, volume 23(P1-D-013), 2015.

Stephanie McGarrity, Rob Mason, Kevin C Fone, Marie Pezze, and Tobias Bast. Hippocampal neural disinhibition causes attentional and memory deficits. *Cerebral Cortex*, 27(9):4447–4462, 2017.

Sam McKenzie, Nick TM Robinson, Lauren Herrera, Jordana C Churchill, and Howard Eichenbaum. Learning causes reorganization of neuronal firing patterns to represent related experiences within a hippocampal schema. *Journal of Neuroscience*, 33(25): 10243–10256, 2013.

Sam McKenzie, Andrea J Frank, Nathaniel R Kinsky, Blake Porter, Pamela D Rivière, and Howard Eichenbaum. Hippocampal representation of related and opposing memories develop within distinct, hierarchically organized neural schemas. *Neuron*, 83(1):202–215, 2014.

Colin G McNamara, Álvaro Tejero-Cantero, Stéphanie Trouche, Natalia Campo-Urriza, and David Dupret. Dopaminergic neurons promote hippocampal reactivation and spatial memory persistence. *Nature Neuroscience*, 17(12):1658–1660, 2014.

Earl K Miller and Jonathan D Cohen. An integrative theory of prefrontal cortex function. *Annual Review of Neuroscience*, 24(1):167–202, 2001.

Kevin J Miller and Sarah Jo C Venditto. Multi-step planning in the brain. *Current Opinion in Behavioral Sciences*, 38:29–39, 2021.

Kevin J Miller, Matthew M Botvinick, and Carlos D Brody. Dorsal hippocampus contributes to model-based planning. *Nature Neuroscience*, 20(9):1269, 2017.

Ida Momennejad, Evan M Russek, Jin H Cheong, Matthew M Botvinick, Nathaniel Douglass Daw, and Samuel J Gershman. The successor representation in human reinforcement learning. *Nature Human Behaviour*, 1(9):680–692, 2017.

Anthony M Monacelli, Laura A Cushman, Voyko Kavcic, and Charles J Duffy. Spatial disorientation in alzheimer's disease: the remembrance of things passed. *Neurology*, 61 (11):1491–1497, 2003.

P Read Montague, Raymond J Dolan, Karl J Friston, and Peter Dayan. Computational psychiatry. *Trends in Cognitive Sciences*, 16(1):72–80, 2012.

Genela Morris, Robert Schmidt, and Hagai Bergman. Striatal action-learning based on dopamine concentration. *Experimental Brain Research*, 200(3-4):307–317, 2010.

RGM Morris, Elizabeth Anderson, GS a Lynch, and Michel Baudry. Selective impairment of learning and blockade of long-term potentiation by an N-methyl-D-aspartate receptor antagonist, AP5. *Nature*, 319(6056):774–776, 1986.

RGM Morris, RF Halliwell, and N Bowery. Synaptic plasticity and learning ii: do different kinds of plasticity underlie different kinds of learning? *Neuropsychologia*, 27(1):41–59, 1989.

RGM Morris, F Schenk, F Tweedie, and LE Jarrard. Ibotenate lesions of hippocampus and/or subiculum: dissociating components of allocentric spatial learning. *European Journal of Neuroscience*, 2(12):1016–1028, 1990.

Richard GM Morris. Spatial localization does not require the presence of local cues. *Learning and Motivation*, 12(2):239–260, 1981.

Richard GM Morris, Paul Garrud, JNP al Rawlins, and John O'Keefe. Place navigation impaired in rats with hippocampal lesions. *Nature*, 297(5868):681–683, 1982.

Edvard I Moser, Emilio Kropff, and May-Britt Moser. Place cells, grid cells, and the brain's spatial representation system. *Annual Review of Neuroscience*, 31:69–89, 2008.

Edvard I Moser, May-Britt Moser, and Bruce L McNaughton. Spatial representation in the hippocampal formation: a history. *Nature Neuroscience*, 20(11):1448, 2017.

May-Britt Moser, Edvard I Moser, Elma Forrest, Per Andersen, and RG Morris. Spatial learning with a minislab in the dorsal hippocampus. *Proceedings of the National Academy of Sciences*, 92(21):9697–9701, 1995.

Martin Müller and Rüdiger Wehner. Path integration in desert ants, cataglyphis fortis. *Proceedings of the National Academy of Sciences*, 85(14):5287–5290, 1988.

Robert U Muller, James B Ranck Jr, and Jeffrey S Taube. Head direction cells: properties and functional significance. *Current Opinion in Neurobiology*, 6(2):196–206, 1996.

Kazu Nakazawa, Linus D Sun, Michael C Quirk, Laure Rondi-Reig, Matthew A Wilson, and Susumu Tonegawa. Hippocampal ca3 nmda receptors are crucial for memory acquisition of one-time experience. *Neuron*, 38(2):305–315, 2003.

