BIOLOGICALLY PLAUSIBLE COMPUTATIONS UNDERLYING ONE-SHOT LEARNING OF PAIRED ASSOCIATIONS

M GANESHKUMAR
(B.Sc. (Hons.), NUS)

A THESIS SUBMITTED FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

INTEGRATIVE SCIENCES AND ENGINEERING PROGRAMME, NUS GRADUATE SCHOOL

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2022

Thesis advisors:
Dr Yen Shih-Cheng, Main Thesis Advisor
Assistant Professor Tan Yong-Yi Andrew, Co-Advisor

Examiners:
Associate Professor Yeo Boon Thye Thomas
Associate Professor Zhou Juan

DECLARATION

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

M Ganeshkumar

11 December 2022

ACKNOWLEDGEMENTS

The work in this thesis could not have been accomplished without the generous help of many people. I would like to thank Shih-Cheng Yen, my main thesis advisor, for his invaluable support, patience, and advice in pursuing interdisciplinary research. I wish to thank Andrew Tan, my co-advisor, for his guidance, enthusiasm, mentorship, and for convincing me that physics is the most powerful schema.

The journey of learning different disciplines could not have been possible without my Thesis Advisory Council. I would like to thank Cheston Tan who taught me to use my pre-learned schemas to ask critical questions even in a new field, Camilo Libedinsky who made it seem research, including animal experiments, was easy and Thomas Yeo for giving me the confidence to say that I do not understand a concept.

I appreciate research fellows Tan Hui Min and Roger Herikstad for sharing their knowledge on the hippocampus and computational techniques as well as graduate student Evangelos Sigalas for looking at my weird equations. I would also like to thank the people I met during the MIT's CBMM summer school who made me believe intelligence was a problem worth solving.

Of course, this marathon could not have been possible without my peers Mark Seow, Sharmelee Selvaraji and Wan Lin Yew who kept my sanity in check by gleefully listening to my whims of riding around Southeast Asia on my motorcycle.

Most importantly, I would like to thank my family for moulding me into the person I am today. Specifically, my mother Mani Sukuna for her unconditional support and inspiring me to pursue a PhD while she pursed a bachelor's degree at the age of 50. To my father P Maniyarasu for teaching me patience, responsibility and to not take life for granted. I am thankful for my sisters Ma Diviyatharsini and Ma Priyaatharsini who had to deal with my craziness, forced me to take breaks and have a life outside research. I am also grateful for my wife Ang Li Yi for encouraging me to explore philosophy and for sharing my hopes to solve poverty, diseases, war, and climate change by improving intelligence through education.

TABLE OF CONTENTS

SUMMARY

One-shot learning is the ability to solve a problem after a single trial, a feat that animals and humans demonstrate daily. Although symbolic algorithms and recent deep learning algorithms can perform one-shot learning, they do not use biologically plausible learning rules. Hence, how the brain performs computations underlying one-shot learning remain elusive. In this thesis, I outline how biologically plausible neural circuits and learning rules can perform one-shot learning of new paired associations in a spatial navigation task. The phenomenon of one-shot learning has been studied as a part of various paradigms by different fields i.e. memory schemas in psychology and transfer-learning or meta-learning in computer science. A generally accepted perspective is that if new information can fit into a previously learned knowledge structure or schema, the novel task can be solved rapidly. Tse et al. (2007) trained rats to perform a two-part task. In the first part rats gradually learned multiple FLAVOUR-LOCATION paired associations. In the second part, rats were exposed to two new paired associations for a single trial, which they were able to recall in subsequent probe trials, demonstrating one-shot learning. Hence, I first developed a biologically plausible reinforcement learning agent that successfully learned multiple paired associations as in the first part of Tse's task. However, the agent was not able to learn new paired associations after a single trial as in the second part of Tse's experiment. Three schemas were missing from this agent, 1) the ability to learn a metric representation of current location 2) the ability to form new associations between relevant cues and the goal location after a single trial and 3) the ability to compute the direction to arbitrary goals from current location. I constructed an agent that learned (1) using place cell activity and self-motion information to compute and minimise a temporal difference error based on the principles of path integration. For (2), the synaptic weights between a reservoir of recurrently connected units and output units were trained using a reward-modulated Exploratory Hebbian plasticity rule to store and recall multiple goal coordinates after a single trial and for (3), a deep neural network was pretrained. With these biologically plausible schemas, I have demonstrated agents that, after an initial period of gradual learning, can navigate to multiple new goal locations after a single trial of learning, replicating the rodent behaviour results from Tse et al. (2007).

PUBLICATIONS INCLUDED IN THIS THESIS

Publications	Contents included in
• Kumar, M. G., Tan, C., Libedinsky, C., Yen, S.-C., & Tan, A. Y. Y. (2022). A Nonlinear Hidden Layer Enables Actor–Critic Agents to Learn Multiple Paired Association Navigation. *Cerebral Cortex*, 1–20. https://doi.org/10.1093/cercor/bhab456	**CHAPTER 2**
• Kumar, M. G., Tan, C., Libedinsky, C., Yen, S.-C., & Tan, A. Y.-Y. (2021). One-shot learning of paired associations by a reservoir computing model with Hebbian plasticity. *ArXiv Preprint ArXiv:2106.03580*. http://arxiv.org/abs/2106.03580 • Kumar, M. G., Tan, C., Libedinsky, C., Yen, S.-C., & Tan, A. Y.-Y. (2022). One-shot learning of paired association navigation with schemas and reward-modulated Hebbian plasticity. In preparation.	**CHAPTER 3**

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

α	β	γ	δ	ε	θ	π
alpha	beta	gamma	delta	epsilon	theta	pi

ρ	σ	τ	φ	ω	η	λ
rho	sigma	tau	psi	omega	eta	lambda

ν	ξ	ϕ	Δ	χ	Σ	$\in$
nu	xi	phi	change	chi	sum	Element of

∂	Θ	ς
Partial differentiation	Step function	Sigma variant

CHAPTER 1 Introduction

Imagine that you are going out on a date and when you present some flowers, your date frowns. On the subsequent date, you recall that your date dislikes flowers, and you present them with chocolates instead. This time, your date is ecstatic! From then on, you would present chocolates to this specific date.

This is an example of one-shot learning where we utilize our prior knowledge (that we should get either flowers or chocolates as a gift when going for a date) to solve a problem (chocolate is the better gift for date X) after a single episode. We perform such computations daily and yet how the brain performs these computations remains poorly understood.

To begin this inquiry, we turn to Marr's three levels of analysis (Marr 2010). He argues that if we want to understand the phenomenon of flight, looking only at the feathers of a bird will not be sufficient. Similarly, if we want to understand the biological computations underlying one-shot learning, looking only at the neuronal level will not be sufficient. Marr suggests decomposing the problem to three levels, namely

1) Computational: What is the problem the system is solving and why is it important?
2) Algorithmic: What algorithm or software is the system using to solve the problem?
3) Implementation: How is the algorithm implemented in the physiological hardware?

Some might argue that the three levels could be broken down to infinite levels, or that understanding all three levels is unnecessary. Nevertheless, this hierarchy is a good starting point towards attaining a holistic understanding (functional and mechanistic) of how a biological system performs one-shot learning.

In the next section, I attempt to categorise past works on one-shot learning into the three levels although the distinction between levels should be treated as a guide and not definitive.

1.1 Computational problem

The computational problem is to solve a novel task after a single episode. This could mean that the system initially requires several attempts to figure out the solution, but once a solution is found, the system should be able to replicate it in the following episode to demonstrate one-shot learning.

Genetic adaptation to changing environments or novel tasks happens over several generations, for example the proliferation of new COVID-19 strains albeit within a few months (Gu et al. 2020). However, genetic adaptation processes are very slow for animals and humans and instead the nervous system offers individuals the ability to adapt to an array of problems within a generation.

Understanding how the brain performs one-shot learning could offer us solutions to optimise education pedagogies and curricula to improve our adaptation capabilities (Carrell and Eisterhold 1983; Jitendra and Star 2011; Johansen 1997; McVee, Dunsmore, and Gavelek 2005), alleviate learning disabilities (Engineer, Hays, and Kilgard 2017; Nemeroff et al. 2006; Weingartner 1981) and at the same time, develop efficient algorithms that can solve other problems (Hassabis et al. 2017).

1.1.1 Memory schemas for one-shot learning

Memory schemas have been an appealing theory for how animals and humans demonstrate one-shot learning.

Bartlett was the earliest to demonstrate subjects using prior knowledge or schemas to reconstruct information to solve memory recall tasks (Bartlett and Burt 1932). Strikingly, when subjects could not recall the details of a narrated story, they filled in

the gaps using their own schemas which had subjective biases (Anderson and Pichert 1978; Barto, Sutton, and Anderson 1983; Darley and Gross 1983; DiMaggio 1997).

For example, subjects were briefly shown an image of a bathroom and they were more likely to recall the presence of schema congruent object such as a toilet and sink, although the sink was absent. Moreover, subjects failed to recall schema incongruent objects such as the radio and flower vase, although the flower vase was present. (Brewer and Treyens 1981; Graesser and Nakamura 1982; Webb and Dennis 2019).

Minsky and Rumelhart conceptualised schemas as knowledge frameworks that organized how information was to be associated or processed. Importantly, schemas contained empty placeholders in which to slot new information while specific rules governed what information could fit the respective placeholders (Minsky 1974; Rumelhart, Smolensky, and McClelland 1987). A schema with either some or all placeholders filled can be used to reconstruct information and serve as a guide for inference (Rumelhart 1980; Rumelhart and Ortony 1977).

For example, the NAVIGATE schema (Fig. 1.1 left) describes the process of moving from an arbitrary location X to a goal location Y by selecting appropriate actions from subschema ACTIONS that contains directions of movement. This single uninstantiated schema can be used to generate numerous instantiations for goal-directed movement by slotting appropriate information into the placeholders.

Instantiated schemas allow us to make inferences. For example, once a suitable ANIMAL schema is populated, we can make associative inferences that Salmon and Robin are red, but Salmon is a FISH and Robin is a BIRD.

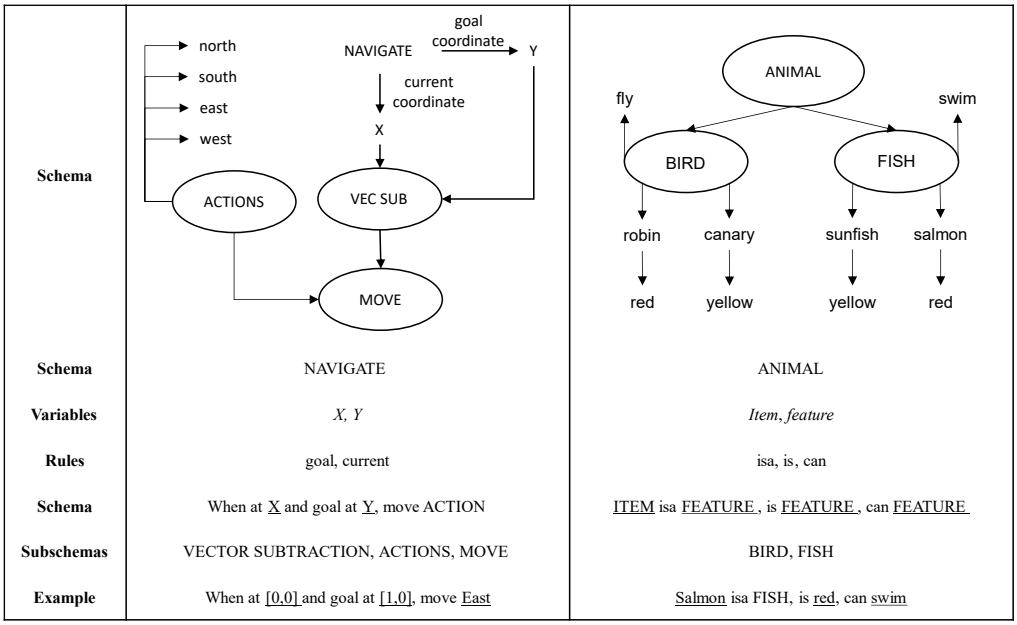


Figure 1.1. A single episode is sufficient to fill schema placeholders for inference. Schemas facilitate one-shot learning where a single episode is sufficient to fill empty placeholders with the relevant information to make an inference. Left: NAVIGATE schema describes which actions can be taken to navigate to a goal from arbitrary locations using placeholders Y and X respectively. Right: ANIMAL schema describes the similarities and differences between bird or fish based on associations.

When a placeholder is unfilled, the rules governing the placeholder can be used to prompt a question, such as "What is the goal coordinate to navigate to?" Once the relevant information becomes available, it can be slotted into the schema after a single episode to demonstrate one-shot learning.

Information that could fit a schema was termed schema congruent such as finding a sink in a toilet, while those that did not comply with the rules of the placeholders were termed as schema incongruent (McClelland 2013), such as finding a flower vase in a toilet.

Piaget's influential work on child development noted that when children come across schema incongruent information, they experienced cognitive dissonance or disequilibrium. To reach equilibrium, children resorted to either assimilation, which is to modify the information to fit their schema, or accommodation, where they modified

their schema to fit the information (Piaget, Inhelder, and Chipman 1976; Zhiqing 2015). This process extends to adults, for example a person who strongly believes the earth is flat is likely to modify experimental evidence supporting a spherical earth to fit their flat earth schema.

Schemas can also be conceptualised as task rules which inform how to solve a task. A notable experiment demonstrated monkeys gradually learning to choose the correct image out of two options. The two-image combination was replaced by two completely new images after six trials. The monkeys gradually learned the rule "If A was rewarded in previous trial, then choose A, else choose B". Subsequently, monkeys used this rule to choose the correct image after the first trial even though it was a new two-image combination (Harlow 1949), demonstrating one-shot learning.

More recently, Tse and colleagues developed a two-part rodent experiment where in the first part, rats gradually learned the rule "Flavour X is at Location Y" to associate the given flavour cue with one of several reward locations. In the second part, rats only required a single trial to learn multiple new flavour-location associations and subsequently used the instantiated schema to recall the goal location corresponding to the flavour cue (Tse et al. 2007).

1.1.2 Alternative theories for one-shot navigation

An alternative theory for one-shot navigation is cognitive maps. Tolman demonstrated that rats which freely explored a maze were able to rapidly navigate to a newly given goal location using the shortest path, similar to rats that were trained to navigate to the goal location since the start of the experiment (Tolman 1948). This was evidence that rats learned a latent mental representation of the environment during their free exploration that allowed them to plan their trajectory once a goal was available. Cognitive maps have been predominantly used to explain one-shot navigation in spatial navigation problems, although a recent proposal suggested that cognitive maps can act

like memory schemas (Preston and Eichenbaum 2013) and used to solve nonspatial associative and transitive inference tasks (Whittington et al. 2020). Hence, cognitive maps could be a subset of schema since schemas have also been used to explain spatial navigation (Arkin 1989; Tse et al. 2007), language comprehension (Rumelhart 1975; Rumelhart and McClelland 1982; Rumelhart and Ortony 1977) and recognition tasks (Brewer and Treyens 1981; Palmer 1975; Webb and Dennis 2019).

In summary, memory schemas offer a framework to organise new information to facilitate rapid learning. By composing pre-learnt schemas, it reduces the need to learn each task as an individual problem, allowing the system to rapidly solve novel problems by either adapting new information or the schema. The next section surveys algorithms that can be used to model memory schemas for one-shot learning.

1.2 Algorithmic level

Although there are several classes of algorithms that demonstrate one-shot learning, we will focus on algorithms that solve spatial navigation tasks where an agent needs to navigate to one or multiple goals and receives a reward only after reaching the correct goal location.

1.2.1 Symbolic algorithm for one-shot learning

Symbolic cognitive architectures such as SOAR (Laird, Newell, and Rosenbloom 1987) and ACT-R (Anderson 2013; Lebiere and Anderson 1993) were developed to replicate the adaptive computations performed by the brain, with ACT-R focused on modelling brain structures and fitting model predictions to behavioural data (Borst et al. 2015; Fu and Anderson 2006).

The Common Model of Cognition (Laird 2021) is a general cognitive framework which calls for modular systems (Fig. 1.2A) to perform specific functions. The perception module extracts relevant features from different input modalities and passes them to

the working memory module which compares the features against episodic memories or productions in procedural memories to determine relevant actions for the motor module to execute.

Importantly, these models use learning algorithms to convert newly rewarded actions into chunks and are stored in procedural memory as productions. These productions or action strategies can be retrieved from procedural memory to solve similar tasks (Laird 2021).

Algorithm 1 Pseudo-code for symbolic cognitive agent to solve reinforcement learning task

```
Initialise working memory, episodic memory, and procedural memory
    for state in environment
        Extract features from input into working memory
        Propose operations against episodic and procedural memory
        Apply operation to perform based on input features
        Execute operation using motor module as action
        if state is rewarded then
            Store state and operation in episodic memory
    if impasse was detected and solved then
        Convert sequence of operations into a new production
        Store production in procedural memory
end
```

Take the problem of choosing the correct present for a date. To simplify the problem, there is one state and two actions, to present either flowers F or chocolates C to date X. After initialising the system, the perception module extracts relevant features of your date X, and the working memory module will compare these features against all your previous dates stored in the episodic memory to recall specific episode of date X frowning when presented with flowers. Since gifting flowers was negatively valued,

the working memory module chooses the alternative action, which is to present chocolates. The motor module then executes a sequence of actions to gift the chocolates to date X. If the date is pleased, the association of this date to chocolates X–C is positively valued, and the sequence of actions are added to the episodic and procedural memory respectively. In the subsequent date, the system will recall the X–C association that gifting chocolates is valued higher than gifting flowers and the relevant production will be selected.

Although these cognitive systems can solve problems flexibly, these computations are performed symbolically i.e., expert or rules-based systems that use IF-THEN conditions. Hence, the representations and learning processes in symbolic models are hard to compare against biological processes.

1.2.2 Model-free versus model-based reinforcement learning algorithms

Reinforcement learning (RL) algorithms were developed on the theory of operant conditioning where an agent learns a sequence of actions based on rewards and punishments (Skinner 1963). Similarly, the goal of RL is to maximise cumulative rewards especially in tasks with temporally sparse rewards, like the dating problem, by learning an optimal sequence of actions (Sutton and Barto 2020).

A default implementation of RL is to use model-free algorithms because of their computational efficiency. They learn a value function that describes how rewarding the current states is to influence the policy that specifies the actions to take at each state. No further knowledge of the task is needed. When powerful function approximators such as deep neural networks are trained to learn value and policy functions, model-free algorithms can supersede human performance in solving complex RL tasks (Mnih et al. 2015, 2016).

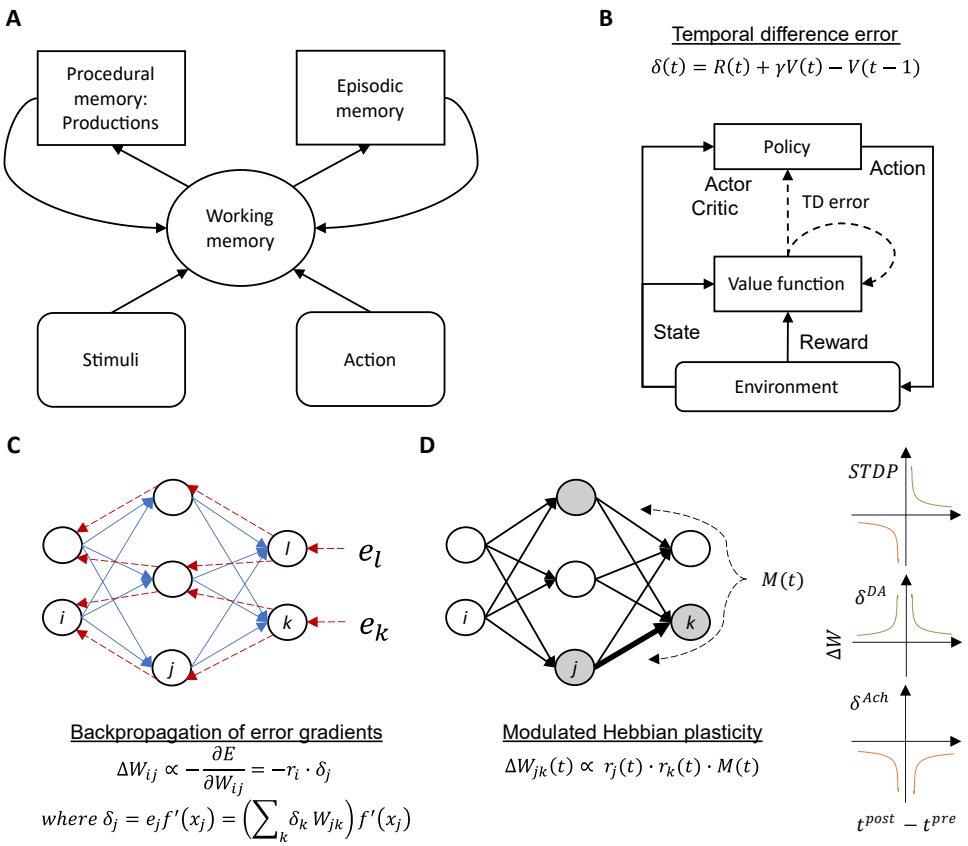


Figure 1.2. Overview of architectures and learning algorithms. A) Architecture of symbolic cognitive architecture with working memory, procedural memory and episodic memory. B) Actor-Critic architecture takes state and reward information as inputs to learn optimal value function. The temporal difference error is computed using the value function and is used to update the synaptic weights of the actor and critic. C) Training synaptic weights using backpropagation is to propagate the vector error from the last layer to the first layer. Non-local information is necessary to determine the weight update. D) Hebbian learning uses local presynaptic and postsynaptic neuron firing activity to compute the weight update while neuromodulatory factors such as dopamine and acetylcholine influence the direction of weight change.

Value-based methods such as Q-learning and SARSA directly learn the value of actions in each state and the policy is to select the action with the highest value in the respective states. Although this algorithm converges quickly, it struggles on high dimensional tasks, or tasks with variable reward structures. Policy-gradient methods like REINFORCE learn to optimise the policy without learning a value function but show better convergence in high dimensional tasks, although this method is sample

inefficient. The Actor-Critic algorithm (Fig. 1.2B) combines the benefits of both techniques by iteratively learning the value function and using it to gradually learn a policy (Sutton and Barto 2020). Several proposals have mapped the actor-critic algorithm to the basal ganglia (Barto 1995; Joel, Niv, and Ruppin 2002; Niv 2009; Yin and Knowlton 2006).

For the critic to learn a value function, a temporal difference (TD) error (Fig. 1.2B) is computed using the Bellman equation (Bellman 1954), by taking the difference between the critic's value estimation of the current and future state coupled with the true reward structure. A corresponding policy is learned by using the TD error to assign credit to actions that either increase or decrease cumulative reward (Sutton and Barto 2020). The TD error can be minimised using a variety of learning algorithms (Foster, Morris, and Dayan 2000; Frémaux, Sprekeler, and Gerstner 2013; Mnih et al. 2016). It has been suggested that the TD error is encoded by the dopamine neuromodulator as a reward prediction error for operant conditioning (P Read Montague, Dayan, and Sejnowski 1996; Niv 2009; W Schultz, Dayan, and Montague 1997), though recent proposals also suggest dopamine encodes other forms of reinforcement learning errors as well (Akam and Walton 2021; Dabney et al. 2020; Gardner, Schoenbaum, and Gershman 2018; Gershman 2018; Sharpe et al. 2017).

However, model-free algorithms need to unlearn and relearn a new value function and policy when state transitions or reward structures change even slightly. Hence, these algorithms do not offer the flexibility to adapt to novel problems (Dolan and Dayan 2013; Kansky et al. 2017).

In contrast, model-based RL algorithms learn an internal model of the state transitions and reward structure of the task. Once an adequate model is learned, the algorithm can use this internal model to plan an optimal policy using a decision tree (Gläscher et al. 2010) to solve novel variations of the task (Fu, Levine, and Abbeel 2016; Piray and

Daw 2021). Alternatively, Dyna is a hybrid model-free and model-based algorithm that uses the internal model to generate state transitions and rewards to update its value and policy functions without needing to sample from the actual task (Sutton and Barto 2020). In this way, model-based algorithms can demonstrate one-shot learning if the reward structure can be learned after a single trial (Gardner et al. 2018; Momennejad et al. 2017).

1.2.3 Using neural networks for function approximation

Multi-layered neural networks can be trained to solve complex cognitive tasks, end-to-end using backpropagation of error signals. After determining the output error, the error gradients specific to each synapse are computed and applied down the hierarchy of layers (Fig. 1.2C). Neural networks trained by backpropagation can resemble neural representations observed in the visual stream (Kar et al. 2019; Rajalingham et al. 2018), prefrontal cortex (Mante et al. 2013; Wang et al. 2018; Yang et al. 2019), entorhinal cortex (Banino et al. 2018; Cueva and Wei 2018), and parietal cortex (Suhaimi et al. 2022), and may help us to understand how biological systems perform computations (Yang and Wang 2020).

Neural networks can also be trained by backpropagation to read and write episodic memories into an external symbolic memory system (Botvinick et al. 2019) or another recurrent neural network (Ramsauer et al. 2020) to solve associative recall tasks (Graves, Wayne, and Danihelka 2014), demonstrate one-shot learning for classification (Ritter et al. 2018; Santoro et al. 2016) and one-shot navigation (Banino et al. 2018; Team et al. 2021; Wayne et al. 2018).

However, backpropagation appears difficult or impossible to map to a biologically plausible learning mechanism (Lillicrap et al. 2020). This is because backpropagation can be nonlocal or acausal. Specifically, backpropagation uses the same weights for both the forward pass and backward error propagation though biological neural

synapses are uni-directional. Furthermore, the error gradient specific to each synapse is computed though this appears to have no biological counterpart. More importantly, the concept of time is non-existent during training as backpropagation of error happens instantaneously over several epochs (Hunsberger, Orchard, and Wong 2017). Hence, even though representations learned using backpropagation are similar to those of the brain, backpropagation does not allow us to understand how biological neural circuits use synaptic plasticity for learning.