Kazu Nakazawa, Thomas J McHugh, Matthew A Wilson, and Susumu Tonegawa. Nmda receptors, place cells and hippocampal spatial memory. *Nature Reviews Neuroscience*, 5(5):361–372, 2004.

Rapeechai Navawongse and Howard Eichenbaum. Distinct pathways for rule-based retrieval and spatial mapping of memory representations in hippocampal neurons. *Journal of Neuroscience*, 33(3):1002–1013, 2013.

Wilten Nicola and Claudia Clopath. Supervised learning in spiking neural networks with force training. *Nature Communications*, 8(1):1–15, 2017.

John O'Doherty, Peter Dayan, Johannes Schultz, Ralf Deichmann, Karl Friston, and Raymond J Dolan. Dissociable roles of ventral and dorsal striatum in instrumental conditioning. *Science*, 304(5669):452–454, 2004.

John O'Keefe and Jonathan Dostrovsky. The hippocampus as a spatial map: preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, 1971.

John O'keefe and Lynn Nadel. *The hippocampus as a cognitive map*. Oxford: Clarendon Press, 1978.

John O'Keefe and Michael L Recce. Phase relationship between hippocampal place units and the eeg theta rhythm. *Hippocampus*, 3(3):317–330, 1993.

Flavio TP Oliveira, John J McDonald, and David Goodman. Performance monitoring in the anterior cingulate is not all error related: expectancy deviation and the representation of action-outcome associations. *Journal of Cognitive Neuroscience*, 19(12): 1994–2004, 2007.

Bruno A Olshausen. Linear Hebbian Learning and PCA. 1998.

CJP Oswald and Mark Good. The effects of combined lesions of the subicular complex and the entorhinal cortex on two forms of spatial navigation in the water maze. *Behavioral Neuroscience*, 114(1):211, 2000.

Colin M O'Carroll, Stephen J Martin, Johan Sandin, Bruno Frenguelli, and Richard GM Morris. Dopaminergic modulation of the persistence of one-trial hippocampus-dependent memory. *Learning & Memory*, 13(6):760–769, 2006.

Mark G Packard and James L McGaugh. Inactivation of hippocampus or caudate nucleus with lidocaine differentially affects expression of place and response learning. *Neurobiology of Learning and Memory*, 65(1):65–72, 1996.

Fabio Pardo, Vitaly Levdik, and Petar Kormushev. Goal-oriented trajectories for efficient exploration. *arXiv preprint arXiv:1807.02078*, 2018.

Ivan Petrovich Pavlov. *The work of the digestive glands*. Charles Griffin, 1902.

John M Pearce, Amanda DL Roberts, and Mark Good. Hippocampal lesions disrupt navigation based on cognitive maps but not heading vectors. *Nature*, 396(6706):75–77, 1998.

CMA Pennartz, R Ito, PFMJ Verschure, FP Battaglia, and TW Robbins. The hippocampal–striatal axis in learning, prediction and goal-directed behavior. *Trends in Neurosciences*, 34(10):548–559, 2011.

Will D Penny, Peter Zeidman, and Neil Burgess. Forward and backward inference in spatial cognition. *PLoS Computational Biology*, 9(12):e1003383, 2013.

Roger Penrose. On best approximate solutions of linear matrix equations. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 52, pages 17–19. Cambridge University Press, 1956.

Marie Pezze and Tobias Bast. Dopaminergic modulation of hippocampus-dependent learning: blockade of hippocampal d1-class receptors during learning impairs 1-trial place memory at a 30-min retention delay. *Neuropharmacology*, 63(4):710–718, 2012.

Giovanni Pezzulo, Francesco Rigoli, and Fabian Chersi. The mixed instrumental controller: using value of information to combine habitual choice and mental simulation. *Frontiers in Psychology*, 4:92, 2013.

Giovanni Pezzulo, Caleb Kemere, and Matthijs AA Van Der Meer. Internally generated hippocampal sequences as a vantage point to probe future-oriented cognition. *Annals of the New York Academy of Sciences*, 1396(1):144–165, 2017.

Brad E Pfeiffer and David J Foster. Hippocampal place-cell sequences depict future paths to remembered goals. *Nature*, 497(7447):74–79, 2013.

S Unnikrishna Pillai, Torsten Suel, and Seunghun Cha. The perron-frobenius theorem: some of its applications. *IEEE Signal Processing Magazine*, 22(2):62–75, 2005.

Bruno Poucet and Vincent Hok. Remembering goal locations. *Current Opinion in Behavioral Sciences*, 17:51–56, 2017.

Steven Poulter, Tom Hartley, and Colin Lever. The neurobiology of mammalian navigation. *Current Biology*, 28(17):R1023–R1042, 2018.

Rodrigo Quian Quiroga and Gabriel Kreiman. Postscript: About grandmother cells and jennifer aniston neurons. 2010.

Rahul Ramesh, Manan Tomar, and Balaraman Ravindran. Successor options: An option discovery framework for reinforcement learning. *arXiv preprint arXiv:1905.05731*, 2019.

A David Redish. Vicarious trial and error. *Nature Reviews Neuroscience*, 17(3):147, 2016.

A David Redish and David S Touretzky. The role of the hippocampus in solving the morris water maze. *Neural Computation*, 10(1):73–111, 1998.

Alexa Riehle, Sonja Grün, Markus Diesmann, and Ad Aertsen. Spike synchronization and rate modulation differentially involved in motor cortical function. *Science*, 278(5345): 1950–1953, 1997.

Bruno Rivard, Yu Li, Pierre-Pascal Lenck-Santini, Bruno Poucet, and Robert U Muller. Representation of objects in space by two classes of hippocampal pyramidal cells. *The Journal of General Physiology*, 124(1):9–25, 2004.

Trevor W Robbins and Barry J Everitt. Functions of dopamine in the dorsal and ventral striatum. In *Seminars in Neuroscience*, volume 4, pages 119–127. Elsevier, 1992.

Matthew R Roesch, Teghpal Singh, P Leon Brown, Sylvina E Mullins, and Geoffrey Schoenbaum. Ventral striatal neurons encode the value of the chosen action in rats deciding between differently delayed or sized rewards. *Journal of Neuroscience*, 29(42): 13365–13376, 2009.

Sebastian-Daniel Rosca and Monica Leba. Using brain-computer-interface for robot arm control. In *MATEC Web of Conferences*, volume 121, page 08006. EDP Sciences, 2017.

Román Rossi-Pool, Antonio Zainos, Manuel Alvarez, Sergio Parra, Jerónimo Zizumbo, and Ranulfo Romo. Invariant timescale hierarchy across the cortical somatosensory network. *Proceedings of the National Academy of Sciences*, 118(3), 2021.

Sarah Ruediger, Dominique Spirig, Flavio Donato, and Pico Caroni. Goal-oriented searching mediated by ventral hippocampus early in trial-and-error learning. *Nature Neuroscience*, 15(11):1563, 2012.

Matthew FS Rushworth and Timothy EJ Behrens. Choice, uncertainty and value in prefrontal and cingulate cortex. *Nature Neuroscience*, 11(4):389–397, 2008.

Evan M Russek, Ida Momennejad, Matthew M Botvinick, Samuel J Gershman, and Nathaniel D Daw. Predictive representations can link model-based reinforcement learning to model-free mechanisms. *PLoS Computational Biology*, 13(9):e1005768, 2017.

Silviu I Rusu and Cyriel MA Pennartz. Learning, memory and consolidation mechanisms for behavioral control in hierarchically organized cortico-basal ganglia systems. *Hippocampus*, 30(1):73–98, 2020.

Kazuyuki Samejima, Yasumasa Ueda, Kenji Doya, and Minoru Kimura. Representation of action-specific reward values in the striatum. *Science*, 310(5752):1337–1340, 2005.

Ayelet Sarel, Arseny Finkelstein, Liora Las, and Nachum Ulanovsky. Vectorial representation of spatial goals in the hippocampus of bats. *Science*, 355(6321):176–180, 2017.

Francesca Sargolini, Marianne Fyhn, Torkel Hafting, Bruce L McNaughton, Menno P Witter, May-Britt Moser, and Edvard I Moser. Conjunctive representation of position, direction, and velocity in entorhinal cortex. *Science*, 312(5774):758–762, 2006.

RJ Sayer, MJ Friedlander, and SJ Redman. The time course and amplitude of epsps evoked at synapses between pairs of CA3/CA1 neurons in the hippocampal slice. *Journal of Neuroscience*, 10(3):826–836, 1990.

Fabian Schönfeld and Laurenz Wiskott. Modeling place field activity with hierarchical slow feature analysis. *Frontiers in Computational Neuroscience*, 9:51, 2015.

Drew C Schreiner, Rafael Renteria, and Christina M Gremel. Fractionating the all-or-nothing definition of goal-directed and habitual decision-making. *Journal of Neuroscience Research*, 98(6):998–1006, 2020.

Wolfram Schultz. Dopamine reward prediction-error signalling: a two-component response. *Nature Reviews Neuroscience*, 17(3):183, 2016.

Wolfram Schultz, Peter Dayan, and P Read Montague. A neural substrate of prediction and reward. *Science*, 275(5306):1593–1599, 1997.

Nicolas Schweighofer and Kenji Doya. Meta-learning in reinforcement learning. *Neural Networks*, 16(1):5–9, 2003.