Instead of backpropagation, contrastive Hebbian learning uses local firing activity and does not require explicit error gradients to train multi-layer neural networks (Hwu and Krichmar 2020). However, this requires supervised different network dynamics and synaptic plasticity rules in different states, which is also not biologically plausible although neural architecture trained by this strategy are able to demonstrate one-shot learning of new flavour cues and goal location paired associations.

1.2.4 Biologically plausible learning algorithms

Instead of using backpropagation to train the synaptic weights of artificial neural networks, Hebbian learning is accepted to be a biologically plausible synaptic plasticity rule (Fig. 1D). The synaptic connections are either strengthened or weakened based on the localised firing activity of the presynaptic and postsynaptic neurons. This rule is often heuristically stated as “Neurons that fire together, wire together”, meaning that the synaptic weight of neurons with correlated firing activity will strengthen.

This theory has been experimentally validated, and expanded in some cases to Spike Time Dependent Plasticity (STDP) (Caporale and Dan 2008; Markram et al. 1997) where co-activating presynaptic and postsynaptic neurons within certain spike timing windows either strengthens or weakens the synaptic weights to cause long term potentiation (LTP) or depression (LTD) depending on the temporal order of presynaptic and postsynaptic activations.

The direction of weight update can be further modulated by global neuromodulatory factors (Fig. 1.2D) such as dopamine (Frémaux and Gerstner 2016; Frémaux et al. 2013) or acetylcholine (Brzosko et al. 2017). The presynaptic activity, postsynaptic activity and neuromodulatory factor enter the 3-factor Hebbian plasticity rule which can be used to solve classification (Hoerzer, Legenstein, and Maass 2012; Lindsay et al. 2017; Maass, Joshi, and Sontag 2007; Miconi 2017; Xie and Seung 2004) and navigation problems (Arleo and Gerstner 2000; Brown and Sharp 1995; Foster et al. 2000; Frémaux et al. 2013; Legenstein et al. 2010).

The synaptic weights within a recurrent neural networks can also be trained using the Hebbian rule to store and recall patterns after a single trial (Hopfield 1982, 1984), replicating the pattern completion function of hippocampal CA3 system (Rolls 2013). However, there has been limited work on training neural networks using Hebbian plasticity to solve complex cognitive tasks, let alone demonstrate one-shot learning.

1.2.5 Alternative learning strategies

Recently, there have been alternative strategies that straddle between backpropagation and Hebbian rule. Some address the backward propagation of error signals by using a separate neural network to directly send the teaching signal to each synapse (Bellec et al. 2020; Lillicrap et al. 2016; Murray 2019). However, this architecture is yet to be experimentally validated. Others use local presynaptic and postsynaptic information (Bellec et al. 2019; Scherr, Stöckl, and Maass 2020), although the modulatory factor uses synapse specific error gradients computed using backpropagation, unlike using a global neuromodulatory factor.

Another approach is to use backpropagation and a Hebbian rule to separately update two sets of synaptic weights. The meta-learning framework involves two training loops where the outer loop trains the synapses of deep neural networks using backpropagation while the inner loop uses a Hebbian rule to update another set of synapses to perform

one-trial associations (Limbacher and Legenstein 2020; Scherr et al. 2020; Whittington et al. 2020). However, most of the complex computations are performed by synapses trained using backpropagation while the network trained using a Hebbian rule only stores episodic memories.

Hence, to my knowledge there are no neural network architectures that has demonstrated rapid learning on complex cognitive tasks with synapses trained using biologically plausible rules.

1.3 Implementation level

The implementation level looks at biological neural circuits involved in one-shot learning during spatial navigation. Since the focus is on learning computations, brain regions whose functions are more usually associated with representational computations such as the visual and parietal cortex are beyond the scope of the present discussion.

1.3.1 Striatum for reinforcement learning

The basal ganglia has been shown to be crucial for stimulus-response learning and mappings of the model-free actor-critic reinforcement learning algorithms to specific regions have been proposed (Barto 1995; Joel et al. 2002; Yin and Knowlton 2006). Typically, the ventral striatum takes in state information from the hippocampus or the cortex and is proposed to learn state values while the dorsal striatum learns the stimulus-response associations (Graybiel 2008; Lipton, Gonzales, and Citri 2019; O'Doherty et al. 2004); some proposals argue that the dorsomedial striatum is involved in goal-directed learning (Balleine and Dickinson 1998; Yin et al. 2005).

Synaptic plasticity in the striatum is modulated by dopamine from midbrain dopaminergic neurons which encode reward prediction errors (P Read Montague et al. 1996; W Schultz et al. 1997) that reflect the difference in expected and actual reward

outcomes. A higher magnitude of phasic dopaminergic neuron activity represents a positive prediction error (Bayer and Glimcher 2005) that drives the learning of positive state values and assign credit to actions.

1.3.2 Hippocampus for one-shot learning of episodic memories

The hippocampus remains a primary region of interest when studying learning and memory (Eichenbaum 2004; McNaughton and Morris 1987; Moser, Rowland, and Moser 2015). While the dentate gyrus (DG) receives and integrates inputs from the cortex, the granule cells within the DG perform pattern separation to minimise overlap between different stimuli and convey the information to the CA3 (Rolls 2013). Both CA1 and CA3 encode spatial (McKenzie et al. 2014; O'Keefe and Dostrovsky 1971), non-spatial (Gulli et al. 2020) and temporal (Dragoi and Buzsáki 2006; Dragoi and Tonegawa 2011) variables. Most notably, CA1 and CA3 cells encoding spatial information are called place cells and have Gaussian tuning curves (O'Keefe and Burgess 1996).

More importantly, recurrent connectivity and synaptic plasticity within the CA3 system is postulated to enable it to function as an autoassociative network capable of one-shot association between different stimuli or rewards (Guzman et al. 2016; Rolls 2007). Once an associative memory is formed, when fragments of a related stimulus are given, the attractor dynamics within the CA3 system performs pattern completion to recall the entire memory (Neunuebel and Knierim 2014; Rolls 2013), similar to the Hopfield network.

Newly formed episodic memories are only temporarily hippocampus-dependent, perhaps indicating that they are only initially stored in the hippocampus. With time, relevant memories are organized and permanently stored in the cortex, becoming hippocampus-independent through a process called memory consolidation (Squire et al. 2015; Yonelinas et al. 2019). These consolidated semantic memories encompass

knowledge frameworks or schemas that guide subsequent learning and behaviour (Baram et al. 2021; Van Kesteren et al. 2012; Kumaran, Hassabis, and McClelland 2016; McClelland 2013; McKenzie and Eichenbaum 2011; Preston and Eichenbaum 2013). Interestingly, the firing of CA1 and CA3 neurons in a novel environment are consistent with that of a familiar environment (Baraduc, Duhamel, and Wirth 2019; McKenzie et al. 2014), suggesting that the encoding of new information by the hippocampus is organized against the backdrop of a schema consolidated in the familiar environment.

1.3.3 Entorhinal cortex for navigation

While hippocampal place cells play a central role in learning, the entorhinal cortex (EC) has been shown to be necessary, particularly in spatial navigation. Entorhinal grid cells are also place modulated but they exhibit periodic or hexagonal grid-like firing activity in one-dimensional and two-dimensional mazes respectively (Hafting et al. 2005). Place cells undergo either rate remapping or global remapping when there are small or significant changes in environmental cues. On the other hand, grid cell activity remains fairly consistent (Fyhn et al. 2007) as it is hypothesized to be solely based on the animal's self-motion information (McNaughton et al. 2006) to perform path integration (Burak and Fiete 2009; Fuhs and Touretzky 2006; Widloski and Fiete 2014) and navigation to goals (Banino et al. 2018; Giocomo, Moser, and Moser 2011; Sosa and Giocomo 2021). Object-vector (Høydal et al. 2019) and border cells (Solstad et al. 2008) in the EC have also been hypothesized to support the animal's ability to self-localise. However, it has also been suggested that grid cells alone cannot be used for goal directed navigation. Rather, the stability of grid fields endow the animal with an error-correcting mechanism to learn a separate metric representation for a more efficient vector-based navigation that does not require the animal to search through past trajectories (Bush et al. 2015; Fiete, Burak, and Brookings 2008).

1.3.4 Prefrontal cortex for learning schemas

The prefrontal cortex (PFC) is crucial for learning knowledge frameworks which facilitate cognitive control and flexibility. The prefrontal cortex may learn abstract task rules or schemas (Mansouri, Freedman, and Buckley 2020) and exert top down control on other brain regions to facilitate efficient mappings between inputs and actions (Miller 2000; Miller and Cohen 2001). When encoding new information after a schema is learned, the angular gyrus, hippocampus and the prefrontal cortex show higher BOLD signals (Gilboa and Marlatte 2017) while the hippocampus shows similar spatial tuning (Baraduc et al. 2019), although the exact circuit mechanisms are not known.

Studies have shown that as animals gradually learn the task rule or schema, the PFC neural dynamics gradually shifts from a noisy high dimensional to a low dimensional abstract representation to describe the critical task variables (Bernardi et al. 2020; Mack, Preston, and Love 2020; Mante et al. 2013; Zhou et al. 2020). This convergence of network dynamics also has been reported in biologically plausible networks learning task rules (Hoerzer et al. 2012; Miconi 2017).

Both the anterior cingulate cortex (ACC) (Akam et al. 2021; Kennerley et al. 2006) and orbitofrontal cortex (OFC) (Schuck et al. 2016; Wilson et al. 2014) have been shown to be involved in model-based reinforcement learning where the transition statistics and value of actions are evaluated for goal-directed behaviour. It has also been suggested that the ACC monitors conflicts against the learned rule by performing error detection (Botvinick et al. 1999; Carter et al. 1998) and regulates behaviour for error correction (Bush, Luu, and Posner 2000; Devinsky, Morrell, and Vogt 1995). Instead, it has been suggested that since the OFC integrates information from various sensory modalities, hippocampus and amygdala while projecting to the striatum and ventral tegmental area that regulates dopamine (Rolls 2000, 2004; Rolls, Cheng, and Feng 2020), it learns task specific value functions and computes dopamine based prediction errors to rapidly map

novel information into pre-learned rules (Banerjee et al. 2020). In addition, when performing schema dependent tasks, studies describe a higher ventromedial PFC coupling to hippocampal activity to increase coordination of encoding between the two regions (Baram et al. 2021; Van Buuren et al. 2014; Gilboa and Marlatte 2017; Van Kesteren et al. 2010, 2012).

Although some work has tried to delineate the function of each region in the PFC, there is increasing evidence that similar computations may happen in PFC regions (Mansouri et al. 2020). Hence, an alternative proposal conceives the PFC as a hierarchy of function instead of modular networks that perform distinct functions (Maisson et al. 2021; Riley et al. 2018; Stalnaker, Cooch, and Schoenbaum 2015).

Besides learning schemas, the PFC is also known to be responsible for selectively attending to relevant information (Baddeley 2012; Kane and Engle 2002), maintaining it in working memory for further manipulation (Curtis and D'Esposito 2003; Parthasarathy et al. 2019; Wimmer et al. 2014) and planning (Ehrlich and Murray 2021; Hoshi and Tanji 2004; Hunt et al. 2021) such as for movement (Tang et al. 2020). PFC neurons are found to have mixed selectivity (linear and nonlinear) to various task variables (Parthasarathy et al. 2017; Rigotti et al. 2013). This increases the dimensionality of neural representations to improve discriminability of information by other brain regions (Fusi, Miller, and Rigotti 2016). Theoretical work has shown that the discriminability–generalisation trade-off could be balanced (Barak, Rigotti, and Fusi 2013), but how the PFC solves this problem is yet to be determined.

1.4 Notable examples of one-shot learning by rodents

In a notable water maze experiment, the task was to navigate to a hidden platform that was displaced to a new location every four trials. After an initial acquisition period, rats showed significant savings in latency between the first and second trials, even after the platform was displaced, demonstrating one-shot navigation to the newly displaced

location. Hippocampal lesions significantly affected the rats' one-shot learning ability. Interestingly, blocking hippocampal NMDA receptors using D-AP5 did not show significant impact on one-shot learning if the trials were 15 seconds apart. However, if the trials were 20 minutes or 2 hours apart, rats did not demonstrate one-shot learning. This suggests that hippocampal plasticity was necessary to consolidate the goal location information into long term memory for recall after a longer interval (Steele and Morris 1999). Subsequent computational modelling replicated the one-shot learning results by developing agents that gradually learned a metric representation to self-localize using dead reckoning, stored goal coordinates in memory and had a navigation module that performed vector subtraction between the agent's current and goal coordinates to decide the direction to move (Foster et al. 2000). However, goal coordinates were stored symbolically after a single trial, and the navigation module was a symbolic function. This work showed that the gradual learning of a metric representation and one-shot learning of goal location could underlie one-shot navigation to single goals.

Tse and colleagues increased the complexity of the navigation task by developing a two-part multiple paired association (MPA) task. In the first part, rats were given one out of six possible flavour cues in the start box and had to navigate to the correct location in an open field arena which had six sand wells as possible reward locations. Rats gradually learned the Original FLAVOUR–LOCATION Paired Association (OPA) task over 20 sessions while hippocampal lesioned rats could not. Interestingly, when hippocampal lesions were introduced after learning OPA, rats were able to recall the correct flavour–location combination in the subsequent probe trial. By introducing either CNQX or D-AP5 to the prelimbic region, Tse demonstrated that the recall of flavour–location combinations could be blocked, suggesting that the initially hippocampus dependent OPA knowledge had consolidated to the prelimbic region.

In the second part, two of the six original flavour-location pairs were replaced with two New Paired Associates (NPA) and the rats trained on OPA were introduced to the two NPA combinations, each for a single trial. The next trial was a nonrewarded probe, but rats navigated to the correct NPA locations, demonstrating one-shot learning of two new flavour-location pairs. Hippocampal lesions and administration of D-AP5 to the prelimbic region prior to the learning of NPAs or administration of CNQX after learning NPAs affected one-shot learning of NPA. Hippocampal lesions, 3 hours instead of 24 hours after learning NPA affected the one-shot learning of NPA as well. This meant that both the hippocampus and the prelimbic regions were needed to learn the new flavour-location associations after a single trial and but only the prelimbic region was needed for recall.

Interestingly, when the rats trained on the OPA were introduced to a New Maze (NM) condition with new environmental cues and flavour–location combinations, while the task rule of associating flavour cues to location was kept the same, rats could not learn and recall the six new paired associates after a single trial and took another 20 trials sessions to reach performance criteria. The hypothesis was that when rats were placed in a new environment, place cells underwent global remapping, and they could not use the previously learned schema to solve NM. However, the mechanism for the failure to demonstrate one-shot learning was not determined, though the prelimbic region was not as involved in encoding NM compared to when encoding the NPAs.

1.5 Gaps in current research

I have briefly outlined past work that characterised the general computations, computational models to replicate, and the brain regions involved in one-shot learning, before concluding with two notable animal experiments. Given this overview, there are still some gaps in our understanding of how the brain performs one-shot learning.

If the brain indeed uses schemas for efficient learning, how are schemas represented in the biological neural circuits? We require several schemas, each performing a different function. Given a symbolic description of a schema, we would further like to understand how neurons and synapses integrate new information for one-shot learning. Neuroimaging studies point to the hippocampus and the prefrontal cortex but lack the temporal resolution on the computations performed at each time step. Trying to understand the circuit narrative using electrophysiology experiments remains difficult. Although deep learning offers us several solutions, trying to characterise the diverse schema-like computations performed by a large network is difficult. Perhaps training smaller networks that perform a single function and fitting them modularly could give us insights on how different schema networks work cooperatively to solve a problem.

Secondly, how are schemas used for learning? Schemas are formalised as knowledge structures with placeholders that when filled, can be used to make inference. How does this explain the initial gradual learning of a task? Are schemas gradually learned from scratch for each task or is the animal using pre-existing schemas to learn a task specific representation that can be generalised within the domain of the task? For example, Foster et al. (2000) demonstrated that agents need to gradually learn a stable metric representation using path integration before the agent can accurately perform vector subtraction to move towards the goal. Would the learning of a metric representation of an environment or the ability to infer direction of movement despite any goal information constitute as a schema? Solutions offered by backpropagation are not informative of how the biological circuits learn. Hence, what is the learning process with respect to a schema?

To not fall into the fallacy that schemas are the way the brain demonstrates efficient learning, the strategy is to develop a biological model that is devoid of the hypotheses and assumptions of how schemas compute. Instead, the main outcome is to train neural network models using biologically plausible learning rules to demonstrate both gradual

and the subsequent one-shot learning behaviour in a complex task, such as Tse et al. (2007).

In this thesis, I first describe a biologically plausible reinforcement learning agent that gradually learns the first part of the multiple paired associations task. Thereafter, I describe three neural schemas and demonstrate how their composition demonstrates one-shot learning of multiple new paired associations. I conclude the thesis with a summary of contributions, limitations, and future directions to understand how the brain performs one-shot learning.

CHAPTER 2 A nonlinear hidden layer enables actor-critic agents to learn multiple paired association navigation (Kumar et al., 2022)

(The contents of this chapter have been published. Please refer to page VI for details.)

Abstract

Navigation to multiple cued reward locations has been increasingly used to study rodent learning. Though deep reinforcement learning agents have been shown to be able to learn the task, they are not biologically plausible. Biologically plausible classic actor– critic agents have been shown to learn to navigate to single reward locations, but which biologically plausible agents are able to learn multiple cue–reward location tasks has remained unclear. In this computational study, we show versions of classic agents that learn to navigate to a single reward location, and adapt to reward location displacement, but are not able to learn multiple paired association navigation. The limitation is overcome by an agent in which place cell and cue information are first processed by a feedforward nonlinear hidden layer with synapses to the actor and critic subject to temporal difference error-modulated plasticity. Faster learning is obtained when the feedforward layer is replaced by a recurrent reservoir network.

2.1 Introduction

Navigation to remembered locations is important for many animals (Healy and Hurly 1995; Menzel and Müller 1996). Tasks like the Barnes maze and the Morris water maze requiring navigation to a single reward location are often used to study rodent learning (Barnes 1979; Hok et al. 2007; Hok, Save, and Poucet 2005; Jackson, Johnson, and Redish 2006; Jackson and Redish 2007; Morris et al. 1982; Rossier et al. 2000). More recently, there has been increasing use of a multiple paired association navigation task for rodents involving more than one reward location (Bethus, Tse, and Morris 2010;

Day, Langston, and Morris 2003; Kakeyama et al. 2014; Kesner, Hunsaker, and Warthen 2008; Spiers, Olafsdottir, and Lever 2018; Tse et al. 2007, 2011; Wang, Tse, and Morris 2012). The multiple paired association task takes place in an arena where the reward is hidden. Each trial starts with the animal in one of several positions at the arena boundary, where the animal receives one of several sensory cues, such as a particular odour. Each sensory cue consistently represents a possible reward location, and indicates where the animal must go to obtain a reward.

Deep reinforcement learning algorithms have progressed considerably to show human level performance in computer games and other remarkable capabilities, and provide useful frameworks for interpreting brain function (Banino et al. 2018; Botvinick et al. 2020; Dabney et al. 2020; Mnih et al. 2015, 2016; Song, Yang, and Wang 2017; Wang et al. 2018). However, deep reinforcement learning uses gradient descent algorithms that do not seem to correspond to any biologically-plausible learning rule (Botvinick et al. 2020). Physiological experiments suggest that synaptic plasticity in reinforcement learning is a function of presynaptic activity, postsynaptic activity, and globally available teaching signals carrying reward information (Bakin and Weinberger 1996; Brzosko, Mierau, and Paulsen 2019; Brzosko, Schultz, and Paulsen 2015; D'amour and Froemke 2015; Dennis et al. 2016; Froemke, Merzenich, and Schreiner 2007; He et al. 2015; Karachot et al. 2001; Kilgard 1998; Palacios-Filardo and Mellor 2019; Reynolds JNJ, Hyland BI, and Wickens JR 2001; Seol et al. 2007). Computational studies have successfully applied neural network agents with such biologically-plausible synaptic plasticity rules to a wide range of tasks (Baras and Meir 2007; Brea, Senn, and Pfister 2013; Farries and Fairhall 2007; Fiete and Seung 2006; Frémaux and Gerstner 2016; Hoerzer et al. 2012; Izhikevich 2007; Legenstein et al. 2010; Legenstein, Pecevski, and Maass 2008; Miconi 2017; Pfister et al. 2006; Senn and Pfister 2014; Suri and Schultz 1998, 1999; Urbanczik and Senn 2009; Xie and Seung 2004), including navigation to a single reward location (Arleo and Gerstner 2000; Brzosko et al. 2017; Foster et al.

2000; Frémaux et al. 2013; Potjans, Diesmann, and Morrison 2011; Potjans, Morrison, and Diesmann 2009; Vasilaki et al. 2009; Zannone et al. 2018). Here we extend previous work by describing agents with biologically plausible synaptic plasticity that learn the multiple paired association navigation task.

We build on the classic actor-critic agents developed by Foster and colleagues, and Frémaux and colleagues that learn to navigate to a single reward location (Foster et al. 2000; Frémaux et al. 2013). In the discrete-time agent of Foster and colleagues with rate-based neurons, place cells encoding the animal's location project to an actor that outputs the animal's movement, and to a critic that outputs a (estimated) value function, which is an estimate of the cumulative reward that may be obtained. The value function and reward obtained are used to calculate the temporal difference (TD) error, a reward prediction error encoded with various degrees of fidelity by midbrain dopamine neurons (P. Read Montague, Dayan, and Sejnowski 1996; W Schultz et al. 1997), cholinergic basal forebrain neurons (Hangya et al. 2015), and mouse cerebellar climbing fibers (Ohmae and Medina 2015). Plasticity in place cell to actor synapses obeys a TD error-modulated Hebbian rule, depending on the product of the TD error, presynaptic activity and postsynaptic activity. Plasticity in place cell to critic synapses depends on the product of the TD error and presynaptic activity. The agent of Frémaux and colleagues has the same architecture, but uses spiking neurons, actor neurons connected in a ring, TD error-modulated Hebbian plasticity for place cell to critic synapses, and a continuous-time TD error (Doya 2000).

We find that although a similar classic agent learns to navigate to a single reward location, and adapts to displacement of the reward location after the initial learning (Zannone et al. 2018), it is not able to learn the multiple paired association navigation task. This limitation is overcome by an agent in which place cell and cue information do not go directly to the actor and critic, but are first processed by a nonlinear hidden

layer whose synapses onto the actor and critic are subject to TD error-modulated plasticity.

2.2 Methods

2.2.1 Paired association spatial navigation tasks

In all paired association navigation tasks, an agent moves within a spatially continuous two-dimensional square arena bounded by walls of length 1.6 m, with possible agent positions $x = (\pm 0.8\text{ m}, \pm 0.8\text{ m})$. The agent also receives a sensory cue c that remains constant throughout the trial, or that may be presented only at the start of the trial. At the start of each trial, the agent's internal activity is randomly initialized, with its position drawn with equal probability from the midpoints of the four boundary walls. The agent moves by executing time-dependent actions $a(t)$ that affect its velocity according to

$$\dot{x}(t) = a(t) \tag{1}$$

Using Euler's method of discretization with time step Δt, this results in position updates

$$x(t + \Delta t) = x(t) + a(t).\Delta t \tag{2}$$

If that updated position ends up outside the arena, the agent instead moves 0.01 m inward perpendicular to the closest boundary from its last position. We used a time step of 100 ms for all simulations, but the main results have been checked to also hold at time steps of 20 ms, 15 ms and 5 ms.

Across all trials for an agent, any particular sensory cue is consistently associated with a reward in only one of 49 possible reward locations distributed throughout the maze such that the centres of possible reward locations are 0.2 m from each other or a boundary. All possible reward locations are circles with a radius of 0.03 meters.

The agent is free to explore the arena for a maximum duration T_{max} per trial. If it finds the reward before T_{max}, the agent remains stationary until the trial ends to model consummatory behavior. After the agent reaches the reward, a total reward value $R = 1$ (Fig. 2.1-2.5) or $R = 4$ (Fig. 2.6) is disbursed at a reward rate $r(t)$, defined by

$$\dot{r}_{decay}(t) = -\frac{r_{decay}(t)}{\tau_{decay}}\,; \quad \dot{r}_{rise}(t) = -\frac{r_{rise}(t)}{\tau_{rise}} \tag{3}$$

$$r(t) = \frac{r_{decay}(t) - r_{rise}(t)}{\tau_{decay} - \tau_{rise}} \tag{4}$$

with $\tau_{rise} = 120$ ms and $\tau_{decay} = 250$ ms. When the agent reaches the reward, instantaneous updates

$$r_{rise}(t) \to r_{rise}(t) + R; \quad r_{decay}(t) \to r_{decay}(t) + R \tag{5}$$

are made, such that $r(t)$ integrates to R. To prevent infinitely long trials, trials in which the reward is reached before T_{max} are terminated when 99.99% of the reward has been consumed. Trials in which the reward is not reached before T_{max} are terminated at T_{max}.

2.2.2 Agent: place cells

All agents have 49 place cells whose firing rates depend on the agent's position. The firing rate of the ith place cell is

$$u_i^{pc}(t) = \exp\left(-\frac{(x(t) - x_i)^2}{2{\sigma_{pc}}^2}\right) \tag{6}$$

with $\sigma_{pc} = 0.267$ m, and place cells centres x_i spaced 0.267 m apart at the intersections of a regular 7-by7 grid.