Pablo Scleidorovich, Martin Llofriu, Jean-Marc Fellous, and Alfredo Weitzenfeld. A computational model for spatial cognition combining dorsal and ventral hippocampal place field maps: multiscale navigation. *Biological Cybernetics*, pages 1–21, 2020.

William Beecher Scoville and Brenda Milner. Loss of recent memory after bilateral hippocampal lesions. *Journal of Neurology, Neurosurgery, and Psychiatry*, 20(1):11, 1957.

Jeremy K Seamans, Stanley B Floresco, and Anthony G Phillips. Functional differences between the prelimbic and anterior cingulate regions of the rat prefrontal cortex. *Behavioral Neuroscience*, 109(6):1063, 1995.

Adam Seaton. *An investigation of the role of the nucleus accumbens in the hippocampal learning-behaviour translation.* PhD thesis, University of Nottingham, 2019.

Matthew L Shapiro and Christine O'Connor. N-methyl-d-aspartate receptor antagonist mk-801 and spatial memory representation: Working memory is impaired in an unfamiliar but not in a familiar environment. *Behavioral Neuroscience*, 106(4):604, 1992.

Justin D Shin, Wenbo Tang, and Shantanu P Jadhav. Dynamics of awake hippocampal-prefrontal replay for spatial learning and memory-guided decision making. *Neuron*, 104 (6):1110–1125, 2019.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Jesper Sjöström and Wulfram Gerstner. Spike-timing dependent plasticity. *Spike-timing Dependent Plasticity*, 35(0):0–0, 2010.

William E Skaggs, Bruce L McNaughton, Matthew A Wilson, and Carol A Barnes. Theta phase precession in hippocampal neuronal populations and the compression of temporal sequences. *Hippocampus*, 6(2):149–172, 1996.

Burrhus F Skinner. Operant conditioning. *The Encyclopedia of Education*, 7:29–33, 1971.

Kyle S Smith and Ann M Graybiel. A dual operator view of habitual behavior reflecting cortical and striatal dynamics. *Neuron*, 79(2):361–374, 2013.

Spencer L Smith, Ikuko T Smith, Tiago Branco, and Michael Häusser. Dendritic spikes enhance stimulus selectivity in cortical neurons in vivo. *Nature*, 503(7474):115–120, 2013.

Timothy Spellman, Mattia Rigotti, Susanne E Ahmari, Stefano Fusi, Joseph A Gogos, and Joshua A Gordon. Hippocampal–prefrontal input supports spatial encoding in working memory. *Nature*, 522(7556):309–314, 2015.

Hugo J Spiers. Keeping the goal in mind: Prefrontal contributions to spatial navigation. *Neuropsychologia*, 46(7):2106, 2008.

Kimberly L Stachenfeld, Matthew Botvinick, and Samuel J Gershman. Design principles of the hippocampal cognitive map. *Advances in Neural Information Processing Systems*, 27:2528–2536, 2014.

Kimberly L Stachenfeld, Matthew M Botvinick, and Samuel J Gershman. The hippocampus as a predictive map. *Nature Neuroscience*, 20(11):1643, 2017.

RJ Steele and RGM Morris. Delay-dependent impairment of a matching-to-place task with chronic and intrahippocampal infusion of the nmda-antagonist d-ap5. *Hippocampus*, 9 (2):118–136, 1999.

Ivilin Peev Stoianov, Cyriel MA Pennartz, Carien S Lansink, and Giovani Pezzulo. Model-based spatial navigation in the hippocampus-ventral striatum circuit: A computational analysis. *PLoS Computational Biology*, 14(9):e1006316, 2018.

Bryan A Strange, Menno P Witter, Ed S Lein, and Edvard I Moser. Functional organization of the hippocampal longitudinal axis. *Nature Reviews Neuroscience*, 15(10): 655–669, 2014.

Steven P Strong, Roland Koberle, Rob R De Ruyter Van Steveninck, and William Bialek. Entropy and information in neural spike trains. *Physical Review Letters*, 80(1):197, 1998.

Chen Sun, Wannan Yang, Jared Martin, and Susumu Tonegawa. Hippocampal neurons represent events as transferable units of experience. *Nature Neuroscience*, pages 1–13, 2020.

David Sussillo and Larry F Abbott. Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557, 2009.

Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

Richard S Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems*, pages 1038–1044, 1996.

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

Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211, 1999.

Kyra Swanson, Bruno B Averbeck, and Mark Laubach. Noradrenergic regulation of win-stay/lose-shift policy and choice determinism in a two-armed bandit task. *bioRxiv*, 2020.

Lung-Hao Tai, A Moses Lee, Nora Benavidez, Antonello Bonci, and Linda Wilbrecht. Transient stimulation of distinct subpopulations of striatal neurons mimics changes in action value. *Nature Neuroscience*, 15(9):1281, 2012.