Sensory Cue

Each cue c is encoded by u^{cue}, a one-hot vector of length 18 with gain 3, e.g. $u^{cue} = [3,0,0,0,\dots]$ for the first cue. The cue and u^{cue} were constant throughout each trial, except for the task of Fig. 2.6. In Fig. 2.6, the cue was presented briefly at the start of each trial as in the experiment of Tse and colleagues (Tse et al. 2007); u^{cue} was constant for the first 5 seconds with place cell activity and agent actions silenced to simulate cue presentation to the rat in the starting box with no knowledge of its position in the maze; the cue was then switched off, u^{cue} set to zero, and place cell activity and agent actions switched on for navigation; the cue reappeared and u^{cue} was reactivated for the time step in which the reward was found; however, results are similar without cue reappearance and u^{cue} reactivation.

2.2.3 Agent: actor

All agents have an actor of $M = 40$ neurons. The firing rate of the *k*th actor neuron is

$$\rho_k(t) = R\text{eLU}[q_k(t)] \tag{7}$$

where the rectified linear unit (ReLU) activation function is

$$\text{ReLU}(x) = \begin{cases} 0, & x \le 0 \\ x, & x > 0 \end{cases} \tag{8}$$

and the membrane potential q_k has dynamics

$$\tau_q \dot{q}_k(t) = -q_k(t) + \sum_{j=1}^{N} W_{jk}^{actor} r_j^{agent}(t) + \sum_{h=1}^{M} W_{hk}^{lateral} \rho_h(t) + \sqrt{\tau_q {\sigma_{actor}}^2}\xi(t) \tag{9}$$

with $\tau_q = 150\,ms$, and $\sigma_{actor} = 0.25$. The W_{jk}^{actor} are synaptic weights for the N elements of r_j^{agent}, which is r_j^{cl}, r_j^{clex}, r_j^{hlin}, r_j^{hnlin}, r_j^{res} respectively for the Classic,

Expanded Classic, Linear Hidden Layer, Nonlinear Hidden Layer, and Reservoir agents. The synaptic weights

$$W_{hk}^{lateral} = \frac{w_-}{M} + w_+ \frac{f(k,h)}{\sum_h f(k,h)} \tag{10}$$

with $f(k,h) = (1-\delta_{kh})e^{\varphi \cos(\theta_k - \theta_h)}$, $w_- = -1$, $w_+ = 1$, and $\varphi = 20$, connect the actor neurons into a ring attractor that smooths the agent's spatial trajectory. Membrane potential dynamics of the actor neuron were discretized with the Euler–Maruyama method:

$$\begin{aligned} q_k(t) = {} & (1-\alpha_q) q_k(t-\Delta t) \\ & + \alpha_q \left(\sum_{i=1}^{N} W_{ij}^{actor} r_i(t-\Delta t) + \sum_{h=1}^{M} W_{hk}^{lateral} \rho_h(t-\Delta t) \right. \\ & \left. + \sqrt{\frac{\sigma^2}{\alpha_q}} N(0,1) \right) \end{aligned} \tag{11}$$

where $\alpha_q \equiv \Delta t/\tau_q$ and N(0,1) is the standard normal distribution.

The kth actor neuron represents a spatial direction $\theta_k = 2\pi k/M$, and the action.

$$a(t) = \frac{a_0}{M} \sum_k \rho_k(t) [\sin\theta_k, \cos\theta_k] \tag{12}$$

is the vector sum of directions weighted by each actor neuron's firing rate, with $a_0 = 0.03$ translating to the agent moving at about 0.7 ms^{-1}.

2.2.4 Agent: critic

All agents have a critic neuron whose firing rate is

$$v(t) = \mathrm{ReLU}[\varsigma_k(t)] \tag{13}$$

where the membrane potential ς_k has dynamics

$$\tau_c \dot{\varsigma}_k(t) = -\varsigma_k(t) + \sum_{j=1}^{N} W_{jk}^{critic} r_j^{agent}(t) + \sqrt{\tau_c {\sigma_{critic}}^2}\xi(t) \qquad (14)$$

where $\tau_c = 150\, ms$, $\sigma_{critic} = 0.0005$, and W_{jk}^{critic} are synaptic weights for r_j^{agent}. The membrane potential dynamics of the critic neuron are discretized with the Euler–Maruyama method:

$$\varsigma_k(t) = (1 - \alpha_c)\varsigma_k(t - \Delta t) + \alpha_c \left(\sum_{j=1}^{N} W_{jk}^{critic} r_j(t) + \sqrt{\frac{{\sigma_{critic}}^2}{\alpha_c}} N(0,1) \right) \qquad (15)$$

where $\alpha_c \equiv \Delta t / \tau_c$.

2.2.5 Input to the actor and critic neurons

We studied five agent architectures, which differ according to how the input to the actor and critic neurons r^{agent} is computed; r^{agent} is r_j^{cl}, r_j^{clex}, r_j^{hlin}, r_j^{hnlin}, r_j^{res} respectively for the Classic, Expanded classic, Linear Hidden Layer, Nonlinear Hidden Layer, and Reservoir agents.

The activity of the 49 place cells and the encoded sensory cue of length 18 are concatenated to form an input vector

$$u(t) = \left[u^{pc}\big(x(t)\big), u^{cue}(t) \right] \qquad (16)$$

with a length of 67.

For the Classic agent,

$$r^{cl}(t) = u(t) \qquad (17)$$

is passed to the 40 actor neurons and the critic neuron, with their synaptic weights constituting 2,747 trainable parameters.

For learning single reward locations, the Expanded classic agent is a variant of the Classic agent in which 16 copies of the activity of the 49 place cells and the encoded sensory cue of length 18 are concatenated as

$$r^{clex}(t) = [u, u, u, u, u, u, u, u, u, u, u, u, u, u, u, u\ ] \tag{18}$$

to form a vector of length 1,072 that was passed to the 40 actor neurons and the critic neuron, with their synaptic weights constituting 43,952 trainable parameters. For learning multiple PAs, the Expanded classic agent was made up of 123 concatenated copies of place cell activity and the encoded sensory cue, so that there were 337,881 trainable parameters.

In the Linear Hidden Layer agent and the Nonlinear Hidden Layer agent, place cell activity and the encoded sensory cue are passed to a hidden layer, whose activity is then passed to the actor and critic neurons. The firing rates of the hidden layer neurons in the Linear Hidden Layer agent are

$$r_j^{hlin}(t) = A\left(\sum_{i=1}^{M} W_{ij}^{in} u_i(t)\right) \tag{19}$$

with the linear hidden layer gain $A = 0.2$ to keep firing rates largely between -1 and 1. The firing rates of the hidden layer neurons in the Nonlinear Hidden Layer agent are

$$r_j^{hnlin}(t) = \mathrm{ReLU}\left[\sum_{i=1}^{M} W_{ij}^{in} u_i(t)\right] \tag{20}$$

Hidden layers had 1024 units when learning single reward locations, and or 8192 units when learning multiple PAs. The synaptic weights W_{ij}^{in} from the input vector to the

hidden layer were drawn from a uniform distribution between [-1,1], and not subject to synaptic plasticity. Only the synaptic weights from the hidden layer to the actor and critic units were subject to synaptic plasticity, such that that there were 41,984 and 335,872 trainable parameters respectively for learning single reward locations and for learning multiple PAs.

In Fig. 2.5, in addition to ReLU, other nonlinear functions for the hidden layer neurons are studied, including Leaky ReLU (LReLU) (Maas, Hannun, and Ng 2013), exponential linear unit (ELU) (Clevert, Unterthiner, and Hochreiter 2016), softplus (Glorot, Bordes, and Bengio 2011), hyperbolic tangent (tanh), sigmoid (logistic), and two nonlinear activation functions

$$\phi^A(x,\theta) = \begin{cases} 0, & x \le \theta \\ x, & x > \theta \end{cases} \tag{21a}$$

and

$$\phi^B(x,\theta) = \begin{cases} \theta, & x \le \theta \\ x, & x > \theta \end{cases} \tag{21b}$$

The nonlinear activation functions ϕ^A and ϕ^B are identical to ReLU when $\theta = 0$. In the Reservoir agent of Fig. 2.6, place cell activity and the sensory cue are encoded in u^{wm} (Eq. 30), which is passed to the reservoir of recurrently connected neurons, whose activity is then passed to the actor and critic neurons. The firing rates of the reservoir neurons are

$$r_j^{res}(t) = \phi^A[x_j(t), \theta = 3] \tag{22}$$

and the membrane potential x_j were described by

$$\tau_r \dot{x}_j(t) = -x_j(t) + \sum_{i=1}^{M} W_{ij}^{in} u_i^{wm}(t) + \lambda \sum_{h=1}^{N} W_{hj}^{rec} \tanh[x_h(t)] + \sqrt{\tau_r \sigma_{res}{}^2}\,\xi(t) \quad (23)$$

with $\lambda = 1.5$, $\tau_r = 150\,ms$, and $\sigma_{res} = 0.025$. The synaptic weights W_{ij}^{in} are drawn from a uniform distribution between [-1, 1]; W_{hj}^{rec} are drawn from a Gaussian distribution with mean 0 and variance $1/pN$ with connection probability p = 1. These synaptic weights are not subject to synaptic plasticity. Only the synaptic weights from the reservoir to the actor and critic units were subject to synaptic plasticity. The membrane potential dynamics of the reservoir neurons were discretized with the Euler–Maruyama method:

$$x_j(t) = (1-\alpha_r)x_j(t-\Delta t) + \alpha_r \left(\sum_{i=1}^{M} W_{ij}^{in} u_i^{wm}(t-\Delta t) + \lambda \sum_{j'=1}^{N} W_{hj}^{rec} \tanh[x_h(t)-\Delta t] + \sqrt{\frac{\sigma_{res}}{\alpha_r}} N(0,1) \right) \quad (24)$$

All trainable parameters in all agents were initialized at zero before learning.

2.2.6 Working memory

In Fig. 2.6, the sensory cue is presented only briefly, and working memory is needed to maintain a neural representation of the sensory cue. We implemented the working memory with a bump attractor (Parthasarathy et al. 2019; Wimmer et al. 2014). There are $N_{bump} = 54$ bump attractor neurons. The firing rate of a bump attractor neuron is given by

$$u_j^b(t) = \text{ReLU}\,[x^b{}_j(t)] \tag{25}$$

where the membrane potential x_j^b has dynamics

$$\begin{aligned}\tau_b \dot{x}_j^b(t) = & -x_j^b(t) + \sum_{i=1}^{M_{cue}} W_{ij}^{inwm} u_i^{cue}(t) \\ & + \sum_{h=1}^{N_{bump}} W_{hj}^{bump}\, \omega\left[x_h^b(t)\right] + \sqrt{\tau_b {\sigma_{bump}}^2}\,\xi(t)\end{aligned} \tag{26}$$

with $\tau_b = 150$ ms, $\sigma_{bump} = 0.1$, and nonlinear activation function

$$\omega(x) = \begin{cases} 0, & x < 0 \\ x^2, & 0 < x \leq 0.5 \\ \sqrt{2x - 0.5}, & x > 0.5 \end{cases} \tag{27}$$

The bump attractor neuron membrane potential dynamics were discretized with the Euler–Maruyama method:

$$\begin{aligned}x_j^b(t) = & (1 - \alpha_b) x_j^b(t - \Delta t) \\ & + \alpha_b \left(\sum_{i=1}^{M} W_{ij}^{inwm} {u^{cue}}_i(t - \Delta t) \right. \\ & \left. + \sum_{h=1}^{N_{bump}} W_{hj}^{bump}\, \omega\left[x_h^b(t - \Delta t)\right] + \sqrt{\frac{{\sigma_{bump}}^2}{\alpha_b}} N(0,1) \right)\end{aligned} \tag{28}$$

The synaptic weights

$$W_{hj}^{bump} = \frac{w_-}{N_{bump}} + \frac{f(k,h)}{\sum_h f(k,h)} \tag{29}$$

with $f(j,h) = e^{\varphi \cos(\theta_j - \theta_\varepsilon)}$, $w_- = -0.75$, and $\varphi = 300$, connected the neurons in a ring. Since each of the 18 cues was encoded as a one-hot vector, the W_{ij}^{inwm} are

specified such that each cue activated three adjacent units in the ring, and the total strength of the weights for each cue passed to the bump attractor was 1.

For all agents in Fig. 2.6, the bump attractor activity is concatenated with the place cell activity and encoded sensory cue to form the input vector

$$u^{wm}(t) = \left[u^{pc}\left(x(t)\right), u^{cue}(t), u^{b}\ (t)\right] \tag{30}$$

2.2.7 Continuous temporal difference error and synaptic plasticity

The output of the critic $v(t)$ and the reward $r(t)$ are used to define the continuous TD error (Doya 2000; Frémaux et al. 2013; Jordan, Weidel, and Morrison 2019)

$$\delta(t) = r(t) + \dot{v}(t) - \frac{1}{\tau_g} v(t) \tag{31}$$

As noted by Doya (Doya 2000), discretization by substituting $\dot{v}(t) \approx (v(t) - v(t - \Delta t)\ )/\Delta t$ together with approximating reward and critic output by their values at the end of the time interval used for approximating $\dot{v}(t)$, i.e. $r(t) \approx r(t)$ and $v(t) \approx v(t)$, gives

$$\delta(t) = r(t) + \frac{1}{\Delta t}[(1 - \alpha)v(t) - v(t - \Delta t)] \tag{32}$$

where $\alpha = \Delta t/\tau_g$, which has the same form as the discrete time TD error

$$\delta_d(t) = r(t) + \gamma \cdot v_d(t) - v_d(t - 1) \tag{33}$$

if we take $\gamma = 1 - \frac{\Delta t}{\tau_g}$ and $v_d = v/\Delta t$. Alternatively, discretization by substituting $\dot{v}(t) \approx (v(t + \Delta t) - v(t)\ )/\Delta t$ together with approximating reward and critic output by their values at the start of the time interval used for approximating $\dot{v}(t)$, i.e. $r(t) \approx r(t - \Delta t)$ and $v(t) \approx v(t - \Delta t)$ gives

$$\delta(t) = r(t - \Delta t) + \frac{1}{\Delta t}[v(t) - (1 + \alpha)v(t - \Delta t)] \quad (34)$$

We used $\tau_g = 2000$ ms (equivalent to $\gamma = 0.95$). Figures and analyses in this chapter are from simulations in which the continuous TD error was implemented using Eq. 34. Simulations using Eq. 32 gave similar results with time steps of 50 ms, 20 ms and 5 ms.

Synaptic plasticity of the weights onto the critic are governed by a 2-factor rule, being modulated by the continuous TD error and the presynaptic firing rate (Foster et al. 2000; Sutton and Barto 2020):

$$\dot{W}^{critic}(t) = \eta_{critic} \cdot r_j(t) \cdot \delta(t) \quad (35)$$

which we discretized using Euler's method:

$$W^{critic}(t) = W^{critic}(t - \Delta t) + \Delta t \cdot \eta_{critic} \cdot r_j(t) \cdot \delta(t) \quad (36)$$

Synaptic plasticity of the weights onto the actor are governed by a 3-factor rule, being modulated by the continuous TD error, the presynaptic firing rate, and the postsynaptic firing rate (Foster et al. 2000; Frémaux et al. 2013; Sutton and Barto 2020):

$$\dot{W}^{actor} = \eta_{actor} \cdot r_j(t) \cdot \rho_k(t) \cdot \delta(t) \quad (37)$$

which we discretized using Euler's method:

$$W^{actor}(t) = W^{actor}(t - \Delta t) + \Delta t \cdot \eta_{actor} \cdot r_j(t) \cdot \rho_k(t) \cdot \delta(t) \quad (38)$$

The learning rates used for η_{critic} and η_{actor} were chosen using grid search to optimize speed and consistency of learning for a single reward location. When the same learning rates were used for the multiple PA task, the agent got stuck in corners. Hence, the learning rates for the multiple PA task were gradually reduced from those used in the

single reward location task until successful learning was achieved. For the single reward location task, learning rates were 0.015 for the classic agent, 0.0005 for the Expanded classic and Linear Hidden Layer agents, and 0.0001 for the Nonlinear Hidden Layer and Reservoir agents. For the multiple paired association navigation task, learning rates were 0.001 for the Classic agent, and 0.00001 for the Expanded Classic, Linear Hidden Layer, Nonlinear Hidden Layer, and Reservoir agents.

2.2.8 Generation of value and policy maps

Each agent's trajectory was binned into a 15 × 15 square grid that covered the maze's dimensions. Spatial value maps were generated from the critic's firing rate. The mean critic firing rate at each bin was computed for the duration of the cue-probe trial over all iterations and visualized as a heatmap. Spatial policy maps were generated from the actor's action $a(t)$. The vector sum of the action at each bin was computed for the duration of the cue-probe trial over all iterations and visualized as a quiver plot.

2.2.9 Hidden layer activity dimensionality

A random sequence of 500 input vectors, each drawn independently from the input vectors corresponding to all possible combinations of one location from a grid of 2500 possible locations within the arena and one of the 18 possible sensory cues, were provided as inputs to a hidden layer with a variable number of neurons and activation functions. Principal components analysis (PCA) was performed on the corresponding hidden layer output sequence. Dimensionality was estimated as the number of principal components needed to explain 95% of the variance.

2.2.10 Data availability

Code for our results is available at https://github.com/mgkumar138/TDHL_6PA. As stated in the Introduction, deep RL algorithms can learn the multiple paired association

navigation task. As their performance on this specific task has not been reported, we have also included code for training A2C, a deep RL algorithm, on the task.

2.3 Results

We first verify the ability of four actor-critic agents, the Classic, Expanded Classic, Linear Hidden Layer, and Nonlinear Hidden Layer agents, to learn the single reward location task, as well as a variant task requiring adaptation to displacement of the single reward location after the initial learning. We then study their performance on a version of the multiple paired association task in which the sensory cue indicating the reward location is present throughout each trial, and find that only the Nonlinear Hidden Layer agent is able to learn the task. We visualize the different policy and value maps learned by the Classic and Nonlinear Hidden Layer agents, and characterize the effect of agent hyperparameters on learning. Finally, we demonstrate that a bump attractor to provide working memory can be integrated with the Nonlinear Hidden Layer agent or to a reservoir agent, a variant of the Nonlinear Hidden Layer agent, to learn a version of the multiple paired association task that resembles the biological experiments more closely, with the sensory cue presented only at the start of each trial.

2.3.1 Learning to navigate to a single reward location

We begin by verifying that the agents can learn to navigate to a single reward location, located in the north-west corner of the maze (Fig. 2.1A). In each trial, the agent starts at a randomly chosen midpoint of the north, south, east, or west boundaries of a 1.6 m^2 square arena. The agent then receives the same sensory cue on every timestep until it reaches the reward location. The next trial begins with the starting point selected randomly from one of the midpoints.

All the agents have rate-based neurons. Agents receive input from place cells that encode the animal's location and encoded information about the presence and identity of the sensory cue. They have an actor made up of neurons connected in a ring whose

output dictates the speed and direction of agents (see Methods). They also have a critic whose output is the value function. The value function and reward obtained by the agent are used to calculate the temporal difference (TD) error. Only synapses connected to the actor and the critic are plastic. Plasticity at synapses connected to the actor obeys a TD error-modulated Hebbian rule, depending on the product of the TD error, presynaptic activity and postsynaptic activity. Plasticity at synapses connected to the critic depends on the product of the TD error and the presynaptic activity.

In the Classic agent (Fig. 2.1A), place cells and cue cells encoding sensory information project directly onto the actor and critic neurons. In the Expanded Classic agent (Fig. 2.1A), place cells and cue cells also project directly onto the actor and critic, but there are multiple copies of each connection onto the actor and critic, each with its own plastic synapse; this creates a variant of the Classic agent without a hidden layer, but with the same number of trainable parameters as the agents with a hidden layer. In the Linear Hidden Layer agent (Fig. 2.1B) and the Nonlinear Hidden Layer agent (Fig. 2.1B), place cell and cue information do not go directly to the actor and critic, but are first processed by a hidden layer whose neurons synapse onto the actor and critic. Please see the Methods section for details.

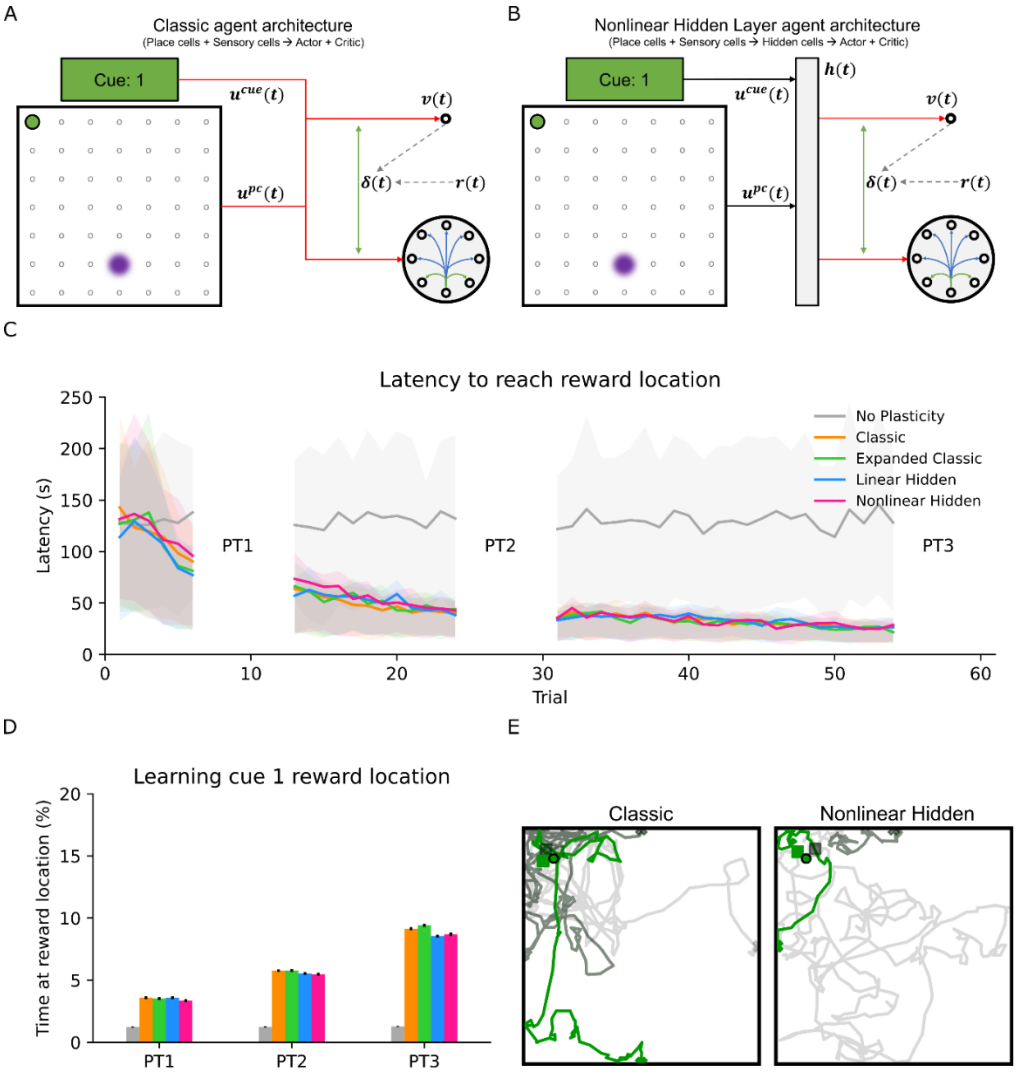


Figure 2.1. Classic and Nonlinear Hidden Layer agents learn single reward locations equally well. (A) Schematic of arena and classic agent. The single reward location (green) is in the north-west corner of the maze and the activity of the place cells (centred on grey circles) represent agent position (purple). Place cell activity, $u^{pc}(t)$, and encoded cue information, $u^{cue}(t)$, are passed directly to the actor (whose global inhibition and local excitation connection structure are shown in the blue and green lines, respectively) and critic, whose respective outputs are agent velocity, $\rho(t)$, and an (estimated) value function, *v(t)* (see Methods). The value function and reward, $r(t)$, obtained by the agent are used to calculate the TD error, $\delta(t)$, which modulates synaptic plasticity (shown in the green arrows). Only the red connections are plastic. **(B)** The Nonlinear Hidden Layer agent has an architecture similar to that of the classic agent, except that place cell and cue information do not go directly to the actor and critic, but are first processed by a hidden layer whose neurons synapse onto the actor and critic. **(C)** Mean latency to reach the reward location versus trial number (200 simulations per agent type, shaded area indicates 25th and 75th quantiles) for different types of agents (see legend in D). Three sets of six probe trials (labelled as PT1, PT2, and PT3) were used to assess learning progress. **(D)** Mean time spent near the reward location in non-rewarded probe trials (200 simulations per agent type, error bars are

standard errors). **(E)** Trajectories (truncated when the reward location is reached) of a classic agent (left) and a Nonlinear Hidden Layer agent (right) on the first trials of PT1 (light grey), PT2 (dark grey) and PT3 (green). Crosses and squares indicate an agent's start and end location respectively.

All agents, except the control Classic agent without plasticity, learned to navigate to the single reward location comparably well. This was demonstrated by the decrease in latency in reaching the single reward location over 42 trials (Fig. 2.1C). Their learning was also seen with the probe trials that occurred on trials 7–12 (PT1), 25–30 (PT2), and 55–60 (PT3). In a probe trial, no reward was given even if the agent reached the correct reward location. Agent plasticity was switched off, and the trial ends after 60 seconds, allowing one to determine whether the amount of time an agent spent near the reward location increased with learning. All agents, except the control agent, spent similarly increasing amounts of time near the reward location across probe trials (Fig. 2.1D). Example Classic and Nonlinear Hidden Layer agents both showed more direct movement to the reward location in later probe trials (Fig. 2.1E, green trajectory). They also showed value maps with higher values that were more concentrated near the reward location, and policy maps that were more directed toward the reward location in later probe trials (Supplementary Fig. 2.1A).

2.3.2 Learning to navigate to a displaced reward location

In a variant of the single reward location task, the reward location is displaced on subsequent trials. The variant task can be learned by biologically-plausible reinforcement learning agents (Zannone et al. 2018), including a classic actor-critic agent (Foster et al, 2000). Previous work did not characterize how performance varies with degree of displacement, which we here describe for the various actor-critic agents. After the 42 trials in which an agent has learned the first reward location, it continues with 42 more trials with either the original cue-reward location pair (Reward Location

1), or a new cue-reward location pair in which the reward location is displaced by an integer multiple of 0.28 m along the diagonal (Reward Locations 2–7; Fig. 2.2A).

We again gauged learning by the amount of time each agent spent near the reward location during probe trials, which occurred on trials 7–12 (PT4), 25–30 (PT5), and 55–60 (PT6) after displacement of the reward location; PT4, PT5, and PT6 may thus be compared with PT1, PT2, and PT3, respectively. For the case in which the reward location was not displaced, all agents with plasticity continued to increase the amount of time spent near the reward location from PT3 (Fig. 2.1D) to PT6. For the displaced reward locations, all agents spent increasing amounts of time near the reward location from PT4 to PT6 (Fig. 2.2B–D). For all agents, the closer the displaced reward location was to the original reward location, the sooner the agent reached a given level of performance measured by time spent near the reward location. Compared with their performance on the original reward location at PT3, all agents reached comparable or better performance at PT6 for reward locations 1–4, and worse performance for reward locations 5–7. The higher performance with more trials for Reward Locations 1–4 is reflected in example Classic and Nonlinear Hidden Layer agents having value maps with higher values more concentrated near the displaced reward location, and policy maps that were more directed toward the displaced reward location in later probe trials (Fig. 2.2E, Supplementary Fig. 2.1B–D). The higher performance for smaller displacements of the reward location can also be seen in the trajectories of both example Classic and Nonlinear Hidden Layer agents (Fig. 2.2F).

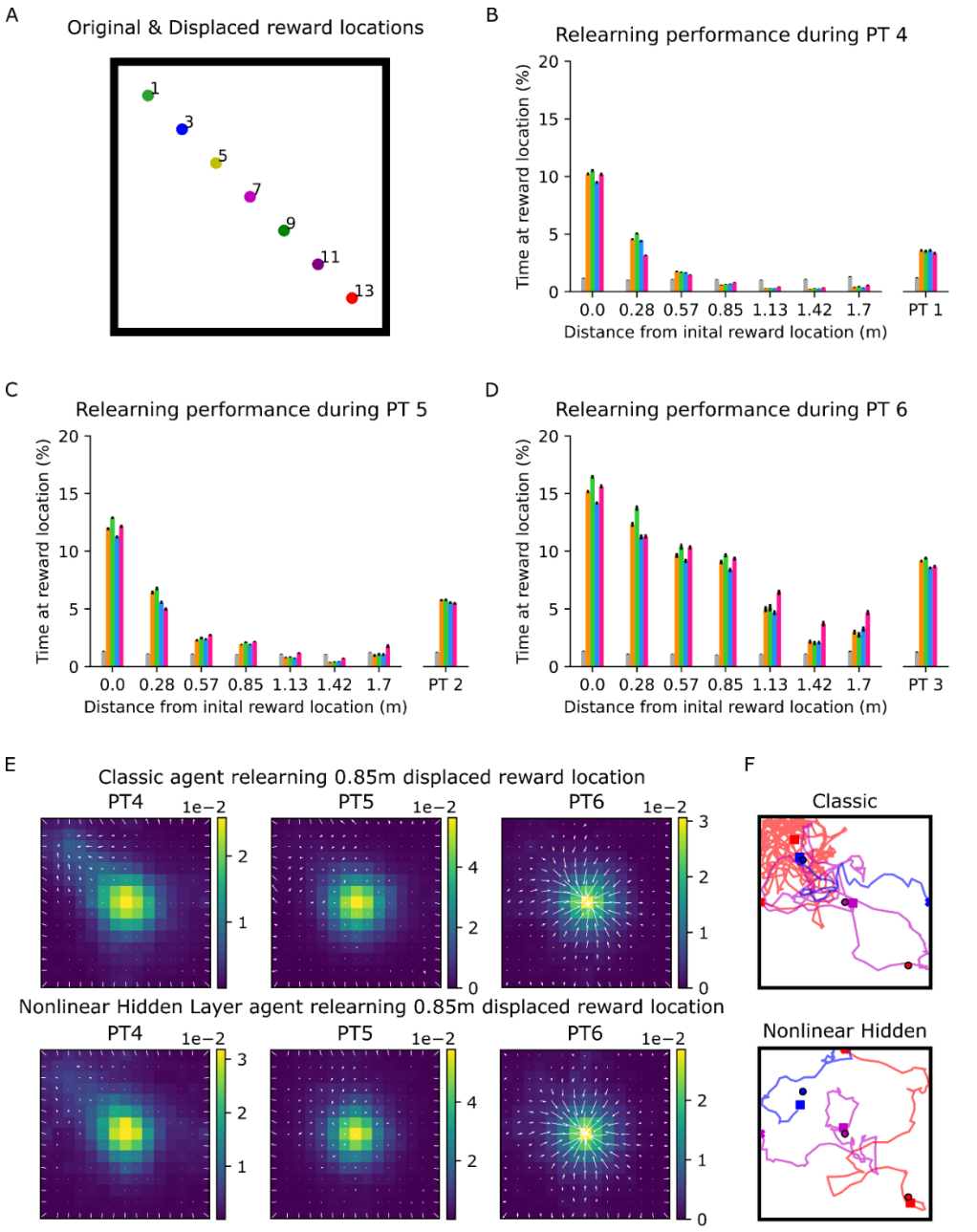


Figure 2.2. Learning to navigate to a displaced reward location depends on the degree of displacement. (A) Original and displaced reward locations numbered 1–7. **(B)–(D)** Mean time spent near displaced reward locations during non-rewarded probe trials PT4, PT5, and PT6. PT1, PT2, and PT3 performance from Fig. 2.1D included as an inset to compare relearning performance against PT4, PT5, and PT6. **(E)** Superimposed value (colour) and policy (small white arrows) maps of example Classic (top) and Nonlinear Hidden Layer (bottom) agents on the first trials of PT4, PT5, and PT6. **(F)** Trajectories (truncated when the reward location is reached) of a Classic agent (top) and a Nonlinear Hidden Layer agent (bottom) on the first trials of PT6 for displaced Reward Locations 2 (blue), 4 (purple), and 7 (red). Crosses and squares indicate an agent's start and end location respectively.

Earlier work with biologically-plausible reinforcement learning agents that did not have an actor-critic structure showed that positive and negative modulation of synaptic plasticity, compared against purely positive modulation, accelerated adaptation to displaced reward locations (Zannone et al. 2018). In the actor-critic agents we studied, the TD error similarly provided positive and negative modulation of synaptic plasticity that aided adaptation to displaced reward locations. This is seen in the example TD error maps for probe sessions, which have a negative trace value at the original reward location, where the agent had learned to expect reward but did not receive it (Supplementary Fig. 2.1A–D).

2.3.3 Learning multiple paired association navigation

Having shown that both Classic and the Nonlinear Hidden Layer agents learned the single reward location task, we subsequently compared their ability to learn a multiple paired association navigation task using six cue-reward location pairs. During each trial, one of the cues was presented throughout, and the agent received a reward only if it reached the correct reward location (Fig. 2.3A, bottom). Training was organized into sessions, each consisting of six trials across which the agent was exposed to six cues in random order (Fig. 2.3A, top).

Learning rates were reduced for this task (see Methods), as the learning rates used in the single reward location task did not allow all paired associations to be learned. The Linear and Nonlinear Hidden Layer agents had 8192 hidden units, while the Expanded Classic agent had a comparable number of plastic synaptic weights that were redundantly connected between the input neurons and the actor and critic neurons (see Methods); the Classic and the control agents had the same number of plastic synaptic weights as in the single reward location task.

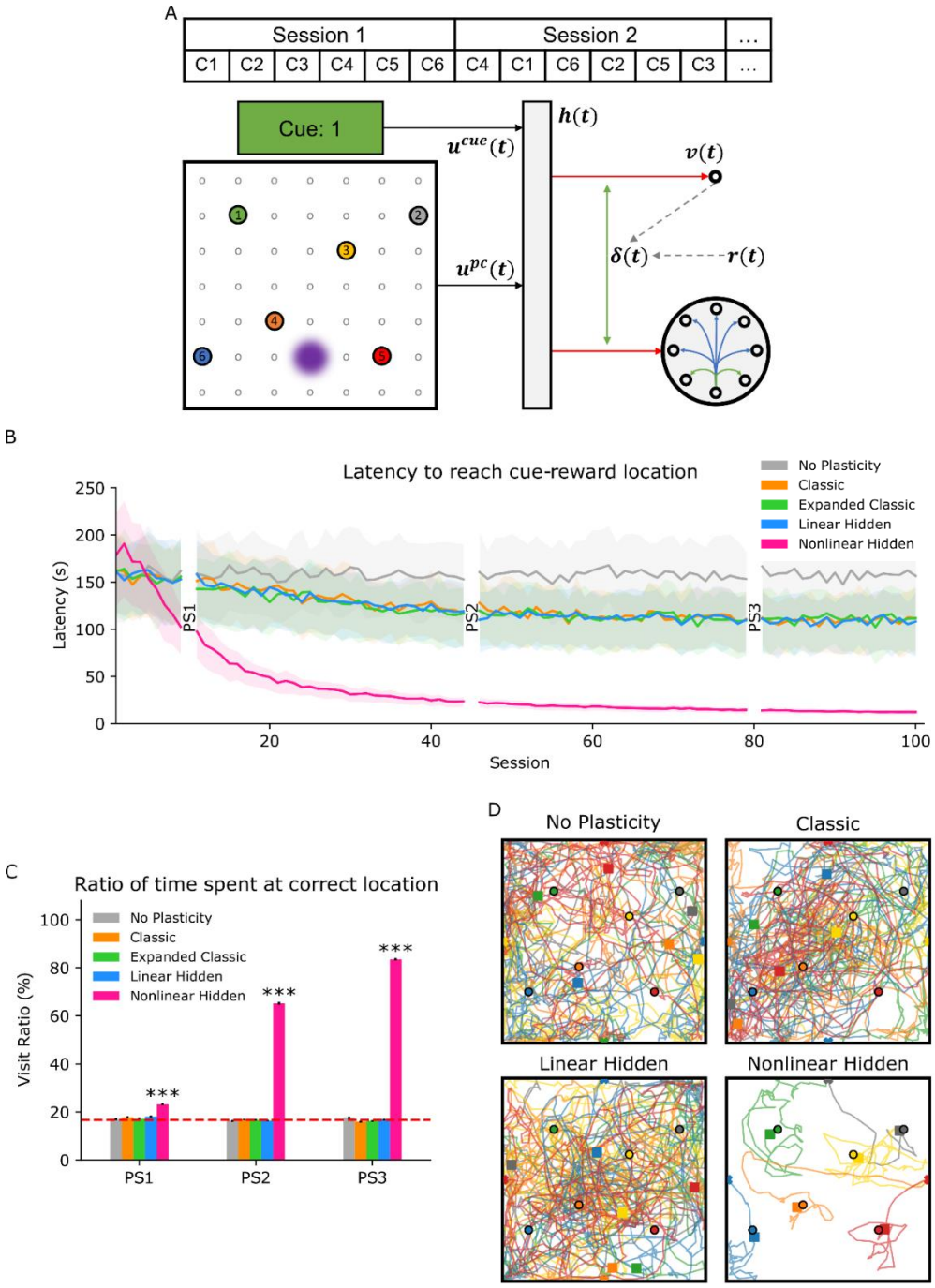


Figure 2.3. Only Nonlinear Hidden Layer agents learned a multiple paired association navigation task. (A) Bottom: Schematic of multiple paired association navigation task with six cue-reward location pairs, and a hidden layer agent. Top: In each session all six cues were presented in random order, with a different cue in each trial. **(B)** Mean latency across all trials in a session to reach the correct reward location versus session number (200 simulations per agent, shaded area indicates 25th and 75th quantiles). Ratio of time spent near the correct cue-reward location compared to the other 5 reward locations during non-rewarded probe session PS1, PS2, and PS3. **(C)** Mean visit ratios in probe sessions with 1 probe trial per cue-reward pair. Student's t test performed against chance performance of 16.7% showed that only the Nonlinear Hidden Layer agent showed above chance performance ($p < 0.001$). **(D)** Example

agent trajectories in PS3 (truncated when the reward location was reached) where each trace colour corresponds to the cued reward location the agent has to navigate to e.g. green trace corresponds to cue 1 reward location in the top left of the maze. Crosses and squares indicate an agent's start and end location respectively.

Figure 2.3B shows the latency required to reach the reward across sessions, averaged across all trials in each session. The latency of all plastic agents decreased below that of the control. The Classic, Expanded Classic and Linear Hidden Layer agents' latencies plateaued at 110 seconds, while the Nonlinear Hidden Layer agent's latency decreased to 13 seconds.

Figure 2.3C shows the visit ratio on non-rewarded Probe Sessions (PS) 10, 45 and 80. An agent's visit ratio was the time it spent within 0.1 m of the centre of the correct reward location, divided by the time it spent within 0.1 m of any of the six possible reward locations. A visit ratio of 16.7% was consistent with chance performance, where the agent visited all reward locations equally, but might also be due to the agent visiting a particular reward location regardless of cue. Although the Classic, Expanded Classic and Linear Hidden Layer agents exhibited modest decreases in latencies, their visit ratios were consistent with chance performance. In contrast, the Nonlinear Hidden Layer agent showed above chance ($p < 0.0001$) visit ratios in all probe sessions, and improved from PS1 to PS2 to PS3.

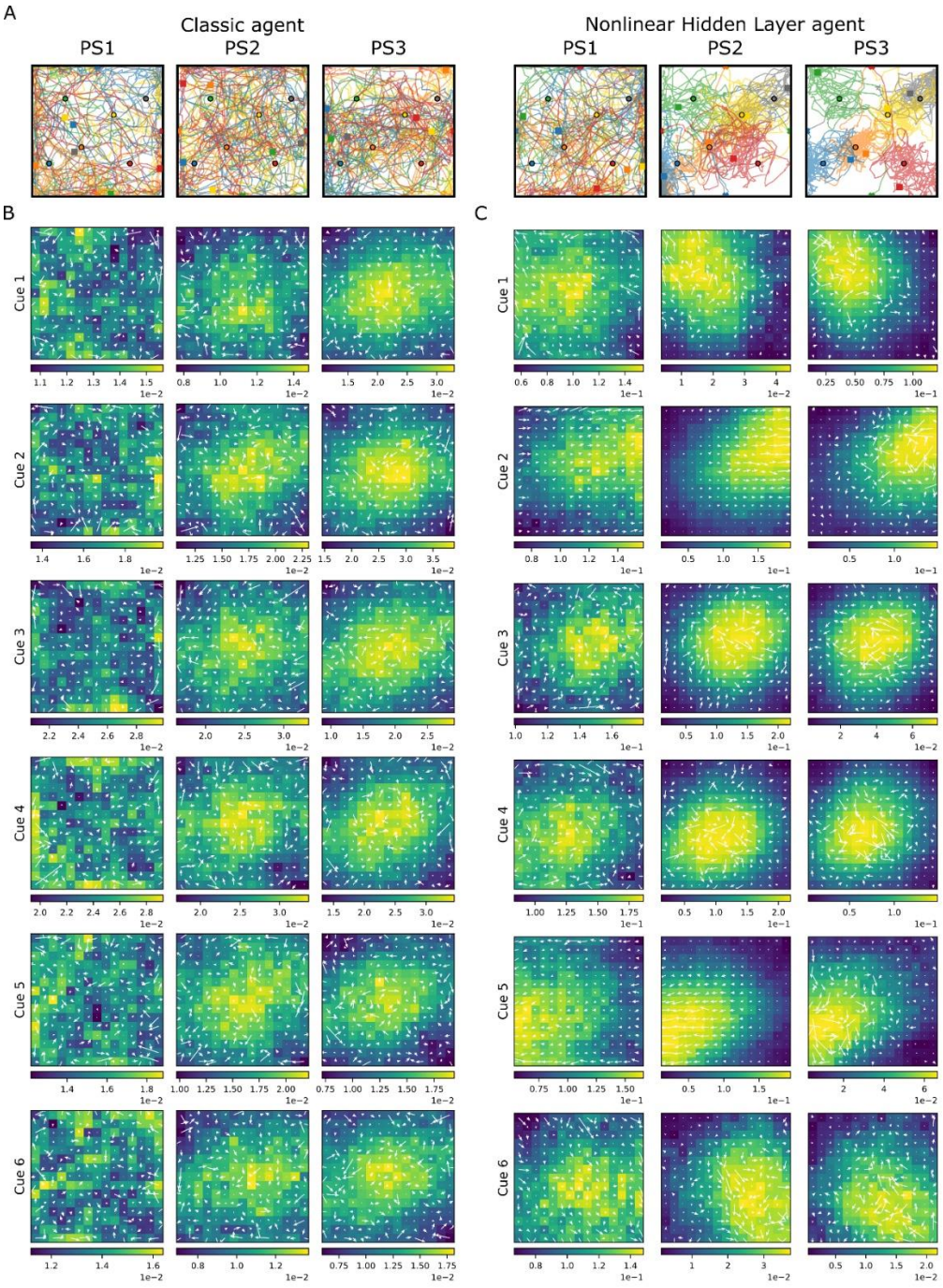


Figure 2.4. Nonlinear Hidden Layer agent learns distinct value and policy maps for each PA. (A) Full trajectories of example Classic and Nonlinear Hidden Layer agents for each of the six different cues during non-rewarded probe sessions. Crosses and squares indicate an agent's start and end location respectively. **(B)–(C)** Superimposed value (colour) and policy (small white arrows) maps for example Classic **(B)** and Nonlinear Hidden Layer **(C)** agents during non-rewarded probe sessions (averaged over 200 simulations per agent).

Trajectories of example Classic and Linear Hidden Layer agents suggested that they learned to avoid the arena boundaries where there were no rewards, and spent more time near the centre of the maze in the vicinity of all reward locations, which may explain how latency can decrease without preferential visits to the correct reward location (Fig. 2.3D, 4A). In contrast, an example Nonlinear Hidden Layer agent moved more directly to the correct reward location for each cue (Fig. 2.3D, 4A). Similarly, an example Classic agent learned value maps with a broad peak of high values near the arena centre encompassing many cues and policy maps directed away from the arena boundary; similar maps were learned for different cues (Fig. 2.4B). In notable contrast, an example Nonlinear Hidden Layer agent learned different maps for different cues, with value maps more concentrated near, and policy maps more exclusively directed towards, the correct reward location for each cue (Fig. 2.4C).

The Classic, Expanded Classic, and Linear Hidden Layer agents did not exhibit above chance performance on the multiple paired association task, despite varying learning rates (Classic: from 0.01 to 0.0001; Expanded Classic and Linear Hidden Layer: from 0.0001 to 0.000001) and Linear Hidden Layer gain (from 0.1 to 1), or increasing the number of training sessions to 500. While we were not able to exclude the possibility that the Classic, Expanded Classic or Linear Hidden Layer agents might succeed in other parameter regimes, these results suggested that adding a Nonlinear Hidden Layer enabled actor-critic agents to learn a multiple paired association task more robustly.

2.3.4 Critical hyperparameters for learning multiple paired associates

A nonlinear hidden layer may facilitate some algorithms by generating a higher dimensional representation of its input (Buonomano and Maass 2009; Litwin-Kumar et al. 2017; Cayco-Gajic and Silver 2019). We therefore examined the effect of hyperparameters that affects hidden layer output dimensionality: the number of hidden units, hidden unit activation function, and distribution of excitatory and inhibitory

synaptic weights onto hidden units. In this section, we used a version of the multiple paired association navigation task with 16 cue-reward pairs (Fig. 2.5A). We increased the number of cue-reward pairs from 6 (in previous sections) to 16 (in this section) to investigate whether particular hyperparameter choices would enable an agent to learn more associations. Agents were trained for 100 sessions, with sixteen trials per session, and a non-rewarded probe session conducted at Session 101. Example trajectories and the value and policy maps after learning 16 associations using $\emptyset^A = 3$ and 8192 hidden units are shown in Figure 2.5B–C. We arbitrarily defined a cue-reward pair to have been learned if an agent achieved a visit ratio of more than 40% for the pair, well above the 6.25% expected if all reward locations were visited randomly. We estimated dimensionality as the number of principal components that explained 95% of the hidden layer output variance given random place and cue inputs.

Figure 2.5D (green line plot) shows the effect of different numbers of hidden units with ReLU activation functions. The expansion ratio was the ratio of the number of hidden layer units to the number of inputs; in this plot the number of inputs was fixed at 67. Increasing the expansion ratio from 1 to 175 increased the average number of cue-reward associations learned from 0.02 ± 0.14 (SD) to 14.9 ± 0.9 (SD) (the results in Fig. 2.3 used 8192 hidden units, which corresponded to an expansion ratio of 122); as the expansion ratio further increased to 250, the average number of associations learned declined modestly ($t = -4.2$, $p < 0.001$) to 13.6 ± 1.9. However, increasing the duration and number of training sessions allowed comparable performance with expansion ratios of 175 and 250 (data not shown). These results indicate that between 1675 (expansion ratio of 25) and 3350 (expansion ratio of 50) hidden units were sufficient to learn six paired associations.

Figure 2.5E (green points) shows the effect of different activation functions in a hidden layer with 8192 units. The ReLU nonlinearity allowed 14.9 ± 0.9 (SD) associations to be learned on average. Its variants, Leaky ReLU (LReLU), exponential LU (ELU) and

Softplus learned 13.3 ± 1.5, 13.7 ± 1.3, 10.9 ± 2.0 (SD) associations respectively on average. The hyperbolic tangent (tanh) nonlinearity enabled 9.7 ± 1.6 (SD) associations to be learned on average. All of these were sufficient to learn the six paired associations used above in Figure 2.3, though the number of associations learned were significantly less than when ReLU was used (independent 2 sample t-test, $p < 0.001$). In contrast, the sigmoid (logistic) nonlinearity learned 0.0 ± 0.0 (SD) associations, performing worse than the linear activation function ($t = -9.0$, $p < 0.001$) with unit gain, which learned 1.4 ± 1.0 (SD) association on average. These results showed that ReLU (and its variants) and the tanh nonlinearity were more suitable in learning multiple paired associations.

We further examined the effect of different activation functions by defining two variants of ReLU. The activation function ϕ^A returned 0 if the input was below the threshold *A*, and was linear with unit gain if the input was above the threshold (inset in Fig. 2.5F); if the threshold was 0, the input-output curve was identical to ReLU (black); if the threshold was negative, the output would be 0 initially before turning negative and then positive (purple); if the threshold was positive, the output would be 0 before turning positive (blue). The activation function ϕ^B returned the threshold value *B* if the input was below the threshold, and was linear with unit gain if the input was above the threshold; if the threshold was 0, the input-output curve was identical to ReLU (black); the input-output curve was non-decreasing for all other threshold values (negative threshold – purple, positive threshold – blue). For both ϕ^A and ϕ^B, the number of associations learned changed nonmonotonically with the threshold (Fig. 2.5F–G, green). The best performance across the hyperparameter regimes we studied was obtained with ϕ^A and a threshold of 3, which allowed 16 ± 0 (SD) associations to be learned (Fig. 2.5F, green); higher than the canonical ReLU activation function with threshold of 0 ($t = 7.5$, $p < 0.001$) which learnt 14.8 ± 1.0 (SD) and ϕ^A with a threshold of 2 ($t = 2.1$, $p = 0.04$) which learnt 16.0 ± 0.2 (SD) .