Frederick Winslow Taylor. *The principles of scientific management.* Harper & brothers, 1919.

Brad Theilman, Krista Perks, and Timothy Q Gentner. Spike train coactivity encodes learned natural stimulus invariances in songbird auditory cortex. *Journal of Neuroscience*, 41(1):73–88, 2021.

Edward L Thorndike. The law of effect. *The American Journal of Psychology*, 39(1/4): 212–222, 1927.

Simon Thorpe, Arnaud Delorme, and Rufin Van Rullen. Spike-based strategies for rapid processing. *Neural Networks*, 14(6-7):715–725, 2001.

Sebastian B Thrun. Efficient exploration in reinforcement learning. 1992.

Edward C Tolman. Cognitive maps in rats and men. *Psychological Review*, 55(4):189, 1948.

Momchil S Tomov, Van Q Truong, Rohan A Hundia, and Samuel J Gershman. Dissociable neural correlates of uncertainty underlie different exploration strategies. *Nature Communications*, 11(1):1–12, 2020.

Bryan P Tripp and Chris Eliasmith. Population models of temporal differentiation. *Neural Computation*, 22(3):621–659, 2010.

Endel Tulving and Hans J Markowitsch. Episodic and declarative memory: role of the hippocampus. *Hippocampus*, 8(3):198–204, 1998.

Matthijs AA Van Der Meer and A David Redish. Theta phase precession in rat ventral striatum links place and reward information. *Journal of Neuroscience*, 31(8):2843–2854, 2011.

Matthijs AA van Der Meer and A David Redish. Ventral striatum: a critical look at models of learning and evaluation. *Current Opinion in Neurobiology*, 21(3):387–392, 2011.

Jamilja AJ van der Meulen, Ruud NJMA Joosten, Jan PC de Bruin, and Matthijs GP Feenstra. Dopamine and noradrenaline efflux in the medial prefrontal cortex during serial reversals and extinction of instrumental goal-directed behavior. *Cerebral Cortex*, 17(6):1444–1453, 2007.

Wim van Drongelen. 14 - spike train analysis. In Wim van Drongelen, editor, *Signal Processing for Neuroscientists*, pages 219–243. Academic Press, Burlington, 2007. ISBN 978-0-12-370867-0.

JH Van Hateren, Roland Kern, G Schwerdtfeger, and Martin Egelhaaf. Function and coding in the blowfly h1 neuron during naturalistic optic flow. *Journal of Neuroscience*, 25(17):4343–4352, 2005.

Bas Van Opheusden, Gianni Galbiati, Zahy Bnaya, Yunqi Li, and Wei Ji Ma. A computational model for decision tree search. In *Cognitive Science Society*, 2017.

Rufin VanRullen, Rudy Guyonneau, and Simon J Thorpe. Spike times make sense. *Trends in Neurosciences*, 28(1):1–4, 2005.

Eleni Vasilaki, Nicolas Frémaux, Robert Urbanczik, Walter Senn, and Wulfram Gerstner. Spike-based reinforcement learning in continuous state and action space: when policy gradient methods fail. *PLoS Computational Biology*, 5(12):e1000586, 2009.

Sabine Vollstädt-Klein, Svenja Wichert, Juri Rabinstein, Mira Bühler, Oliver Klein, Gabriele Ende, Derik Hermann, and Karl Mann. Initial, habitual and compulsive alcohol use is characterized by a shift of cue processing from ventral to dorsal striatum. *Addiction*, 105(10):1741–1749, 2010.

Pieter Voorn, Louk JMJ Vanderschuren, Henk J Groenewegen, Trevor W Robbins, and Cyriel MA Pennartz. Putting a spin on the dorsal–ventral divide of the striatum. *Trends in Neurosciences*, 27(8):468–474, 2004.

Florian Walter, Florian Röhrbein, and Alois Knoll. Computation by time. *Neural Processing Letters*, 44(1):103–124, 2016.

Jane X Wang, Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Demis Hassabis, and Matthew Botvinick. Prefrontal cortex as a meta-reinforcement learning system. *Nature Neuroscience*, 21(6):860–868, 2018.

Lei Phillip Wang, Fei Li, Dong Wang, Kun Xie, Deheng Wang, Xiaoming Shen, and Joe Z Tsien. Nmda receptors in dopaminergic neurons are crucial for habit learning. *Neuron*, 72(6):1055–1066, 2011.

Anne-Kathrin Warzecha, Jutta Kretzberg, and Martin Egelhaaf. Temporal precision of the encoding of motion information by visual interneurons. *Current Biology*, 8(7):359–368, 1998.

Mark J West, Paul D Coleman, Dorothy G Flood, and Juan C Troncoso. Differences in the pattern of hippocampal neuronal loss in normal ageing and alzheimer's disease. *The Lancet*, 344(8925):769–772, 1994.