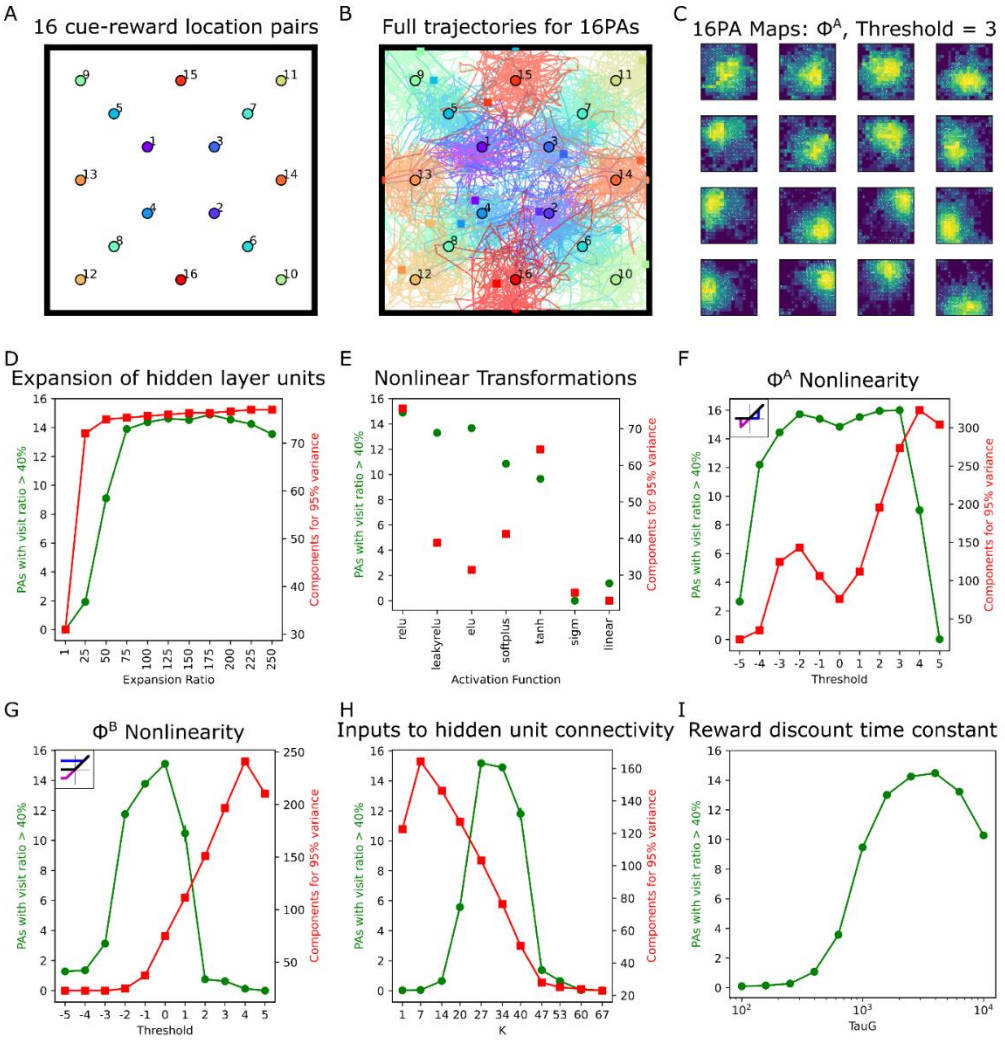


Figure 2.5. Hyperparameters affecting the Nonlinear Hidden Layer agent's ability to learn 16 cue-reward pairs. (A) Schematic of 16 cue-reward pairs. **(B)** Example agent (ϕ^A threshold = 3) trajectories corresponding to each of the 16 cues during the probe session. Crosses and squares indicate an agent's start and end location respectively. **(C)** Example agent (ϕ^A threshold = 3) superimposed value (color) and policy (small white arrows) maps (averaged over 5 simulations per agent) during the probe session. **(D)** Number of associations learned (green) and hidden layer output dimensionality (red) versus expansion ratio. **(E)** Number of associations learned (green) and hidden layer output dimensionality (red) for various activation functions. **(F)–(G)** Number of associations learned (green) and hidden layer output dimensionality (red) versus ϕ^A threshold **(F)** and ϕ^B threshold **(G)**. Inset shows hidden unit's firing rate when threshold is set at -2 (red), 0 (black), 2 (blue). See methods (Eq. 21) for each activation function's formulation. **(H)** Number of associations learned (green) and hidden layer output dimensionality (red) versus K, the number of excitatory inputs **(I)** Number of associations learned versus TD time constant. 40 simulations per hyperparameter condition with error bars indicating standard error.

Figure 2.5H (green) shows the effects of the distribution of excitatory and inhibitory synaptic weights from the 67 inputs to each of the 8192 hidden units with ReLU activation functions. The connectivity hyperparameter K indicated the total number of excitatory inputs each hidden unit received out of the 67 inputs, with the K excitatory input weights drawn from a uniform distribution between 0 and 1, and the remaining inhibitory input weights drawn from a uniform distribution between -1 and 0. The number of associations learned was nonmonotonic in K, with the best and comparable ($t = 1.3$, $p = 0.18$) performances of 15.2 ± 0.8 (SD) and 14.9 ± 1.0 (SD) achieved when $K = 27$ and 34 respectively; almost equal numbers of excitatory and inhibitory inputs ($K = 33.5$) onto each hidden unit.

Figures 2.5D–H also show estimated hidden layer output dimensionality (red) in addition to the number of associations learned (green) for the various hyperparameters. Across hyperparameter regimes, learning 14 or more associations corresponded to a dimensionality of approximately 70 or greater, which was not inconsistent with a minimum dimensionality being needed for a certain learning capacity. However, dimensionality was not sufficient for determining learning, as there were many hyperparameter regimes in which dimensionality and learning capacity trended in opposite directions.

The temporal difference error time constant (τ_g) in Doya's formulation can be considered a function of the discount factor (often denoted γ), or the trace decay factor (denoted λ) in TD(λ) (Doya 2000), both of which are tunable hyperparameters (Bertsekas and Tsitsiklis 1996; Van Seijen et al. 2016; Sutton and Barto 2020; Xu, Van Hasselt, and Silver 2018). We therefore examined the effect of varying the TD time constant in the continuous temporal difference error formulation (Eq. 31) and how this affected the learning of paired associations. Fig. 2.5I shows the effect of varying the TD error time constant in agents with 8192 hidden layer units with ReLU activation functions. As the time constant increased from 100 ms to 3981 ms (equivalent to $\gamma = 0$

to 0.975), the number of associations learned increased monotonically from 0.1 ± 0.3 (SD) to 14.5 ± 1.2 (SD). However, with a further increase in time constant to 10,000 ms ($\gamma = 0.99$), the agent learned only 10.3 ± 1.7 (SD) associations. However, comparable performance was attained with time constant of 10,000 ms and 4000 ms when the learning rate was reduced from 0.00001 to 0.0000075.

2.3.5 Integrating working memory for learning multiple PAs

To focus on the key features of the task and the agent architecture needed, previous sections used a version of the task in which the cue was present throughout each trial, and the hidden layers in the agents were feedforward layers. Here, we address some features of the biological experiments that were omitted in previous sections. First, in the biological experiments, the cue was present only at the start of each trial; second, some brain regions that may be involved are more commonly modelled as recurrent networks than feedforward networks; third, animals learnt the task faster than the agents in the previous sections. In this final section of the results, we therefore considered a slightly different version of the task with 6 cue-reward location pairs in which the cue was present only at the start of each trial (Tse et al. 2007) (Fig. 2.6A). We show that adding a bump attractor to the agent provided working memory that enabled the task to be learned when the cue was present only at the start of each trial. We also showed that using a reservoir of recurrently connected neurons for the hidden layer improved learning, and allowed agents to learn as quickly as animals.

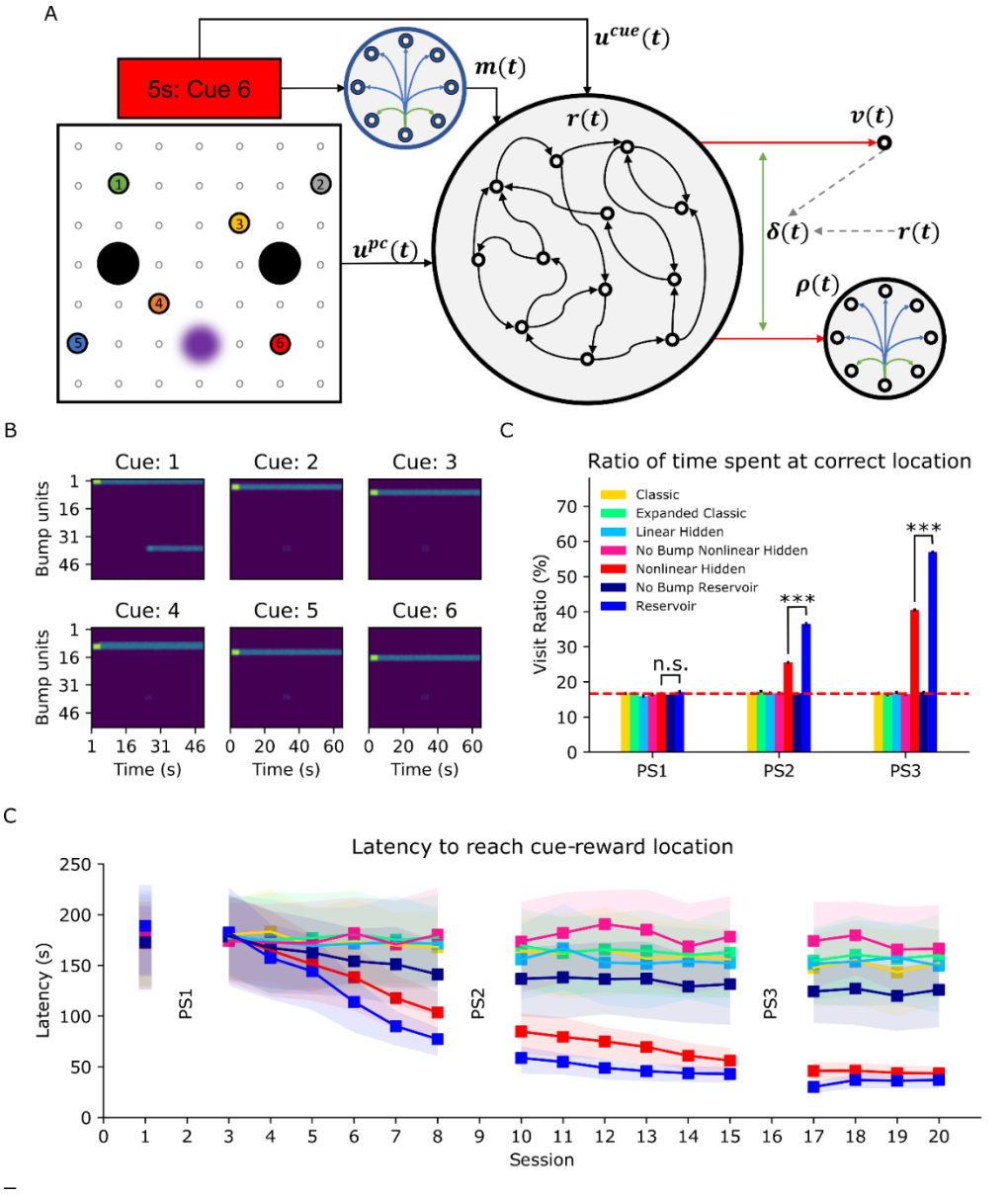


Figure 2.6. Learning multiple paired association navigation with transient cues. **(A)** Schematic of agent with bump attractor and recurrent reservoir; the bump attractor received the encoded cue as input and excited the reservoir; the reservoir provided inputs to the actor and critic. **(B)** Persistent bump attractor activity after presenting cues 1 to 6 for the first 5 seconds; despite a distractor at 30 seconds, the main bump persisted, and the distractor was suppressed 82.8% of the time over 30 simulation runs. **(C)** Mean visit ratios in probe sessions (200 simulations per agent). **(D)** Mean latency across all trials in a session to reach the correct reward location versus session number (200 simulations per agent, shaded area indicates 25th and 75th quantiles).

In agents with working memory, the encoded cue excited not only the hidden layer, but also excited a bump attractor (Fig. 2.6A). Each cue caused persistent activity throughout the trial in a different subset of bump attractor neurons (Fig. 2.6B). The activity of the bump attractor was an additional input to the hidden layer, which was either a nonlinear feedforward layer or a recurrent reservoir (Fig. 2.6A). The activation function for both types of hidden layer was $\phi^A[x_j(t), \theta = 3]$, which had given the best performance for feedforward layers (Fig. 2.5G). While previous sections used a total reward value of 1, here the total reward value was 4, which enabled the agent to learn more quickly. Alternatively, the learning rate of the critic can be increased to achieve similar learning outcomes.

Fig. 2.6C shows that with probe sessions on Sessions 2 (PS1), 9 (PS2) and 16 (PS3), the feedforward and reservoir agents with working memory attained visit ratios comparable to or better than the approximately 36% attained by animals on PS3 in a similar task (Tse et al. 2007). Feedforward and reservoir agents without the bump attractor to provide working memory had visit ratios comparable to those of Classic and Linear Hidden Layer agents and to that of chance performance. Fig. 2.6D similarly shows that the nonlinear feedforward and reservoir agents with working memory exhibited decreases in latency to the correct reward location that were markedly better than the decreases in latency of agents without working memory. Successful learning can also be seen in the value and policy maps and example trajectories of an example reservoir agent (Supplementary Fig. 2.2A). Strikingly, the Reservoir agents learned faster than the Nonlinear Hidden Layer agents, showing an advantage of a recurrent reservoir over a feedforward layer [Fig. 2.6C–D].

2.4 Discussion

We have shown that adding a nonlinear hidden layer to classic actor-critic agents with biologically plausible synaptic plasticity enables them to learn multiple paired

association navigation. A nonlinear hidden layer that was a feedforward layer was sufficient, but even faster learning was obtained with a recurrent reservoir. Deep reinforcement learning actor-critic agents learn the task, but do not have biologically plausible plasticity (Botvinick et al. 2020). We verified that Classic actor critic agents with biologically plausible plasticity learn to navigate to a single reward location (Foster et al. 2000; Frémaux et al. 2013), and showed that they also adapt to reward location displacement, yet could not learn multiple paired association navigation. Addition of hidden layers to actor-critic agents had been discussed, but largely unused or not explicitly used in investigations of their capabilities (Barto et al. 1983; Houk, Adams, and Barto 1994). In a sense, the Classic agents implicitly contain hidden layers, since they use place cells, which are constructed in part by many layers of cortical circuitry between sensory input and the hippocampus. We should therefore more precisely say that we have added a hidden layer that processes information from place cells and sensory input before sending it to the actor and critic.

We do not have a clear theoretical understanding of when a hidden layer is needed. However, our addition of a hidden layer was motivated by the successes of reservoir computing, in which the internal connections of the recurrent reservoir are not plastic, and plasticity is restricted to connections that read out from the reservoir (Cazin et al. 2019; Enel et al. 2016; Hoerzer et al. 2012; Maass, Natschläger, and Markram 2002; Sussillo and Abbott 2009; Xiong, Znamenskiy, and Zador 2015; Zhang et al. 2018). Similarly, the hidden layer or reservoir in our agents is not plastic, and plasticity is restricted to connections to the actor and critic that read out from the hidden layer or reservoir. A hidden layer or reservoir has been suggested to facilitate performance of some tasks by representing its inputs in a higher dimensional space (Marr 1969; Albus 1971; Rigotti et al. 2013; Cayco-Gajic et al. 2017; Litwin-Kumar et al. 2017; Cayco-Gajic and Silver 2019). Our results are not inconsistent with a minimum dimensionality for learning the task, but suggest that dimensionality is not sufficient to determine

performance (Cayco-Gajic and Silver 2019; Litwin-Kumar et al. 2017b). Yet, it is unclear why the actor-critic with the reservoir learns the multiple paired association task better than the actor-critic with the nonlinear hidden layer. We postulate that the reservoir's internal noise increases the stochasticity of the input which has been shown to facilitate better policy convergence (An 1996; Asabuki, Hiratani, and Fukai 2018; Neelakantan et al. 2015). Furthermore, we suggest the reservoir's recurrent dynamics convolves the temporal inputs, allowing the reservoir-actor-critic agent to better assign credits, similar to an actor-critic with eligibility traces (Grondman et al. 2012; Kimura and Kobayashi 1998; Sutton and Barto 2020).

The plasticity rules we have used are biologically plausible in the sense that they are functions of a global neuromodulatory factor, presynaptic activity, and postsynaptic activity. Plasticity at actor synapses depends on all three factors, taking the commonly used form of a neuromodulated Hebbian rule (Frémaux and Gerstner 2016). Plasticity at critic synapses depends on a global factor and presynaptic activity, but not postsynaptic activity, taking the form of a two-factor neuromodulated non-Hebbian rule that is less common, but is used to model plasticity at cerebellar Purkinje cells (Medina et al. 2000; Medina and Mauk 1999; Piochon et al. 2013). Non-Hebbian plasticity without postsynaptic activity has also been described at several synapses (Humeau et al. 2003; Lechner and Byrne 1998; Piochon et al. 2013). Interestingly, while the earliest versions of actor-critic agents with biologically plausible plasticity have used a two-factor rule for plasticity at critic synapses, Frémaux and colleagues have successfully used a three-factor rule at critic synapses by using an exponential activation function for the critic (Frémaux et al. 2013).

Beyond the form of the plasticity rules, could the agent's architecture be mapped to anatomical structures in the brain to produce testable hypotheses? Because the dopamine neurons encode some form of TD error that modulates plasticity at corticostriatal (neocortico-striatal and hippocampal-striatal) synapses in the basal

ganglia (P. Read Montague et al. 1996; Reynolds JNJ et al. 2001; Reynolds and Wickens 2002; W Schultz et al. 1997), the actor and critic have most often been suggested to correspond to different basal ganglia divisions (Houk et al. 1994; Joel et al. 2002; Niv 2009). Learning to navigate is broadly consistent with a basal ganglia actor and critic, as hippocampal place cells project strongly to the ventral striatum, with the hippocampus and ventral striatum required or important for learning to navigate (Arleo and Gerstner 2000; Brown and Sharp 1995). While there is tension between such proposals and experiments showing that the dorsolateral striatum is not needed for expressing simple learned allocentric navigation (Packard and McGaugh 1996), both may perhaps be accommodated along the lines of proposals that the ventral and dorsomedial striatum may be more involved in goal-directed learning, while the dorsolateral striatum may be more important for the learning of habits (Everitt and Robbins 2016; Graybiel 2008; Lipton et al. 2019; Yin and Knowlton 2006). Further, ventral tegmental area dopaminergic signals also modulate hippocampal and neocortical plasticity, which may therefore play a role in learning to navigate in addition to corticostriatal plasticity (Palacios-Filardo and Mellor 2019; Seamans and Yang 2004; Sheynikhovich, Otani, and Arleo 2013; Sosa and Giocomo 2021; Xiao, Lin, and Fellous 2020).

As place cells in hippocampal CA3 project to CA1, the latter may correspond to the hidden layer in our agent (Muller 1996). Since CA1 place cells can form without CA3 place cells (Brun et al. 2002; Moser et al. 2015), the hidden layer may also correspond to prefrontal and parietal cortical regions that are downstream of CA3 and CA1 and required for navigation (De Bruin, Swinkels, and De Brabander 1997; Ethier et al. 2001; Kesner, Farnsworth, and DiMattia 1989; Kolb et al. 1994; Negrón-Oyarzo et al. 2018; Poucet and Hok 2017; Sutherland, Wishaw, and Kolb 1988; Whitlock et al. 2008). Damage to the prefrontal cortex slows, but does not prevent learning to navigate to a single reward location after extensive training, consistent with our results that the

task does not require a hidden layer (Whitlock et al. 2008). Postulating that the hidden layer exists in the prefrontal cortex predicts that prefrontal damage would nonetheless prevent learning multiple paired association navigation.

However, such an effect of prefrontal damage would seem to be also explainable by other hypotheses. To aid the design of experiments that could distinguish them, future computational work would refine the biological plausibility of the present agent, and investigate alternative agent architectures. Agent refinement may include development of a version with spiking neurons, and incorporation of biological details such as timing effects of dopaminergic plasticity modulation and effects of other neuromodulators (Brzosko et al. 2015; Pawlak et al. 2010; Yagishita et al. 2014). Such considerations may also suggest other agent architectures. Recent experimental data on acetylcholine and dopamine led to an agent without an actor-critic architecture that learns single reward locations (Brzosko et al. 2017; Zannone et al. 2018). Anatomical considerations have led to actor-critic agents that do not depend on the TD error (O'Reilly et al. 2007; O'Reilly and Frank 2006). However, whether these agents can learn multiple paired association navigation has not been studied yet. It would also be interesting to evaluate agents on cued task switching, which resembles multiple paired association learning in requiring context-dependent behaviour, and is often considered indicative of cognitive flexibility (Monsell 2003; Schneider and Logan 2009; Stokes et al. 2013; Wallis, Anderson, and Miller 2001). Finally, multiple paired association navigation has been part of a suite of tasks to investigate few-shot learning (Tse et al. 2007, 2011). Few-shot learning to displaced single reward locations is displayed by rodents (Steele and Morris 1999), and had also been addressed by the modelling work of Foster and colleagues (2000). Their classic actor-critic agent did not exhibit few-shot learning for displaced single reward locations, nor did any of the actor-critic agents in the present chapter based on their classic actor-critic agent. However, Foster and colleagues (2000) demonstrated few-shot learning by a coordinate-learning actor-critic agent augmented

with dead reckoning capability. In separate work, we have built on their coordinate learning and dead reckoning-based framework to demonstrate agents capable of few-shot learning in multiple paired association navigation (Kumar et al. 2021). We do not know whether few-shot learning can be performed by other variations of the type of actor-critic agents we have studied in the present chapter that do not have built-in dead reckoning. We also do not know what other agents without built-in dead reckoning might perform few-shot navigation learning, and it remains a challenge to extend current biologically-plausible agents to perform comparably to animals on all navigation tasks.

CHAPTER 3 One-shot learning of paired association navigation with schemas and reward-modulated Hebbian plasticity (Kumar et al., 2021)

(The contents of this chapter have been published. Please refer to page VI for details.)

Abstract

Schemas are knowledge structures that can enable one-shot learning. Rodent one-shot learning in a multiple paired association navigation task has been postulated to be schema-dependent. However, the correspondence between schemas and neural implementations remains poorly understood, and biologically plausible computational models of the rodents' learning had not been demonstrated. Here, we compose such an agent from schemas with biologically plausible implementations. The agent contains an associative memory component that can form one-shot associations between sensory cues and goal coordinates. This is implemented using a reservoir of recurrently connected neurons which receives sensory cues as inputs and whose output weights are modified using a novel 4-factor Exploratory Hebbian (EH) rule. Adding an actor-critic allows the agent to succeed even if obstacles prevent navigation by direct heading. We also show that temporal-difference learning of a working memory gate enables one-shot learning even if each cue is transiently presented, replicating the rodent behaviour.

3.1 Introduction

Schemas are mental frameworks of relationships among information and actions. Schemas can aid learning. For example, one often better recalls content from a lecture on a subject in which one already has a framework for understanding. To make the biological mechanisms of schema-dependent learning accessible to investigation with the techniques of experimental neuroscience, Tse and colleagues devised a behavioural paradigm in which rodents demonstrated more rapid learning after an initial learning experience, analogous to that displayed by people during schema-dependent learning (Tse et al, 2007). Rodents performed a two-stage multiple paired associations (MPA)

task. In the first stage, rodents were given a cue at the start of each trial, indicating where they had to go to get a reward. Different cues were presented on different trials, and rodents learned to associate different cues with different target locations. Learning was relatively slow in the first stage. In the second stage, new cues were given, and learning was rapid, with rodents demonstrating one-shot learning, needing only a single exposure to each new cue to navigate to the correct location on subsequently encountering the cue.

Machine learning algorithms from fields like transfer learning and meta-learning behave similarly, with prior learning accelerating subsequent learning to the point of being few-shot or one-shot (Hospedales et al, 2020; Ravi and Larochelle, 2016; Finn et al, 2017; Wang et al, 2018; Ritter et al, 2018). Such algorithms have been adapted for modelling scenarios resembling the experimental paradigm of Tse and colleagues (McClelland, 2013; Hwu et al, 2020). However, those algorithms depend on backpropagation or contrastive Hebbian learning, which are not biologically plausible, as synaptic plasticity in backpropagation is acausal or nonlocal, while contrastive Hebbian learning depends on synaptic rules that differ in alternating phases (Murray, 2019; Bellec, 2020; Lillicrap et al, 2020). Accordingly, we present here an agent in which synaptic plasticity is governed by biologically-plausible rules, and that replicates the one-shot learning of rodents. Given that Tse and colleagues devised their experimental paradigm to address schema-dependent learning, we also explicitly describe schemas that could underlie the observed rodent behaviour, and explain how each schema has a counterpart biologically-plausible implementation in the agent.

We build on work by Foster and colleagues (2000), who demonstrated a biologically-plausible agent that modelled rodent one-shot learning of a delayed matching-to-place (DMP) task. In the DMP task, rats are required to navigate to a target whose position remains the same throughout four trials each day, but whose position is changed every day. During the first few days, the time taken to find the newly displaced target

gradually decreases from trial to trial, but after several days, rats find the target on the second trial (Steele and Morris 1999). Their agent may be thought of as composed of 3 schemas. (1) LEARN METRIC REPRESENTATION allows the agent to learn coordinates that are a continuous metric representation of its current position; this schema was neurally implemented with biologically-plausible synaptic plasticity involving a generalized vector temporal difference (TD) error. (2) LEARN GOAL COORDINATES allows the agent to learn the goal coordinates in one shot; this schema had a non-neural, symbolic implementation to store the target coordinates at which a reward was disbursed. (3) NAVIGATE allows the agent to perform vector subtraction between coordinates of its current and goal locations to obtain a direction in which to head to reach the goal; this schema had a non-neural, symbolic implementation. This agent shows gradual learning before transitioning to one shot learning because it executes 2 learning schemas: LEARN METRIC REPRESENTATION learns slowly; LEARN GOAL COORDINATES always learns in one shot. Initial learning is gradual as it involves both the schemas, but then becomes one shot once LEARN METRIC REPRESENTATION has completed learning and new learning depends only on LEARN GOAL COORDINATES.

Because the LEARN GOAL COORDINATE schema is a non-associative memory that stores the coordinates of a single goal, the agent of Foster and colleagues is unable to learn the MPA task; this schema received only a non-neural, symbolic implementation. We therefore replaced it with the LEARN FLAVOUR-LOCATION schema, which is an associative memory that is able to learn each of several multiple cue-location paired associations in one shot.

We demonstrate two agents in which the LEARN GOAL COORDINATE schema is replaced with the LEARN FLAVOUR-LOCATION schema, enabling one-shot learning in both DMP and MPA tasks. One of the agents is symbolic, while the other is neural. In both symbolic and neural agents, LEARN METRIC REPRESENTATION

is neurally implemented as by Foster and colleagues. The symbolic agent implements LEARN FLAVOUR-LOCATION symbolically with a key-value matrix to store and recall cue-associated goal coordinates; and NAVIGATE is symbolically implemented as by Foster and colleagues. The neural agent implements LEARN FLAVOR-LOCATION with a reservoir of recurrently connected units whose readout weights undergo synaptic plasticity governed by a novel biologically-plausible 4-factor exploratory Hebbian (EH) learning rule to learn a flavour-location association after one trial; additionally, NAVIGATE is neurally implemented by using backpropagation to train a network whose input-output relationships closely match those of the symbolic NAVIGATE implementation; as backpropagation is not biologically plausible, we assume that the neural implementation of NAVIGATE arises via processes during development or prior experience that we do not model. When these agents are supplemented with an actor-critic, they can demonstrate one-shot learning of new association pairs even in an arena with obstacles. Lastly, we show that if the agent uses a reward prediction error to learn a working memory gating policy, the agent demonstrates one-shot learning even if the cue is presented only at the start of a trial, and despite distractor stimuli being present during navigation.

3.2 Methods

3.2.1 General neuron model

The membrane potential dynamics $x_i(t)$ of all neurons, except place cells and units within the neural NAVIGATE schema, were simulated using

$$\tau \dot{x_j}(t) = -x_j(t) + \sum_{i=1}^{N} W_{ij} u_i(t) + \sqrt{\tau \sigma^2} \xi(t) \qquad (1)$$

with membrane time constant $\tau = 100\, ms$, inputs $u_j(t)$ linearly weighted using synaptic weights W_{ij} and stochasticity defined using Gaussian white noise process $\xi(t)$

with zero mean and unit variance and tuned individually using σ. The firing rates are modelled using either a linear or nonlinear activation function. Each neuron's dynamics was discretized with the Euler–Maruyama method:

$$x_j(t) = (1-\alpha)x_j(t-\Delta t) + \alpha\left(\sum_{i=1}^{N} W_{ij}u_i(t) + \sqrt{\frac{\sigma^2}{\alpha}}N(0,1)\right) \qquad (2)$$

where $\alpha \equiv \Delta t/\tau$ and $N(0,1)$ is the standard normal distribution. We used a time step of $20\ ms$ for all simulations. The specific implementation is outlined in the subsequent sections.

Model-free reinforcement learning

The biologically plausible Actor-Critic was adapted from (Kumar et al. 2022). All agents have 49 place cells whose firing rates depend on the agent's position in the arena $s(t)$. The firing rate of the ith place cell is

$$u_i^{pc}(t) = \exp\left(-\frac{(s(t)-s_i)^2}{2\sigma_{pc}^2}\right) \qquad (3)$$

with $\sigma_{pc} = 0.267\ m$, and place cells centers s_i spaced $0.267\ m$ apart at the intersections of a regular 7-by-grid. Each cue is encoded by u^{cue}, a one-hot vector of length 18 with gain 3, for example, $u^{cue} = [0,3,0,\dots]$ for the second cue. The cue and u^{cue} were constant throughout each trial, except for the working memory task in Figure 3.7 where the cue was presented 1 second after the start of each trial for 2 seconds, similar to the experiment by Tse et al. (2007). During the cue presentation period, place cell activity and agent actions were silenced to simulate cue presentation to the rat in the starting box with no knowledge of its position in the maze. Subsequently, u^{cue} was set to zero while place cell activity and agent actions were switched on for navigation.

The place cell activities and sensory cue were concatenated to form an input vector

$$u_i(t) = \left[u_i^{pc}(t), u_i^{cue}(t)\right] \tag{4}$$

with length $N_{inputs} = 67$ and passed to the reservoir of recurrently connected neurons as inputs. The firing rates of the reservoir neurons are

$$r_j(t) = \phi\left[x_j(t)\right] \tag{5}$$

with the nonlinear activation function

$$f(x) = \begin{cases} 0, & x < 3 \\ x, & x \geq 3 \end{cases} \tag{6}$$

And membrane potential dynamics

$$\tau \dot{x}_j(t) = -x_j(t) + \sum_{j=1}^{N_{inputs}} W_{ij}^{inp} u_i(t) + \lambda \sum_{k=1}^{N} W_{jk}^{rec} \tanh\left[x_k(t)\right] + \sqrt{\tau \sigma_{res}^2}\xi(t) \tag{7}$$

with $\lambda = 1.5$ and $\sigma_{res} = 0.025$. The synaptic weights W_{ij}^{inp} were drawn from a uniform distribution between $[-1,1]$, W_{ij}^{rec} from a Gaussian distribution with zero mean and variance $1/pN$ with connection probability $p = 0.1$.

All agents have an actor of $M = 40$ neurons with firing rate of the kth actor neuron

$$\rho_k(t) = \mathrm{ReLU}[q_k(t)] \tag{8}$$

With the rectified linear unit activation (ReLU) function and membrane potential q_k has dynamics

$$\tau\dot{q}_k(t) = -q_k(t) + \beta^{control} q_l^{NAV}(t) + (1 - \beta^{control}) \sum_{j=1}^{N} W_{jk}^{actor} r_j(t) + \sum_{h=1}^{M} W_{hk}^{lateral} \rho_h(t) + \sqrt{\tau\sigma_{actor}^2}\xi(t) \quad (9)$$

With $\sigma_{actor} = 0.25$. q_l^{NAV} is the input from the symbolic or neural NAVIGATE schema. The synaptic weights W_{jk}^{actor} linearly weight the reservoir activity. $\beta^{control}$determines the controls the contributions from the reservoir and the NAVIGATE schema in controlling the agent's actions. $\beta^{control}$ takes values between 0 and 1 inclusive to be either a pure Actor-Critic or schema agent respectively. $\beta^{control} = 0.3$, $\beta^{control} = 0.4$ and $\beta^{control} = 0.9$ were used for the navigation tasks in Figure 3.4, 3.6 and 3.7 respectively. $W_{hk}^{lateral}$ was defined using

$$W_{hk}^{lateral} = \frac{w_-}{M} + w_+ \frac{f(k,h)}{\sum_h f(k,h)} \quad (10)$$

with $f(k,h) = (1 - \delta_{kh})e^{\varphi \cos(\theta_k - \theta_h)}$, $w_+ = 1$, $w_- = -1$ and $\varphi = 20$, connect the actor neurons into a ring attractor that smooths the agent's trajectory. The kth actor neuron represents a spatial direction $\theta_k = 2\pi k/M$ and the action

$$a(t) = \frac{a_0}{M} \sum_k \rho_k(t)[\sin\theta_k, \cos\theta_k] \quad (11)$$

is the vector sum of directions weighted by each actor neuron's firing rate, with $a_0 = 0.03$ translating to the agent moving at about $0.8\, ms^{-1}$. Agents with a critic neuron has firing rate

$$v(t) = \mathrm{ReLU}[\varsigma_k(t)] \quad (12)$$

with membrane potential dynamics

$$\tau\dot{\varsigma}_k(t) = -\varsigma_k(t) + \sum_{j=1}^{N} W_{jk}^{critic} r_j(t) + \sqrt{\tau\sigma_{critic}^2}\xi(t) \quad (13)$$

with $\sigma_{critic} = 1^{-8}$ and W_{jk}^{critic} is the synaptic weights from the reservoir. The output of the critic $v(t)$ and the reward $R(t)$ in Eq. 49 define the continuous temporal difference (TD) error (Doya 2000; Frémaux et al. 2013)

$$\delta^{DA}(t) = R(t) + \dot{v}(t) - \frac{1}{\tau_g} v(t) \quad (14)$$

and discretised according to Kumar et al. (2022)

$$\delta^{DA}(t) = R(t - \Delta t) + \left[v(t) - \left(1 + \alpha_g\right)v(t - \Delta t)\right] \quad (15)$$

with $\alpha_g \equiv \Delta t/\tau_g$ and, $\tau_g = 3000\, ms$ for Figures 3.3, 3.5 and 3.7 that did not have obstacles and $\tau_g = 10{,}000\, ms$ for Figures 3.4 and 3.6 that required the agents to navigate past obstacles (Table 1). Synaptic plasticity of the weights onto the critic is governed by the two-factor rule

$$\Delta W_{jk}^{critic}(t) = \eta_{critic} \cdot r_j(t) \cdot \delta^{DA}(t) \quad (16)$$

with the presynaptic reservoir firing rate $r_j(t)$ modulated by the continuous TD error (Foster et al. 2000; Sutton and Barto 2020). Synaptic plasticity of the weights from the reservoir to the actor is governed by a three-factor rule

$$\Delta W_{jk}^{actor}(t) = \eta_{actor} \cdot\ r_j(t) \cdot \rho_k(t) \cdot \delta^{DA}(t) \quad (17)$$

with the outer product of the presynaptic and postsynaptic activity modulated by the TD error. The learning rates were optimised for each task using a grid search between 0.000001 to 0.0001 for the actor and between 0.00001 to 0.001 for the critic (Table. 1).

Table 1. Actor-Critic learning hyperparameters for each task

Task	Figure	R	τ_g (ms)	η_{critic}	η_{actor}
DMP	3.3	5	3000	0.0002	0.00002
Single goal + Obstacle	3.4	1	10,000	0.0001	0.00001
MPA	3.5 & 3.7	5	3000	0.0002	0.00005
MPA + Obstacle	3.6	1	10,000	0.0001	0.000005

3.2.2 LEARN METRIC REPRESENTATION algorithm

The firing rate of the X and Y coordinate cells follow the linear dynamics of the membrane potential

$$\tau \dot{p}_j(t) = -p_j(t) + \sum_{i=1}^{P} W_{ij}^{coord} u_i^{pc}(t) + \sqrt{\tau \sigma_{coord}^2} \xi(t) \qquad (18)$$

Where $\sigma_{coord}^2 = 1e-8$ and W_{ij}^{coord} is the synaptic weights from place cells to coordinate cells. Although place cells encode spatial information, it is binned instead of a continuous representation of the environment for the agent to perform vector-based navigation. The coordinate cells are an explicit metric representation of the continuous space in the maze that the agent can use to self-localise and flexibly perform vector navigation without needing a lookup table (Bush et al. 2015; Fiete et al. 2008).

Learning a metric representation is formulated as a path integration learning problem by integrating place cell activity and self-motion information $\hat{a}(t)$. The self-motion information is the actual displacement of the agent in an environment after correcting for bouncing off boundaries, making it different from the action $a(t)$ specified by the agent (Eq. 11). An agent can estimate its current coordinates by performing vector addition between its displacement in the arena and the previously estimated coordinates

$$p_j(t) = p_j(t - \Delta t) + \hat{a}_j(t) \tag{19}$$

which yields the self-consistency equation

$$p_j(t) - p_j(t - \Delta t) - \hat{a}_j(t) = 0 \tag{20}$$

if the agent performed perfect path integration to accurately estimate its current coordinates. Path integration errors can be converted into a temporal difference error

$$\delta_j^{coord}(t) = p_j(t) - p_j(t - \Delta t) - \hat{a}_j(t) \tag{21}$$

which the agent minimises by computing an eligibility trace of the place cell activity

$$\tau_{coord}\dot{e}_i(t) = -e_i(t) + u_i{}^{pc}(t) \tag{22}$$

with $\tau_{coord} = 1000\,ms$ and using the two-factor Hebbian rule by taking the presynaptic place cell activity modulated by the path integration TD error

$$\Delta W_{ij}^{coord}(t) = \eta_{coord} \cdot e_i(t) \cdot \delta_j^{coord}(t) \tag{23}$$

with $\eta_{coord} = 0.01$. Reducing the time constant τ_{coord} to $200\,ms$ and $100\,ms$ allowed the agent to learn the metric representation though convergence was increasingly slower.

3.2.3 LEARN FLAVOUR-LOCATION algorithm

When an agent navigates around the arena and receives a reward, its current location is taken to be the goal coordinates. A key–value association matrix is used to store the flavour cue in the key matrix and the agent's coordinates concatenated with a recall value of 1 $[x, y, 1]$ into the value matrices respectively (Fig. 3.2B). In the subsequent trial, the cue vector is treated as a query and a distance-based metric

$$A(t) = softmax\left(\beta^{recall} u^{cue}(t) K^T\right) \tag{24}$$

with $\beta^{recall} = 1$ is used to compute the memory index $A(t)$ which informs if and where the flavour cue is stored in the key matrix. The index recalls the corresponding goal coordinates and recall value from the value matrix

$$g(t) = A(t)V \tag{25}$$

The recall value describes the accuracy of recalling the goal coordinates i.e. when the recall of goal coordinates is imperfect, recall value will be lower than 1. If the trial ends and no reward is disbursed, the row of the key and value matrices corresponding to the cue is set to 0 to delete the cue and coordinate association.

Instead of a symbolic key-value matrix, three readout units from a reservoir are trained to recall the X, Y goal coordinates and recall value when it receives flavour cues as inputs. The firing rate of the goal coordinate neurons $g_i(t)$ follows the membrane potential dynamics

$$\tau \dot{g}_i(t) = -g_i(t) + {g_i}^{noisy}(t) \tag{26}$$

with ${g_i}^{noisy}(t)$ given by a vector sum of the reservoir activity and W_{ij}^{goal} synaptic weights

$${g_i}^{noisy}(t) = \sum_{j=1}^{N} W_{ij}^{goal} r_j(t) + \sqrt{\tau \sigma_{goal}^2}\xi(t) \tag{27}$$

as well as the exploratory white noise with $\sigma_{goal}^2 = 0.05$. To form an association between the flavour cue and the agent's coordinates, a target vector $g_i^{associate}(t)$ is determined according to

$$g_i^{assoiate}(t) = [p_i(t), 1] \tag{28}$$

which is a concatenation of the agent's current coordinates and a scalar value one. The synaptic weights were trained either by the reward modulated Least Mean Square

(LMS) rule which takes the presynaptic reservoir firing activity and the vector error between the target vector $g_i^{assoiate}(t)$ and goal coordinate neurons

$$\Delta W_{ij}^{goal}(t) = \eta_{goal} \cdot r_j(t) \cdot \left(g_i^{associate}(t) - g_i(t)\right) \cdot \Theta\big(R(t)\big) \qquad (29)$$

or the reward-modulated Exploratory Hebbian (EH) rule (Hoerzer et al. 2012) that takes the presynaptic reservoir activity and the difference between the noisy and smooth goal neuron firing activity as postsynaptic neurons

$$\Delta W_{ij}^{goal}(t) = \eta_{goal} \cdot r_j(t) \cdot (g_i{}^{noisy}(t) - g_i(t)) \cdot M(t) \cdot \Theta(R(t)) \qquad (30)$$

A sparse modulatory factor $M(t)$ is computed

$$M(t) = \begin{cases} 1, & \bar{P}(t) < P(t) \\ 0, & otherwise \end{cases} \qquad (31)$$

where the performance index $P(t)$ is the negative mean squared error between the target vector $g_i^{associate}(t)$ and goal neuron $g_i(t)$ activity

$$P(t) = -\sum_{i=1}^{3} \left[g_i^{associate}(t) - g_i{}^{noisy}(t)\right]^2 \qquad (32)$$

and a low pass filter of the performance index is $\bar{P}(t)$ given as

$$\tau \frac{d\bar{P}}{dt}(t) = -\bar{P}(t) + P(t) \qquad (33)$$

with $\tau = 100\,ms$, the same as the neuronal time constant. The Exploratory Hebbian rule is considered to be biological as it uses only local presynaptic and postsynaptic information while the modulatory factor is a sparse scalar value (Hoerzer et al. 2012; Legenstein et al. 2010).

Importantly, the Hebbian plasticity rule needs to be modulated by the presence of the reward so that the coordinates at which a reward is disbursed is learned as the goal coordinates. A step function Θ is used to transform the reward value

$$\Theta(R(t)) = \begin{cases} 1, & R(t) > 0 \\ 0, & otherwise \end{cases} \quad (34)$$

so that reward modulation is 0 for negative or no rewards or 1 for positive rewards. If the trial ends $t = T_{max}$ with no reward $R(t) = 0$, a zero target vector $g_i^{forget}(t)$ of size 3 is used, instead of $g_i^{associate}(t)$, to associate the cue to a vector that has a recall value of zero

$$g_i^{forget}(t) = [0,0,0] \quad (35)$$

The association of the cue to a zero vector can be formed by training the synaptic weights using the same equations for associations (Eq. 29–33) with reward modulation set to 1.

3.2.4 NAVIGATE algorithm

Direct heading is a simple implementation of vector-based navigation. Vector subtraction is performed between the goal and agent's coordinates

$$d_{j\in\{x,y\}}(t) = g_{j\in\{x,y\}}(t) - p_j(t) \quad (36)$$

to determine the direction to move towards the goal from an agent's current position. A spatial direction that is closest to the computed vector $d_{j\in\{x,y\}}(t)$ is chosen out of the 40 possible directions defined by the actor (Eq. 11)

$$q_l^{NAV}(t) = \text{softmax}\left(\sum_{j=1}^{M} K_i^{actions} d_j\right) \cdot \varepsilon(t) \quad (37)$$

to directly head towards the goal. If the recall value $g_{j=3}(t)$ is less than the pre-set threshold value of 0.6, the output is suppressed

$$\varepsilon(t) = \begin{cases} 1, & 0.6 < g_{j=3}(t) \\ 0, & otherwise \end{cases} \tag{38}$$

otherwise, the direction of movement $q_l^{NAV}(t)$ is passed to the actor (Eq. 9) to influence the action $a(t)$.

A dataset with different combinations of current $p_j(t)$, goal and recall values $g_j(t)$ as input and the corresponding suppressed or unsuppressed direction of movement $q_l^{NAV}(t)$ was generated using the equations 36–38. A feedforward neural network with two hidden layers, each with 128 neurons with firing activity transformed using the ReLU activation function, and top layer with 40 neurons with linear activation function was trained using backpropagation to minimise the mean squared error of the dataset. The synaptic weights were fixed and uses as a static module as the agent learned the DMP and MPA tasks.

3.2.5 Learning to gate working memory

Since sensory cue is given only at the start of the trial in Figure 3.7, a persistent representation of the cue is necessary to learn flavour-location associations. A bump attractor has been shown to recreate the persistent working memory dynamics in the prefrontal cortex (Parthasarathy et al. 2019; Wimmer et al. 2014). The bump attractor has $N_{bump} = 54$ neurons with firing rate given by

$$u_i^{bump}(t) = ReLU\left[x_i^{bump}(t)\right] \tag{39}$$

where the membrane potential $x_i^{bump}(t)$ has dynamics

$$\tau \dot{x}_i^{bump}(t) = -x_i^{bump}(t) + \chi(t) \cdot \sum_{j=1}^{M_{cue}} W_{ij}^{inwm} u_j^{cue} + \sum_{h=1}^{N_{bump}} W_{ih}^{bump} \omega\left[x_h^{bump}(t)\right] + \sqrt{\tau \sigma_{bump}^2} \xi(t) \tag{40}$$

with $\sigma_{bump} = 0.1$ and nonlinear activation function

$$\omega(x) = \begin{cases} 0, & x < 0 \\ x^2, & 0 < x < 0.5 \\ \sqrt{2x - 0.5}, & x \geq 0.5 \end{cases} \tag{41}$$

The synaptic weight ${W_{hj}}^{bump}$ is defined similarly as the lateral connectivity in the actor (Eq. 10) to connect the neurons in a ring with $w_+ = 2, w_- = -10$ and $\varphi = 300$. Since the 18 cues are encoded as a one-hot vector, W_{ij}^{inwm} is specified such that each cue activates one unit in the ring using a synaptic weight of 1 and the W_{ih}^{bump} activates two adjacent units so that a total of three neurons form a subpopulation to persistently maintain each cue information.

The gating mechanism $\chi(t)$ controls the information flow from the sensory cues to the bump attractor by either opening the gate to update the working memory with new information or closing the gate to maintain the information persistently maintained in working memory (Lloyd et al. 2012; O'Reilly and Frank 2006; Todd, Niv, and Cohen 2009). There are two gating neurons, each to update or maintain working memory respectively and have the membrane potential dynamics

$$\tau \dot{\vartheta}_i(t) = -\vartheta_i(t) + \sum_{j=1}^{N} W_{ij}^{gate} r_j(t) + \sqrt{\tau \sigma_{gate}^2} \xi(t) \tag{42}$$

and use a softmax selection rule to determine the probability of selecting a particular gating action

$$P^{gate}(t) = \frac{\exp\left[\beta^{gate}\vartheta_i(t)\right]}{\sum_k \exp\left[\beta^{gate}\vartheta_k(t)\right]} \quad (43)$$

with $\beta^{gate} = 2$ and, $\pi_i(t) = 1$ if gating action i was chosen at time t and $\pi_i(t) = 0$ otherwise. The gating mechanism then updates or maintains working memory by

$$\chi(t) = \begin{cases} 1, & \pi_1(t) < \pi_2(t) \\ 0, & otherwise \end{cases} \quad (44)$$

opening $\chi(t) = 1$ or closing $\chi(t) = 0$ information flow from the sensory cues to the bump neurons. The synaptic plasticity of the weights $W_{ij}^{gate}$is governed by a 3-factor temporal difference error modulated Hebbian plasticity rule

$$\Delta W_{ij}^{gate}(t) = \eta_{gate} \cdot r_j(t) \cdot \pi_i(t) \cdot \delta^{DA}(t) \quad (45)$$

using the reservoir's presynaptic activity, gating policy as postsynaptic activity and modulated by the TD error computed by the critic with $\eta_{gate} = 0.0001$ and. All synapses that were trained using the Hebbian rule were initialised to zero at the start of the simulations.

3.2.6 Task descriptions

Random foraging

The task was to understand how LEARN METRIC REPRESENTATION schema uses place cell activity to learn a continuous metric representation. The agent had 49 place cells and coordinate cells representing X and Y axis.

The agent moves within a spatially continuous two-dimensional square arena bounded by walls of length 1.6m with possible agent positions $x = (\pm 0.8\, m, \pm 0.8\, m)$. At the start of each trial, the agent's current coordinate estimation was reset to zeros while its position drawn with equal probability from midpoints of the found boundary walls. The

agent moves by executing time-dependent actions $a(t)$ from a random policy that affect its velocity according to

$$\dot{s}(t) = a(t) \tag{46}$$

Using Euler's method of discretization with time step Δt, this results in position updates

$$s(t + \Delta t) = s(t) + \Delta t \cdot a(t) \tag{47}$$

If the updated position ends up outside the area, the agent moves $0.01\,m$ inwards perpendicular to the closest boundary from its last position given by $\hat{a}(t)$. The agent explored the maze over 20 trials for 30 seconds. Synaptic plasticity from place cells to the coordinate cells was switched on according to Eq. 23

To assess the learning of metric representation, the synaptic weights, true state coordinates and agent estimated current coordinates were plotted for trials 2, 9 and 20.

Associating cues to coordinates

This task required the reservoir with three readout units to associate up to 50 one-hot vector cue inputs with 50 goal coordinates randomly drawn from a uniform distribution between [-1,1]. The task was split into two phases, association and recall, where the cue was persistently presented.

During the association phase, the goal coordinate concatenated with a value of one for example [0.4, -0.2, 1], was set as the target vector $g_i^*(t)$. Synaptic plasticity governed either by the Exploratory Hebbian or Least Mean Squares (LMS) rule was switched on for five seconds. There after plasticity was switched off to determine if the network was able to maintain the learned goal coordinates for five seconds. 1 up to 50 cues were presented for association before the recall phase.

During the recall phase, the reservoir's internal activity was reset by drawing each unit's membrane potential from a Gaussian distribution with zero mean and variance

0.1 and the cue was presented as input to the network. The one-shot recall error was determined by taking the mean square error between the target $g_i^*(t)$ and the readout neuron activity $g_i(t)$.

To delete cue specific association, the target vector was set to zeros $[0, 0, 0]$ and synaptic plasticity was switched on for the reservoir to associate the cue input to a zero vector. The number of neurons within the reservoir was increased incrementally from 128 to 2048 to assess if the size of the reservoir affecting the one-shot learning and recall accuracy.

Displaced match to place (DMP)

Following Steele and Morris (1999), the task involved navigating to a single goal in the square maze described in random foraging. A goal location is randomly chosen out of 49 possible reward locations distributed throughout the maze such that the centres of possible locations are $0.2\,m$ from each other or a boundary. All possible reward locations are circles with a radius of $0.03\,m$. A session constitutes of four trials where the goal remains in the same location. In the following session, a new goal location is chosen. Agents solved the task over nine sessions.

The agent is free to explore the area for a maximum duration T_{max} per trial. If it finds the reward before T_{max}, the agent remains stationary until the trial ends to model consummatory behaviour. After the agent reaches the reward, a total reward value $R = 5$ is disbursed at a reward rate $R(t)$ defined by

$$\dot{R}_{decay}(t) = -\frac{R_{decay}(t)}{\tau_{decay}};\ \dot{R}_{rise}(t) = -\frac{R_{rise}(t)}{\tau_{rise}} \tag{48}$$

$$R(t) = \frac{R_{decay}(t) - R_{rise}(t)}{\tau_{decay} - \tau_{rise}} \tag{49}$$

With $\tau_{rise} = 100\,ms$ and $\tau_{decay} = 250\,ms$. When the agent reaches the reward, instantaneous updates

$$R_{rise}(t) \rightarrow R_{rise}(t) + R; R_{decay}(t) \rightarrow R_{decay}(t) + R \qquad (50)$$

Are made such that $R(t)$ integrate to R. To prevent infinitely long trials, trials in which the reward is reached before T_{max} are terminated when $R - 1^{-8}$of the reward has been consumed. Trials in which the reward is not reached before T_{max} are terminated at T_{max}.

Single goal with obstacles

The actor-critic algorithm allows an agent to navigate past obstacles for single goals (Frémaux et al. 2013), while the NAVIGATE scheme only affords direct heading. To determine if a combination of actor-critic and schema could improve an agent's navigation capability, the task was to navigate past obstacles to a single goal found at the centre of the arena with coordinates $(0, 0)$. The goal was surrounded on three sides by an inverted U-shaped obstacle with width $0.08\,m$ and length $0.6\,m$. The agent's starting positions was constrained to either the north, east or west of the arena to remove trials which the agent could solve by direct heading. The total reward value was reduced to $R = 1$ while following the same reward rate as in Eq. 49 The rest of the task parameters remained the same as in the DMP task.

Multiple paired associations (MPA)

To model Tse et al. (2007), the same task parameters were used as in the DMP task and in Kumar et al. (2022) except each session comprised of six trials with each of the six possible cues were given to the agent in a random sequence. Cues were given to the agent throughout the trial while the total reward value was kept at $R = 5$ and followed the reward rate disbursement in Eq. 49.