Ian Q Whishaw. Formation of a place learning-set by the rat: a new paradigm for neurobehavioral studies. *Physiology & Behavior*, 35(1):139–143, 1985.

Sabrina Winter, Marco Dieckmann, and Kerstin Schwabe. Dopamine in the prefrontal cortex regulates rats behavioral flexibility to changing reward value. *Behavioural Brain Research*, 198(1):206–213, 2009.

Morgen Witzel and Malcolm Warner. Taylorism revisited: Culture, management theory and paradigm-shift. *Journal of General Management*, 40(3):55–70, 2015.

Henry H Yin and Barbara J Knowlton. The role of the basal ganglia in habit formation. *Nature Reviews Neuroscience*, 7(6):464–476, 2006.

Qiang Yu, Haizhou Li, and Kay Chen Tan. Spike timing or rate? neurons learn to make decisions for both through threshold-driven plasticity. *IEEE Transactions on Cybernetics*, 49(6):2178–2189, 2018.

Brian D Ziebart, Andrew L Maas, Anind K Dey, and J Andrew Bagnell. Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, pages 322–331, 2008.

Appendix A

Simulation details for the actor-critic architecture in chapter 4

A.1 Implementation of an actor-critic architecture for navigation to a fixed goal location

The agent was implemented using Julia. All codes and files to generate the figures are available in GitHub at `https://github.com/charlinetess/FMD_SMT/`.

Here, the model was implemented using a first order Euler method with $\mathrm{d}t = 0.1\text{ms}$. Relatively low learning rates (equations (4.10) and (5.26)) are required to guarantee the numerical stability of the implementation. The two learning rates are dependent on one another. To give appropriate feedback to the actor, the critic must first be accurate. To help achieve this, it is useful to set the critic learning faster than the actor. This way, the TD error can more rapidly reflect the actual change in value between two consecutive states. Here we choose $\chi_{\text{Actor}} = 0.01$ and $\chi_{\text{Critic}} = 0.1$. The discount factor γ is chosen as $\gamma = 0.98$. All the other parameter values are chosen as in Foster et al. [2000]: $R = 100$, $r = 5$, $t = 0.1s$, $\alpha = 30m/s$, $\sigma = 30$, $\beta = 2$, $\gamma = 0.98$.

Figure 4.2 shows the mean and standard deviation of the escape latencies of 30 independent agents implemented in the task. A trial starts at a random location sampled from $x_0 \in -95, 0, 95$ and $Y \in -95, 0, 95$. For every independent run a random goal location is sampled from $p \in (\pm 30, 0), (0, \pm 30), (\pm 0.5, \pm 0.5)$. Figure 4.6 shows similar measures obtained from the same runs with the same parameters, but with a change in the goal location from trial 20. Figures **??**, **??**, **??** and **??** show similar measures for respectively $\sigma = 70cm$, $\sigma = 5cm$, $\gamma = 0.1$, $\gamma = 0.98$. The value functions in figures 4.3a.i, 4.3b.i, 4.5c, 4.5b, 4.4c, and 4.4b were generated using a squared mesh grid of $\mathrm{d}x = 0.5$cm spatial step. Figures 4.3a.i and 4.3b.i shows the value function respectively before trial 1 and after trial 20, for the parameters mentioned above. 4.5c, 4.5b, 4.4c, and 4.4b were generated using respectively $\sigma = 70cm$, $\sigma = 5cm$, $\gamma = 0.1$, $\gamma = 0.98$. Figures 4.3a.ii and 4.3b.ii were plotted using the function quiver() showing the directions vector at locations uniformly spaced by $\mathrm{d}x = 10cm$ over the maze, respectively before trial 1 and after trial 20.

A.2 Implementation of an actor-critic architecture with coordinates-based goal directed navigation for the DMP task

The agent was implemented using Julia. All codes and files to generate the figures are available in GitHub at `https://github.com/charlinetess/FMD_DMP`. The coordinate agent was implemented using a first order Euler method using $\mathrm{d}t = 0.1$. The parameters were chosen as in A.1, with the following modifications and additions: discount factor: $\gamma = 0.99$, Learning rate for coordinates : $\chi_{\text{Coord}} = 0.05$, Eligibility trace parameter: $\lambda = 0.8$.

Figures 4.8a and 4.8b show the values of the estimated coordinates using a heatmap with a spatial step of $\mathrm{d}x = 0.5$cm at respectively trial 1 and trial 20. Figure 4.9a and 4.9b shows the mean and standard deviation of respectively the escape latencies and the search preference of 30 independent agents implemented in the task.

Appendix B

Simulation details for the continuous actor-critic architecture in chapter 5

The continuous agent was implemented using Julia using a first order Euler method with $\mathrm{d}t = 0.1$. The code will soon be available on GitHub.