MPA with obstacles

To increase the complexity of the navigation task, obstacles were introduced to the multiple paired association arena. The arena was divided in the centre by an obstacle with width $0.08\,m$ and length $0.8\,m$ with two parallel obstacles from the west to east of width $0.08\,m$ and length $0.68\,m$. These obstacles did not cover the goal locations as in the original paired association, new paired association and new maze configuration described in Tse et al. (2007). The same task parameters were used as in the MPA task though the total reward value was kept at $R = 1$.

Table 2: Possible starting positions for trials with specific FLAVOUR-LOCATION pairs.

Cues given during trial	Possible starting positions
1, 4, 5, 7, 11, 13, 14	East
3, 4, 5, 6 ,8, 15, 16	North
2, 3, 6, 8, 12, 15, 16	West
1, 2, 3, 4, 7, 11, 12, 13, 15	South

To prevent the agents from reaching the goals by direct heading, the starting position of the agent was constrained to Table 2 so that the goal can only be reached by navigating past obstacles. For example, the starting position for cue 1 is randomly chosen to either be the east or south while starting position for cue 2 is either the west or south.

MPA with transient cue and distractor

To fully replicate the biological experimental conditions in Tse et al. (2007), the flavour cue was given to the agent one second after the trial started for two seconds. During

this cue presentation period, the place cell activity and agent's actions were silenced to simulate the rat in the starting box with no knowledge of its position in the maze. The sensory cue was then switched off, setting the sensorial cue activity to zero and place cell activity and agent's actions switched on for navigation. The task relevant cue was not given to the agent thereafter. Instead, distractor cues 17 and 18 were chosen randomly and presented six seconds after navigation has commenced at the frequency of 0.2 Hz. The distractor was presented for one second and either once or twice within a trial.

Code availability

Code for all our models and simulations are available at https://github.com/mgkumar138/Schema4One.

3.3 Result

We begin by describing three schemas, LEARN METRIC REPRESENTATION, LEARN FLAVOUR-LOCATION and NAVIGATE. Thereafter, we verify the ability of three agents, Actor-Critic, Symbolic and Neural, to learn the displaced match to place (DMP) task which requires agents to navigate to a single goal that is displaced to a new location every four trials. We then demonstrate the ability of these agents and hybrid actor-critic-schema agents to navigate to a single goal enclosed by obstacles. Subsequently, we study the ability of the three agents to demonstrate one-shot learning in the multiple paired associations (MPA) task in an open arena. Next, we demonstrate the ability of the hybrid agents to navigate past obstacles and learn new PAs after a single trial. Finally, we demonstrate that a gating policy can be learned using the reward prediction error to ignore distractors and pass only the relevant cue information into a bump attractor to solve the multiple PA task. This task resembles the biological experiments where the sensory cue is presented only at the start of each trial, requiring the agent to hold the task relevant cue in working memory.

3.3.1 Schemas for one-shot navigation to multiple goals

Here, we elaborate on three schemas necessary for one-shot navigation to multiple goals, by outlining the computational problems they solve, the symbolic algorithm and the corresponding neural implementation.

LEARN METRIC REPRESENTATION schema

The first schema is to learn a metric representation of the environment which the agent can use to self-localize and subsequently compute the direction to a goal from any arbitrary location. Although place cells and grid cells provide self-localization information, the former is arbitrarily anchored to environmental landmarks while the latter is a noncontinuous, binned representation of space (Moser, Kropff, and Moser 2008). These representations make it difficult to compute translation vectors from any position to a goal (Fiete et al. 2008), especially for distant locations (Bush et al. 2015). Instead, by transforming the place or grid cell activity to a continuous spatial metric, distance and direction to a goal can be efficiently calculated using vector subtraction, reducing the need to search through previously used solutions or needing a lookup table (Bush et al. 2015; Fiete et al. 2008; Foster et al. 2000).

A) Computational problem: Learning a continuous metric representation for NAVIGATE (LEARN METRIC)

Algorithm 2 LEARN METRIC schema pseudocode uses place cell activity to learn a continuous X,Y coordinate metric representation in any environment for vector navigation.

Neural implementation of LEARN METRIC.
Place cells synapse to X, Y coordinate cells while synapses are learned using the path integration temporal difference error modulated Hebbian plasticity rule with eligibility trace.

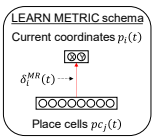
Initialise network weights $W^{coord} \leftarrow 0$
Initialise start coordinates $p(0) = (0,0)$
for t < T:
  Move in direction specified by policy
  Estimate coordinates using place cells (18)
  Compute path integration error (21)
  Compile history of place cell activity (22)
  Minimise path integration error (23)
End

$$\tau \dot{p}_i(t) = -p_i(t) + \sum_{j=1}^{P} W_{ij}^{coord} u_j^{pc}(t) + \sqrt{\tau \sigma_{coord}^2}\xi(t) \quad (18)$$

$$\delta_i^{coord}(t) = p_i(t) - p_i(t - \Delta t) - \hat{a}_i(t) \quad (21)$$

$$\tau_{coord} \dot{e}_j(t) = -e_j(t) + u_j^{pc}(t) \quad (22)$$

$$\Delta W_{ij}^{coord}(t) \propto e_j(t) \cdot \delta_i^{coord}(t) \quad (23)$$



B) Computational problem: Associate flavour to location to recall goal after onetrial for NAVIGATE (LEARN FLAVOUR– LOCATION)

Algorithm 3 LEARN FLAVOUR-LOCATION schema pseudocode takes in any flavour cue as input to associate or recall goal coordinates and recall value of 1 as output.

Neural implementation of LEARN FLAVOUR-LOCATION.
Cue vector passed as input to the reservoir with three readout units representing goal coordinates and recall value. Only the synapses from the reservoir to the readout units are learned using 4 -factor reward gated Exploratory Hebbian rule.

Initialise associative memory with random activity
Get cue from environment
for t < T:
  Pass cue to network to recall goal (27)
  if recall value > threshold **then**
    Navigate to goal using NAVIGATE
    if trial ends with no reward **then**
      Delete (30)-(32) flavour-location association by setting target values to 0 (34)
  else
    Explore maze using random policy
    if reward obtained **then**
      Switch on plasticity (30)-(32) to associate flavour to current coordinates estimated using metric representation (28)
End

$$g_i^{noisy}(t) = \sum_{j=1}^{N} W_{ij}^{goal} r_j(t) + \sqrt{\tau \sigma_{goal}^2}\xi(t) \quad (27)$$

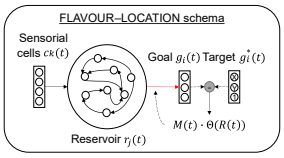
$$g_i^{assoiate}(t) = [p_i(t), 1] \quad (28)$$

$$\Delta W_{ij}^{goal}(t) \propto r_j(t) \cdot (g_i^{noisy}(t) - g_i(t)) \cdot M(t) \cdot \Theta(R(t)) \quad (30)$$

$$M(t) = \begin{cases} 1, & \bar{P}(t) < P(t) \\ 0, & otherwise \end{cases} \quad (31)$$

$$P(t) = -\sum_{i=1}^{3} [g_i^{associate}(t) - g_i^{noisy}(t)]^2 \quad (32)$$

$$g_i^{forget}(t) = [0,0,0] \quad (35)$$



C) Computational problem: Move from current location to goal location (NAVIGATE)

Algorithm 1 NAVIGATE schema pseudocode takes in any current and goal coordinates with recall value as input to output direction to move.

Neural implementation of NAVIGATE.
Two hidden layered neural network was pre -trained by backpropagation on a dataset with different current coordinate, goal coordinate, recall value and actions. Weights were fixed during task learning.

Get agent's current coordinates $p(t)$
Get goal coordinates $g(t)$ with recall value
for t < T:
  if recall value > threshold (37), **then**
    Compute vector subtraction between current and goal coordinates (35)
    Choose action based on direction specified by vector (36)
  else
    No action selected
End

$$d_{j\in\{x,y\}}(t) = g_{j\in\{x,y\}}(t) - p_j(t) \quad (36)$$

$$q_i^{NAV}(t) = \text{softmax}\left(\sum_{j=1}^{M} K_i^{actions} d_j\right) \cdot \varepsilon(t) \quad (37)$$

$$\varepsilon(t) = \begin{cases} 1, & 0.6 < g_{j=3}(t) \\ 0, & otherwise \end{cases} \quad (38)$$

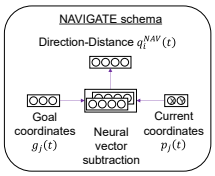


Figure 3.1. Schemas for one-shot navigation to multiple goals. A schema is a framework that specifies the relationship between information and actions to solve a particular computation. A) An agent can use the LEARN METRIC REPRESENTATION schema to learn a continuous metric representation of the arena in the form of X and Y coordinate cells to self-localize and facilitate vector-based navigation by NAVIGATE schema. Place cells anchor the metric representation (Eq. 18). The synapses from place cells to X, Y coordinate cells are learned by minimising a path integration temporal difference error using a two-factor Hebbian rule (Eq. 23). B) The LEARN FLAVOUR–LOCATION schema associates flavour cues to goal coordinates after one trial such that when the same cue is given in the following trial, the schema can be used to recall the corresponding goal coordinates. Synapses from a reservoir to three readout units can be modified using the 4-factor reward-modulated Exploratory Hebbian (EH) rule (Eq. 30) to learn FLAVOUR–LOCATION associations after one trial. The first two units learn the X and Y goal coordinates while the last learns a value of one. A perfect recall is when the third unit's activity approaches a value of one when a cue is given. C) The NAVIGATE scheme performs vector navigation by taking arbitrary current and goal coordinates as input, perform vector subtraction (Eq. 36) and output the direction of movement (Eq. 37). The NAVIGATE computation occurs only when the recall value is greater than the threshold of 0.6 (Eq. 38). A deep neural network was trained by backpropagation to perform these

computations. The synapses of the pre-trained network were fixed since no new learning was necessary to solve the navigation task.

Using the LEARN METRIC REPRESENTATION schema, an agent can gradually learn a metric representation in the form of X, Y coordinates using place cells (Eq. 18 and Fig. 3.1A), or grid cells that have location specific firing activity. This can be reformulated into a path integration learning problem. The agent computes a path integration derived temporal difference (TD) error $\delta_{i\in\{x,y\}}^{coord}(t)$ in the X and Y axis by integrating self-motion information and an estimation of the current and previous coordinates in the arena (Eq. 21). The smooth transitions in place cell activity are captured though an eligibility trace (Eq. 22) and the path integration TD error is minimised using the two-factor Hebbian plasticity rule with the eligibility trace as the presynaptic factor modulated by the temporal difference error (Eq. 23).

As the agent explores the arena, the path integration TD error is gradually minimised (Supplementary Fig. 3.1A) till the synaptic weights from the place cells to coordinate cells converge to a stable representation (Fig. 3.2A top). When the agent moves left to right or bottom to top, the firing activity of the X and Y coordinate cells respectively increase linearly. This translates to an increasing similarity between the agent's coordinate estimation of its current position and the true state coordinates (Fig. 3.2A bottom).

LEARN FLAVOUR–LOCATION schema

The second schema LEARN FLAVOUR-LOCATION is to associate flavour cues to goal coordinates after one trial such that the same goal coordinate is accurately recalled in the subsequent trial. If an agent is rewarded at a particular location, the agent's current coordinates are stored as the goal coordinates. The presence of a reward is used to gate the association of the flavour cue and goal coordinates.

The symbolic agent uses a key–value matrix to store the flavour cue vector and the concatenated goal coordinates with recall value of one in the key and value matrix respectively (for example cue 2 in Fig. 3.2B). In the subsequent trial when a flavour cue is given, a distance-based metric (Eq. 24) is used to compare the cue-based query against the key matrix to identify the memory index. This memory index is used to recall the correspondingly stored goal coordinates from the value matrix (Eq. 25). If recall is accurate, the recall value will be close to 1, and if recall is imperfect, the recall value will be closer to 0. If the agent navigates to the goal coordinate and reward is not disbursed before the trial ends, the association is deleted by setting flavour cue and goal coordinates in that memory index to zeros (Fig. 3.2B cue 8). Hence, a key-value matrix allows writing and deleting specific flavour-location associations after one-trial.

A reservoir of recurrently connected neurons with three readout units (Eq. 27) can be trained to perform one-trial association. To associate a flavour cue to goal coordinates, the agent's current coordinates concatenated with a value of one (Eq. 28) is treated as the target vector $g^{associate}(t)$. Synapses from the reservoir to the readout neurons can either be trained by least mean squares (LMS) algorithm (Eq. 28) (Kumar et al. 2021) or a more biologically plausible 4-factor reward-modulated Exploratory Hebbian rule (Eq. 30) adapted from Hoerzer et al., (2012). The 4th factor, which is the presence or absence of a reward, is crucial to ensure that the association between the flavour cue and agent's current coordinates is learned only when a positive reward is disbursed.

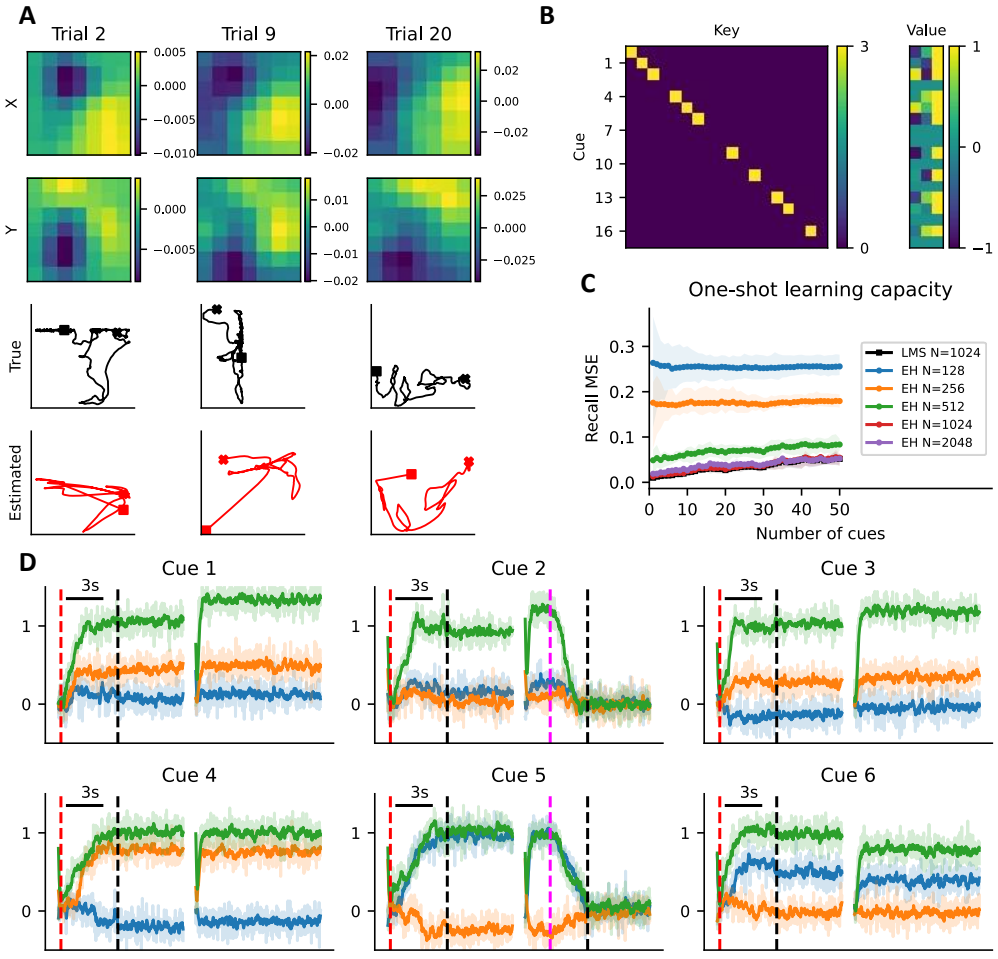


Figure 3.2. Representations learned using LEARN METRIC REPRESENTATION and LEARN FLAVOUR-LOCATION schema. A) As the agent explored the arena, the synaptic weights from 49 place cells to the X and Y coordinate cells converged to a stable representation in the two axes of movement (top). Using the synaptic weights in trial 20, when the agent moved right to left or bottom to top, the firing rates of the X or Y coordinate cells increased respectively. Hence, the agent's ability to self-localize gradually improved to show a higher correspondence with true state coordinates. B) The symbolic agent used a key-value matrix and a distance metric to store and recall cue associated goal coordinates. Goals that were not rewarded were deleted based on the memory index. C) Sufficiently large reservoir (> 512 units) can be trained (using LMS or EH rule) to store and recall up to 50 cue-coordinate paired associations. Although storing more paired associations leads to a monotonic increase in recall error, the recall remains stable and exhibits smooth capacity–accuracy trade-off. D) Example activity of the three reservoir readout units (X coordinate – blue, Y coordinate – orange, recall value – green) that were trained using the Exploratory Hebbian rule to associate, recall, and forget cue-coordinate associations. Plasticity was switched on for 4 seconds from red dashed line to black dashed line to store arbitrary coordinates. Thereafter plasticity was switched off and the network maintained the activity up to 4 seconds. The gap between readout activities for each cue indicates the reservoir membrane potential being reinitialised with random

activity and cue presented to recall the associated coordinate. To forget cue specific associations, plasticity was switched on (magenta to black dashed lines) with the zero vector as the target. Forgetting cue specific goal associations (cue 2 and cue 4) did not affect the recall accuracy of other association pairs.

Reservoirs with three readout units were trained ($N = 24$) to learn one up to 50 cue-coordinate associations for one trial and subsequently the cues were given as input to verify the recall error. The mean square error to recall 50 cue-coordinate pairs decreased from $0.256 \pm 0.030\ (SD)$ to $0.052 \pm 0.022\ (SD)$ as the number of units within the reservoir increased from 128 to 2048 (Fig. 3.2C). The size of the reservoir affected the capacity to learn paired associations after one trial (one-way ANOVA for 50 PAs, $F = 284, p < 0.001$). Reservoir with 1024 neurons, trained by the 4-factor EH rule learned one to 50 flavour-location associations as well as the LMS rule (person's correlation $R = 1.0, p < 0.001$). Unlike autoassociate networks which suffer from catastrophic loss of all patterns once memory cliff is reached (Sharma, Chandra, and Fiete 2022; Tyulmankov et al. 2021), the recall error increased monotonically when more cue-coordinate pairs were learned, from $0.025 \pm 0.007\ s.d.$ for 10 PAs to $0.054 \pm 0.024\ s.d.$ for 50 PAs (Welch's t-test, $T = 5.67, p < 0.001$).

Moreover, the reservoir with readout units can be trained to delete specific cue-coordinate association, like the key-value matrix. Instead of associating the cue to coordinates, the cue can be associated with a zero-target vector $g^{forget}(t) = [0,0,0]$ using the same synaptic plasticity rule. Figure. 3.2C shows that when synaptic plasticity was switched on (red dashed line) to learn cue-coordinate association, the readout units of the reservoir converged to specific X (blue trace), Y (orange trace) coordinates and recall value of 1 within 3 seconds and was maintained after plasticity was switched off (black dashed line). When the reservoir was reset with random activity and presented with the same cue, the readout units recalled the associated goal coordinates with a recall value close to 1 (green trace). When cue 2 and cue 4 associations were deleted

using the zero-target vector (magenta dashed line), the activity of the three readout units fell to zero. More importantly, the deletion of cue 2 and cue 4 associations did not affect the recall of cue 1, cue 3, cue 5 and cue 6 goal coordinates.

NAVIGATE schema

The NAVIGATE schema performs three computations, (1) vector subtraction (Eq. 36) between the goal and current coordinates to determine the distance and direction to the goal from the current location (2) choose a relevant action (Eq. 37) that will bring the agent closer to the goal via direct heading and (3) suppress the action if the recall value falls below the threshold value of 0.6 (Eq. 38). These computations allow the agent to head directly to a goal coordinate from any location, even if it had not traversed that path location prior. This is similar to Rumelhart's (1980) description of a schema where any combination of coordinates can be slotted in to the schema placeholders to infer the direction to move. Vector subtraction and the corresponding action is chosen only if the recall value is greater than a threshold value of 0.6. If the recall accuracy is poor such that the recall value falls below the threshold, the output direction vector is suppressed by returning a zero vector, without specifying the direction of movement.

We pretrained a network with two nonlinear hidden layers each with 128 units using backpropagation on a dataset comprising different current, goal coordinates and recall values as inputs and the relevant action to take, computed symbolically (Eq. 36–38) as outputs. The network weights were fixed throughout the DMP and MPA tasks (purple arrows in Fig. 3.1C and Fig. 3.3A).

3.3.2 One-shot learning to single displaced goal

We begin by verifying if a reservoir-actor-critic agent trained using the temporal difference error (Kumar et al. 2022) and agents that combined the three schemas

outlined in Fig. 3.1 can learn to navigate to a goal that is displaced to a new location after four trials. In each trial, the agent starts at a randomly chosen midpoint of the north, south, east, or west boundaries of a $1.6\ m^2$ arena and receives the same sensory cue on every timestep till it reaches the reward location.

All agents have rate-based neurons and receive input from place cells that encode the animal's location in the arena and sensory cue 1. They have an actor made up of neurons connected in a ring whose output dictates the speed and direction of agents.

The actor-critic agent (Fig. 3.3A left) has an additional critic output that learns a value function to compute the reward prediction error (Eq. 15). Only the synapses from the reservoir to the actor and critic are subject to TD error modulated Hebbian rule (see Methods).

Both the symbolic and neural schema agents learn the metric representation using place cells as inputs and the synapses that project to the X and Y coordinate cells are subject to the path integration TD error modulated Hebbian rule (Eq. 23). The symbolic agent (Fig. 3.3A middle) uses a key-value matrix to store sensory cue 1 and the coordinates at which the reward was disbursed at memory index 1 in the key and value matrices respectively (Fig. 3.2B). Using sensory cue 1 and a distance metric, memory index 1 is identified in the key matrix (Eq. 24) and the goal coordinate is recalled from the value matrix (Eq. 25). The agent's current coordinates and recalled goal coordinates are passed to the symbolic NAVIGATE schema to determine the direction to move (Eq. 37) before passing the direction information to the actor (Eq. 9). The reservoir in the neural agent (Fig. 3.3A right) takes in both sensory cue and place cell activity as inputs and learns the cue-coordinate associations using the 4-factor reward-modulated Exploratory Hebbian rule (Eq. 30) and its current coordinates as the target (Eq. 28). The coordinates learned by LEARN METRIC REPRESENTATION schema and the goal coordinates recalled using the LEARN FLAVOUR-LOCATION schema is fed to

the NAVIGATE schema neural network to determine the distance and direction of movement which is passed as inputs to the 40 actor neurons, similar to the symbolic agent.

Figure 3.3B shows that only the symbolic and neural schema agents showed significantly higher savings in latency between the first and second trials compared to the actor-critic ($p < 0.001$) after the third session (average savings in latency from session 5 to 9, $117 \pm 10\ s$ for symbolic, $84 \pm 8\ s$ for neural), demonstrating one-shot learning of displaced location. The one-shot learning behaviour emerged over 12 trials as the schema agents gradually learned the metric representation to accurately navigate to the goal coordinates. The actor-critic agents showed gradual learning of the goals as the synaptic weights were incrementally updated to converge to a particular value and policy map.

The symbolic schema agent showed significantly higher savings in latency ($t = 105, p < 0.001$) compared to the actor-critic from the first session compared to the neural agent which initially showed worse savings in the first ($t = -208, p < 0.001$) and second ($t = -7, p < 0.001$) sessions but showed significant savings session 3 onwards ($t = 181, p < 0.001$).