Figures 5.23 shows equation (5.24). Figure 5.9 shows the mean and standard deviation of the escape latencies of 30 independent agents implemented in the task. A trial starts at a random location sampled from $x_0 \in -95, 0, 95$ and $Y \in -95, 0, 95$. For every independent run a random goal location is sampled from $p \in (\pm 30, 0), (0, \pm 30), (\pm 0.5, \pm 0.5)$. Figures 5.11a, 5.11a.ii, 5.11b and 5.11b.ii the terms in equation (5.12) in Cartesian and polar coordinates, at trial 1 and at trial 20.

For the figures mentioned above all parameters were the following: $r = 4$ (3.2.2.2); $\beta = 5$, $h = 1$, $\rho_0 = 0.5$ (5.7); $\sigma_{\mathrm{PC}} = 30$, $\rho_{\mathrm{PC}} = 1$ (5.8); $\varepsilon_{\mathrm{PC}} = 1$ (5.9); $\tau_m^{\mathrm{C}} = 0.099$, $\tau_s^{\mathrm{C}} = 0.099$ (5.10); $\nu = 0.99$, $V_0 = 0$ (5.11); $\varepsilon_{\mathrm{PA}} = 1$, $\varepsilon_{\mathrm{A}} = 1$ (5.12); $\tau_m^{\mathrm{A}} = 0.15$, $\tau_s^{\mathrm{A}} = 0.3$ (5.10); $w_- = -1$ $w_+ = 1$ $\xi = 1$ (5.13); $\tau_n = 0.7$ (5.14); $\sigma_n = 0.3$ (5.15); $\rho_n^o = 0.2$ (5.17); $\mu_n = 5$ (5.19); $\sigma_n^0 = 1$ (5.21); $\alpha = 30$ (5.22); $\tau_a = 1.5$, $\tau_b = 1.1$ (5.23); and the learning rates for the weight update (5.25) was chosen 0.1 and (5.26) 0.01.

Figures 5.12a, 5.12b and 5.12c were produced by changing ρ_n^o to respectively $\rho_n^o = 0.001$, $\rho_n^o = 0.5$ and $\rho_n^o = 2$. Figure 5.13a.ii, 5.13a.iii, 5.13b.ii and 5.13b.iii shows value functions using a squared mesh grid of $\mathrm{d}x = 0.5$cm spatial step and for $\rho_n^o = 0.001$.

Appendix C

Simulation details for the hierarchical agent in chapter 6

The hierarchical agent was implemented using Julia using a first order Euler method with dt = 0.1. All codes are available at `https://github.com/charlinetess/MetaTD`. I use ω=5, h=-0.2, ρ=2, τ_σ=15. The agent was first pretrained for 50 trials to each goal location using the model in A.1. Figure 6.9 was generated from 20 independent runs. Figure 6.4a, 6.4b, 6.4c and 6.4d were generated using respectively $\beta = 0.01$, $\beta = 0.2$, $\beta = 0.6$ and $\beta = 2$.

Appendix D

Supplementary material for chapter 7

D.1 Properties of eigenvalues of transition probability matrices

Here I demonstrate that for a transition probability matrix P, the eigenvalues satisfies $1 = |\lambda_0| \geq |\ \lambda_1| \geq ... \geq |\lambda_N|$. Consider $\lambda_k \in \lambda_0, ..., \lambda_{N-1}$ and its associated eigenvector φ_k. For every $n \in [1, \ldots, N]$, φ_k is also an eigenvector of the multi-step transition matrix P^n, *i.e.* $P^n \varphi_k = \lambda_k^n \varphi_k$. For every $i \in [1, \ldots, N]$ we can write:

$$||\lambda_k^n \varphi_k[i]|| = ||\sum_j P_{i,j}^n \varphi_k[j]|| = \sum_j P_{i,j}^n |\varphi_k[j]|, \text{ as } P_{i,j}^n \geq 0. \tag{D.1}$$

Defining $m = \max_j(\varphi_k[j])$ gives

$$|\lambda_k^n| m \leq \sum_j P_{i,j}^n m \leq M,$$

where M is a constant. Therefore $|\lambda_k| \leq 1$ for every $k \in [1, \ldots, N]$.

D.2 Additional eigenvectors

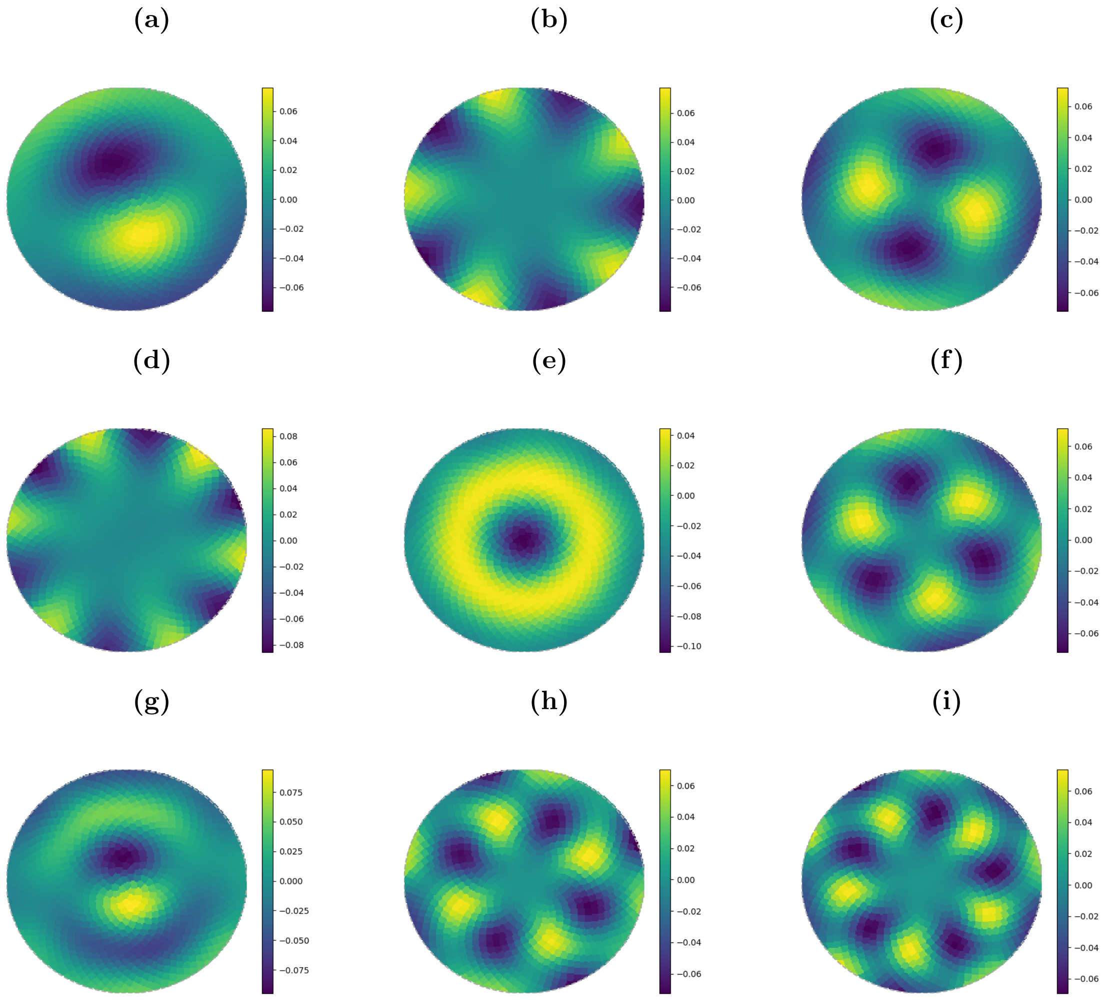


Figure D.1: Eigenvectors in order of decreasing associated eigenvalues. Indexes (a) 7 (b) 9 (c) 11 (d) 13 (e) 14 (f) 16 (g) 20 (h) 23 (i) 31. The spatial frequency increases with the index.

D.3 Simulation details for the predictive model in chapter 7

All the plots presented and the implementations were done using Julia and the codes are available on GitHub at `https://github.com/charlinetess/CorneilGertsner2015` and `https://github.com/charlinetess/SRandWatermaze`.

The model was implemented using a first order Euler method with d$t = 0.001$. The parameters are the following: $\tau = 1$, $g = 5$, $\gamma = 1$, $\alpha = \varepsilon = 0.05$. Figures 7.7a, 7.8a and 7.9a were generated using $q = 2$. Figures 7.7b, 7.8b and 7.9b were generated using

$q = 40$. Figures 7.10a, 7.11a and 7.12a were generated using $w_0 = 0.001$. Figures 7.10b, 7.11b and 7.12b were generated using $w_0 = 3 * \sqrt{(1-\gamma\lambda_1)^{-1}}$.

I used $n = 500$ neurons among $N = 1600$ states. The successor representation was computed using a discount factor $\gamma = 1$. Decoding weights were generated from the activity of $m = 200$ inputs randomly scattered throughout the environment, and using an inbuilt Julia function to compute the pseudoinverse, with the singular value parameter below 0.01 to prevent from overfitting.

Even though theoretically P should be diagonalisable, the diagonalisability of the matrix P depends on the width of the kernel A σ (equation (7.2)). If σ is big, the values of P are too small to be non-zeros depending on the precision of the machine and therefore the matrix will not be diagonalisable.