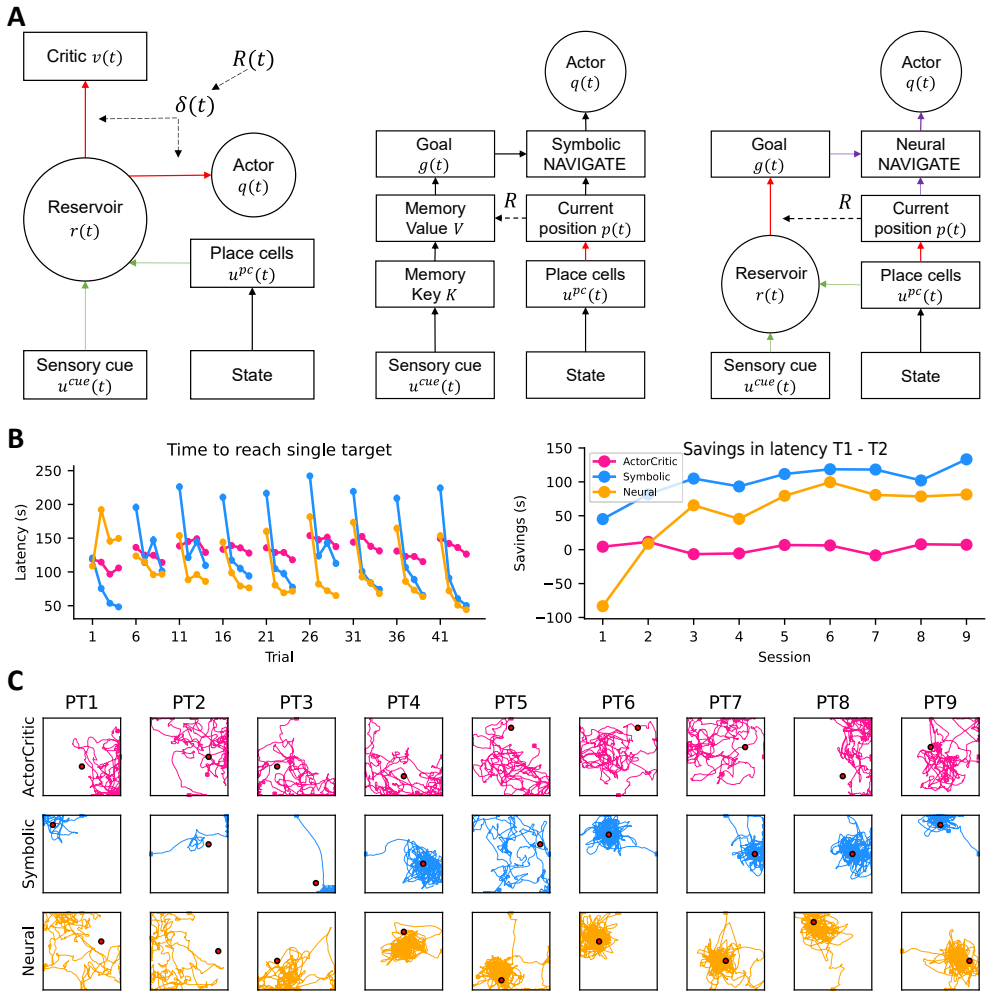


Figure 3.3. One-shot learning of delayed match to place (DMP) task by schema agents. A) Architecture of Actor-Critic (left), Symbolic (centre) and Neural agents. Synapses from the reservoir to the actor and critic were trained using temporal difference error modulated Hebbian plasticity adapted from (Kumar et al. 2022). In both the symbolic and neural agent, the synapses from place cells to coordinate cells were learned using path integration temporal difference error modulated Hebbian plasticity. The symbolic agent uses a symbolic Key-Value memory system whereas the neural agent uses a reservoir with readout synapses trained using reward gated Exploratory Hebbian rule to store and recall the goal coordinates. For the DMP task, all agents were given cue 1 throughout the trial B) Actor-Critic and neural schema agent architecture adapted from Foster et al., 2000 and Kumar et al., 2022. Place cell activity is passed to coordinate cells and these synapses are learned using the velocity based temporal difference error modulated Hebbian plasticity (Foster et al. 2000). C) As an agent explores its environment, it uses self-motion information and place cell activity to estimate its current coordinates. Synapses from place cells to X and Y coordinate cells eventually converge to represent the X and Y axis respectively. D) Estimated coordinates resemble true coordinates as learning progressed. E) Latency (left) to reach single target that is displaced every 4 trials and difference in latency (right) between trials 1 and 2. Agents include Actor–Critic trained by temporal difference error

modulated Hebbian plasticity (pink), Symbolic schema agent using NAVIGATE schema in Fig. 3.1C (blue) and symbolic memory matrix to store target coordinates, and Neural schema agent using pretrained neural network in Fig. 3.2A and target coordinates learned using 4-factor Exploratory Hebbian rule (orange). Both Symbolic and Neural schema agents show one-shot learning of displaced targets session 2 onwards. Error bars indicate standard error. F) Example trajectories of each agent (row) during the probe trial conducted after 4 training trials as the target changed over 9 sessions. 480 simulations per agent with error bars indicating standard error.

However, the average latency to reach a newly displaced goal in the first trial of each session was significantly lower for the neural agent compared to the symbolic agent ($t = -24.3, p < 0.001$), while the average latency to reach the same goal in the second trial of the same session was comparable ($t = -2.48, p = 0.0134$). Although the symbolic agent showed higher savings in latency, the neural agent was more effective in finding the newly displaced goal in the first trial of each session compared to the symbolic agent. Since place cell activity was also passed as inputs to the reservoir, as the agent moved, place cell activity changed and the recall accuracy of goal coordinates wavered between 0.6 to 1 (Supplementary Fig. 3.1B), causing the agent to switch between using the NAVIGATE schema to exploit the recalled goal or to a random policy to explore the arena. Furthermore, the stochasticity in goal coordinate representation (Eq. 27) with $\sigma_{goal} = 0.05$ when recalling the goal location which is $0.03\ m$ in radius allowed the agent to better explore the goal location compared to the symbolic agent that had no stochasticity in goal representation.

Figure 3.3C shows the example trajectory of the three agents during the nonrewarded probe trial after the 4th trial within a training session. The actor-critic agent shows a dispersed trajectory throughout the nine sessions. The symbolic agent heads directly towards the goal, however, the trajectory is suboptimal initially as the agent has not learned a stable metric representation. Hence, the agent uses incorrect coordinates for its current and goal positions to perform vector navigation. For example, in PT 2 and PT3, the symbolic agent misses the goal and spends a longer time at the boundary of

the arena. Navigation to the goal improves after PT3. When the symbolic agent fails to find the goal during the training session, it resorts to a random policy e.g. PT5. The neural agent shows a dispersed trajectory in PT1 and PT2, but the accuracy of navigating towards the goal improves after PT3, like the symbolic agent.

3.3.3 Faster navigation past obstacles to single goal by hybrid agents

Although both symbolic and neural agents perform one-shot learning of single displaced goals, we subsequently compared their ability to navigate past obstacles to a single goal in the centre of the arena. During each trial, agents started from either the north, east or west midpoints of the arena so that a direct path towards the goal was excluded. Training was organised over 60 trials with 18 probe trials during trials 7-12, 30-36 and 54-60.

Besides the actor-critic and schema agents from Figure 3.3A, two additional hybrid actor-critic-schema agents were developed. The first variant used the symbolic implementation, and the second variant used the neural implementation (Fig. 3.4A) of the LEARN FLAVOUR-LOCATION association and NAVIGATE schemas. The actor received inputs from both the reservoir and the NAVIGATE schema (Eq. 9). The contributions by both inputs was optimised using $\beta^{control}$ where $\beta^{control} = 0.3$ means the input to the actor is 30% from the NAVIGATE schema and 70% from the linearly combined reservoir activity. $\beta^{control} = 0$ represents a pure actor-critic agent while $\beta^{control} = 1$ represents a pure schema agent. The temporal difference error time constant was increased from $3000\ ms$ to $10{,}000\ ms$ and learning rates were optimised according to Table. 1 in Methods.

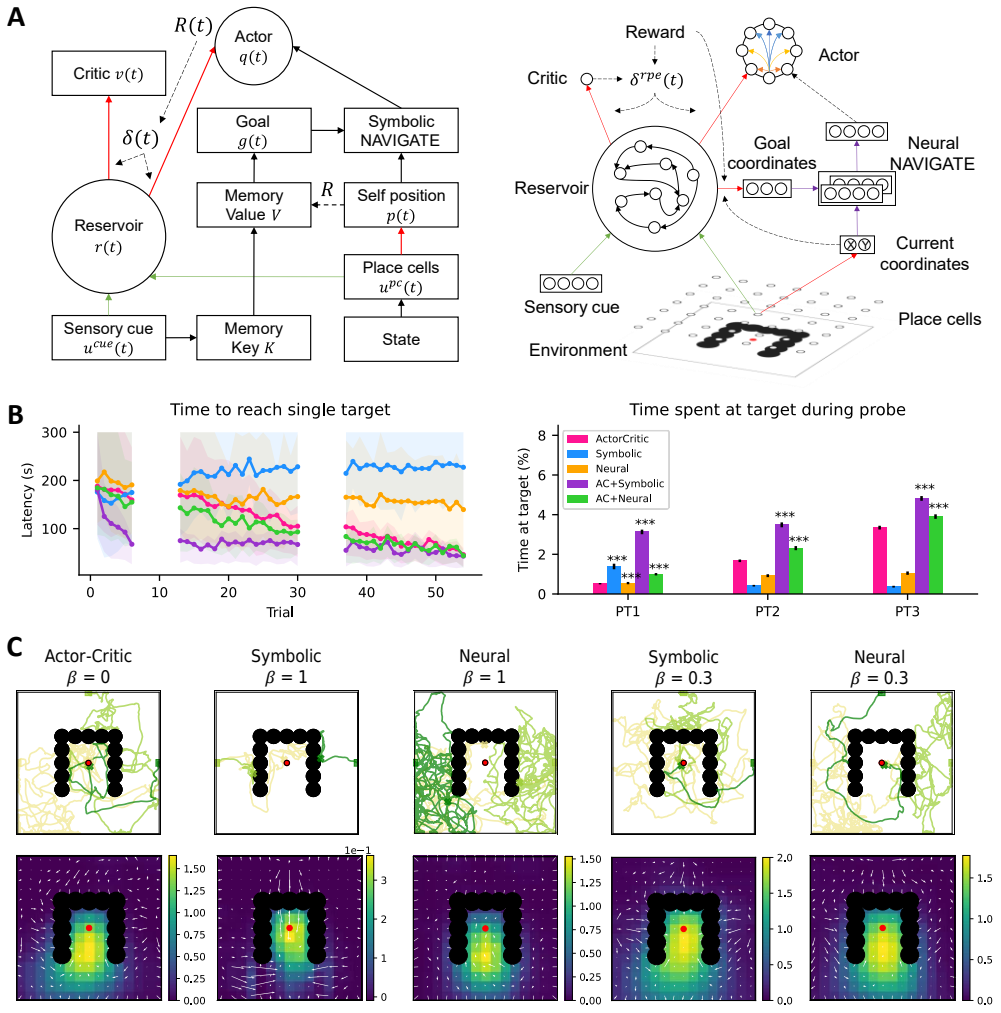


Figure 3.4. Navigating to a single goal past obstacle using a combination of model-free and schema methods. A) Two hybrid actor-critic-schema architectures were developed where actor-critic and symbolic (left) or neural (right) schema algorithms were combined. The actor takes in input from either the reservoir ($\beta = 0$), the NAVIGATE schema ($\beta = 1$) or a linearly weighted combination of both ($\beta = 0.3$) to navigate. Agents start either at the north, east or west of the maze and must navigate to the goal in the centre to obtain a reward. B) Agents that used the actor-critic algorithm either solely (pink) or as a combination with schemas (actor-critic-symbolic – purple, actor-critic-neural – green) showed consistent decrease in latency to reach the goal (top) and spent a higher proportion of time at the goal during the probe trials (bottom) whereas pure schema agents (symbolic – blue, neural – orange) failed to navigate past the obstacle. Agents that used a combination showed faster decrease in latency and spent a significantly higher amount of time at the goal location. C) Example trajectories (top) during probe trials 1 (yellow), 2 (light green) and 3 (green) show both actor-critic ($\beta = 0$) and actor-critic-schema agents ($\beta = 0.3$) choosing different actions to navigate past obstacles and towards the goal, while the pure schema agents ($\beta = 1$) move only by direct heading and get stuck at the obstacle. The policy (white arrows) and value (heatmap) maps (bottom) show the firing activities of the actor and critic respectively. The value map shows which states are more likely to lead to the reward,

which can be used to learn a suitable policy to navigate past obstacles. The pure actor-critic learns a relevant policy while the pure schema agents show an optimal policy when there is no obstacle to block the goal. Agents that use a combination of algorithms learn a mixed policy between the pure actor-critic and schema agents. 240 simulations per agent, shaded area indicates 25^{th} and 75^{th} quantiles while error bars indicate standard error.

Figure. 3.4B shows the latency required to reach the goal in the centre (top) and the average amount of time spent at the goal location during the probe trials (bottom). Only the latency of the actor-critic, actor-critic-symbolic and actor-critic-neural agents decreased to $46\ s \pm 34\ (SD)$, $43\ s \pm 37\ (SD)$ and $44\ s \pm 40\ (SD)$ respectively in the last trial and spent an increasing amount of time at the goal as learning progressed, while the pure symbolic and neural agents showed no improvement in navigation performance with latency in the last trial being $227\ s \pm 100\ (SD)$ and $140\ s \pm 99\ (SD)$ respectively.

The actor-critic agent navigates past obstacles as it learns actions based on the state it is in (Frémaux et al. 2013). Figure 3.4C shows the example trajectory (top) and the critic and actor firing activity (bottom) visualised as a value and policy map for all agents (left to right) during PT3. For the pure actor-critic agent, the critic learns a suitable value function to represent the region in the arena that will lead to a reward and the actor learns a suitable policy to navigate past the obstacles and to the goal.

The pure symbolic agent initially reached the goal, but performance worsened after PT1. This is because in the initial trials, the schema agent was still learning the metric representation, hence the direct heading specified by the NAVIGATE schema did not directly lead to the goal and instead caused the agent to meander (Fig. 3.4C yellow trajectory for symbolic $\beta^{control} = 1$). In the subsequent trials, as the metric representation converged, the NAVIGATE schema specified the agent to head directly to the goal. However, using a direct heading policy cause the agent to get stuck at the

obstacle for the entire trial (Fig. 3.4C green trajectories for symbolic $\beta^{control} = 1$), until the goal coordinate was deleted at the end of the trial when no reward was disbursed. Although the pure neural agent also moved by direct heading, the stochasticity in the goal representation and variable recall value with changing place cell activity caused the agent to straddle between exploiting the goal using the NAVIGATE schema and exploring the arena using a random policy (Fig. 3.4C green trajectories for neural $\beta^{control} = 1$). However, the pure neural agent was unable to demonstrate learning of the task as the actor-critic agent. The pure schema agents learned value and policy maps as the agent architecture was that shown in Figure 3.4A. However, the policy for pure schema agents was not governed by the actor as $\beta^{control}$ was set to 1.

Only the hybrid actor-critic-symbolic agent $\beta^{control} = 0.3$ showed faster decrease in latency (one-way ANOVA $F = 37, p < 0.001$) while the actor-critic-neural agent's latency was comparable to the pure actor-critic agent (one-way ANOVA $F = 2.97, p = 0.0887$). However, both actor-critic-symbolic (PT1: $t = 300, p < 0.001$, PT1: $t = 179, p < 0.001$, PT1: $t = 133, p < 0.001$) and actor-critic-neural (PT1: $t = 95, p < 0.001$, PT1: $t = 72, p < 0.001$, PT1: $t = 52, p < 0.001$) agents spent significantly higher amount of time at the goal location during the probe trials compared to the actor-critic.

The value and policy map for the hybrid schema agents show a mixed policy, partly contributed by the actor-critic and partly contributed by the schema agents. This could facilitate a more optimal policy where the agents navigated away from the obstacle and quickly turned up to move towards the goal by direct heading in the absence of the obstacle (Fig. 3.4C). Hence, the hybrid actor-critic-schema agents can navigate past obstacles by learning state-based actions while showing faster navigation to the goal when a direct path is available.

3.3.4 One-shot learning of multiple new paired associations

Having shown that both the symbolic and neural schema agents demonstrate one-shot learning for single goals in an open arena, we verify their ability to learn multiple new flavour-location paired associations in comparison to the reservoir-actor-critic agent. The task was split into two-parts, the first was to learn six flavour-location paired associations over 20 sessions and the second was to substitute either two (2NPA) or six (6NPA) new flavour-location combinations with the original paired associations (OPA). During each trial, one of six cues was presented throughout, and the agent received a reward only if it reached the correct goal location (insets in Fig. 3.5A right and Fig. 3.5B shows reward location corresponding to the cue). OPA training was organised into 20 sessions, each consisting of six trials across which the agent was exposed to six cues in random order. The 2NPA and 6NPA training was organised into 2 sessions, the first to learn the new paired associates and the second was a nonrewarded probe.

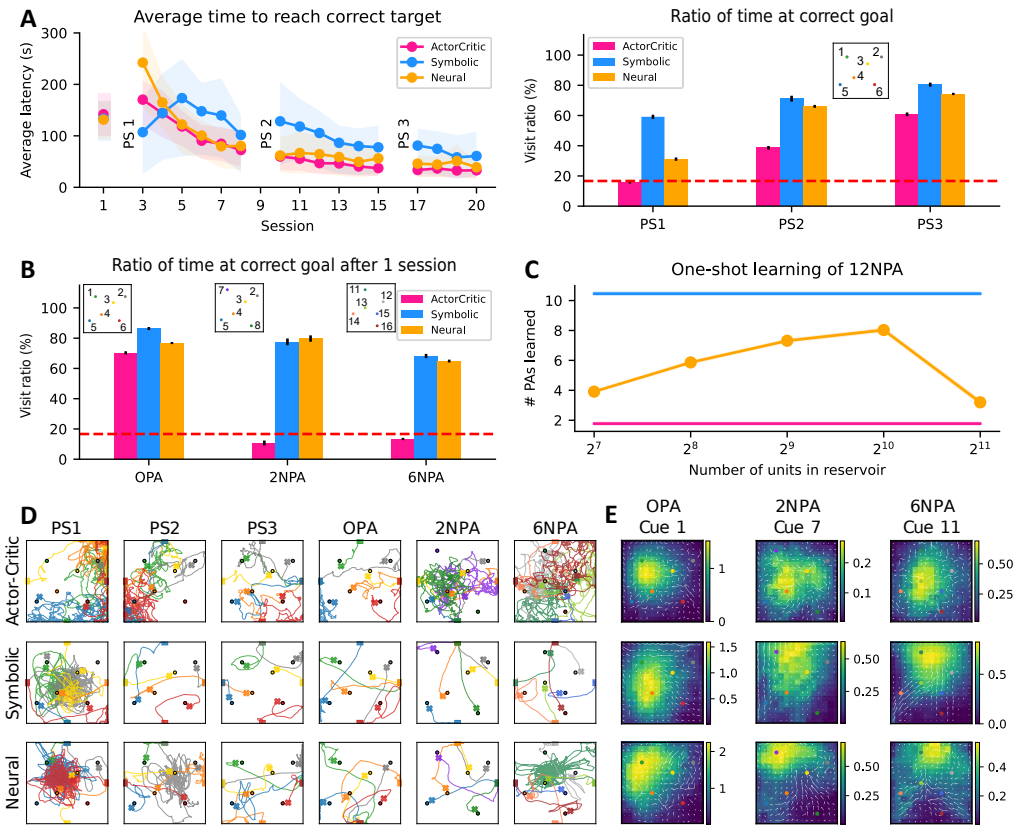


Figure 3.5. Gradual then one-shot learning of multiple new paired associations by schema agents. A) The actor-critic (pink), symbolic (blue) and neural (orange) agents demonstrated a gradual decrease in the average latency to reach six cue specific goals and spent an increasing ratio of time spent at the correct target across probe sessions PS1, PS2 and PS3. B) After 20 training sessions, agents had one training session followed by a probe for three flavour-location combinations, Original Paired Associations (OPA), 2 (2NPA) and 6 New Paired Associations (6NPA). All agents showed above chance visit ratios for the OPA condition while only the symbolic and neural agents showed above chance visit ratios during the 2NPA and 6NPA conditions, demonstrating one-shot learning of two and six new PAs. Chance performance was 16.7% (one out of six targets visited). C) After 20 sessions of training in the OPA condition, agents were introduced to 12 new PAs for a single trial with reward locations randomly chosen out of 43 positions. Actor-Critic agents learned at most two goals while the Symbolic agent learned up to 11 goals. Neural agents showed an increase in one-shot learning capacity when the size of the reservoir was increased from 128 to 1024 units. D) Example trajectories during probe sessions PS1, PS2, PS3 and the subsequent probe sessions OPA, 2NPA and 6NPA. Although all agents navigate to OPA, only the symbolic and neural agents learned the new flavour-location PAs during the 2NPA and 6NPA configurations. E) Actor-Critic agent could not learn distinct value and policy maps after a single trial to navigate to cue 7 and cue 11 goals while both symbolic and neural agents showed cue specific maps after a single trial of learning. 240 simulations per agent, shaded area indicates 25th and 75th quantiles while error bars indicate standard error.

Learning rates and TD error time constants were optimized to maximise the actor-critic agent's learning performance (see Table 1 in Methods). Figure 3.5A shows that the latency required to reach all six goals (left) across sessions gradually decreased to $33\ s \pm 26\ (SD)$, $61\ s \pm 46\ (SD)$ and $39\ s \pm 33\ (SD)$ for the actor-critic, symbolic and neural schema agents respectively during the OPA condition. During the nonrewarded probe sessions, visit ratio was calculated as the amount of time spent within $0.1\ m$ from the centre of the correct goal divided by the time spent within $0.1\ m$ of any of the six possible goals. A visit ratio of 16.7% was consistent with chance performance, where the agent visited all six reward locations equally or visited one location regardless of cue. During the OPA maze condition, all agents showed improvements in visit ratios from PS1 to PS3 and above chance visit ratios in all probe sessions (unpaired t test $p < 0.001$), except the actor-critic agent which showed chance performance for PS1 ($t = -0.75, p = 0.456$) and above chance performance for PS2 ($t = 17.7, p < 0.001$) and PS3 ($t = 36.3, p < 0.001$).

After 20 sessions of learning the OPA maze configuration, the three agents were trained for one session on the OPA, 2NPA and 6NPA configuration followed by a probe session. The 2NPA condition comprised of two new FLAVOUR-LOCATION association pairs where cue 7 and 8 replaced cue 1 and 6 while keeping cues 2 to 5. The 6NPA condition comprised of six new association pairs with cues 11 to 16 replacing all of cues 1 to 6. Unlike in Tse's task, the environmental cues were not changed, hence place cells selective for a particular region in the arena did not remap to be selective for a different location. Both the symbolic and neural schema agents showed above chance visit ratios for the 2NPA (symbolic: $t = 27.2, p < 0.001$, neural: $t = 29.8.2, p < 0.001$) and 6NPA (symbolic: $t = 37.0, p < 0.001$, neural: $t = 52.0, p < 0.001$) condition, demonstrating one-shot learning of two and six new paired associations whereas the actor-critic trained by temporal difference error modulated

Hebbian rule showed chance performance (2NPA: $t = -3.7, p < 0.001$, 6NPA: $t = -4.3, p < 0.001$), like the advantage actor-critic (A2C) agent trained by backpropagation (Kumar et al. 2021).

To study the one-shot learning capacity of the neural agent, 12 new paired associates were introduced for a single trial after learning the OPA configuration over 20 sessions. For each agent simulation, the goal locations for the 12 PAs were randomly chosen out of the 43 remaining reward locations, after excluding the six reward locations that were used for the OPA condition. We arbitrarily defined a PA to have been learned if an agent achieved a visit ratio of more than 16.7% for the pair, well above the 8.3% expected if all 12 goals were visited randomly. The actor-critic learned 1.8 ± 0.08 PAs while the symbolic schema agent learned 10.5 ± 0.1 PA after just one trial of learning. Conversely, a neural agent with a small reservoir of 128 units learned 3.9 ± 0.07 PAs after one trial but the one-shot learning capacity increased monotonically to 8.0 ± 0.1 PAs when the size of the reservoir was increased to 1024. However, the one-shot learning capacity decreased significantly when the size of the reservoir was further increased to 2048. Comparable performance was attained when the learning rate η_{goal} was reduced from 0.000075 to 0.00001 (see supplementary Fig. 3.2 for example trajectories for 12NPA).

Figure 3.5D shows example trajectories during the probe sessions PS1 till OPA, 2NPA and 6NPA. The actor-critic gradually learned to navigate to the correct goals in the OPA condition but failed to navigate to the two or six new PAs. Instead, schema agents navigated to all goals in the OPA, 2NPA and 6NPA conditions by direct heading with stochasticity in the trajectory due to the noise in the actor. If a particular PA was not learned properly, schema agents navigated to the centre of the arena (till a goal was found). Although all agents learned distinct policies for the six PAs during the OPA condition (example map shown for cue 1 in Fig. 3.5E), only the schema agents demonstrated distinct maps to navigate to the new PAs (example map for cue 7 and cue

11 goals shown in Fig. 3.5E). The actor-critic showed similar value and policy maps for all new PAs despite the cue presented, like in Kumar et al. (2022).

3.3.5 One-shot navigation to multiple new paired associates

We have shown that schema agents can perform one-shot learning of multiple new flavour-location paired associates while the actor-critic agent is unable to. We have also demonstrated that schema agents are unable to navigate past obstacles while actor-critic agents and the hybrid actor-critic-schema agents can. Here we study the ability of these agents to navigate past obstacles and perform one-shot learning of multiple new paired associations. The flavour-location configuration for OPA, 2NPA and 6NPA are the same as in Figure 3.5. The arena now has an obstacle that is a 90-degree rotated H configuration. Training was extended to 50 sessions with nonrewarded probe sessions during session 2, 18 and 36. After 50 sessions, agents were introduced to the OPA, 2NPA and 6NPA conditions for a single session followed by a probe. The agent's starting position was constrained based on the goal location (see Table 2 in Methods) so that the agent has to navigate around the obstacle instead of using direct heading to reach the goal.

Figure 3.6A shows that the latency for all agents, except the symbolic agent, gradually decreased (actor-critic: $106\ s \pm 56\ (SD)$, symbolic: $362\ s \pm 94\ (SD)$, neural: $99\ s \pm 47\ (SD)$, actor-critic-symbolic: $110\ s \pm 64\ (SD)$, actor-critic-neural: $108\ s \pm 65\ (SD)$) and the visit ratios increased to above chance performance during the probe sessions PS1 to PS2 to PS3 ($p < 0.001$). This suggests that most agents gradually learned to navigate past the obstacle to reach the correct goal.

Figure 3.6B shows the visit ratios during the probe session after 50 sessions of learning the OPA configuration and one session of learning the OPA, 2NPA and 6NPA configuration. Only the hybrid actor-critic-schema agents demonstrate visit ratios that are above chance performance for the 2NPA (hybrid symbolic: $t = 16.5, p < 0.001$,

hybrid neural: $t = 8.3, p < 0.001$) and 6NPA (hybrid symbolic: $t = 16.9, p < 0.001$, hybrid neural: $t = 3.7, p < 0.001$) configurations, demonstrating one-shot navigation to the two and six new paired associations.

The actor-critic agent learned distinct maps for each cue, allowing it to navigate past obstacles (Fig. 3.6C–D and Supplementary Fig. 3.2). However, it failed to learn an appropriate policy to navigate to the new PAs after a single trial, like in Fig. 3.5E.

The pure symbolic agent $\beta^{control} = 1$ moved towards the goal using direct heading and got stuck at the obstacle to show chance visit ratios during the probe sessions, demonstrating its inability to navigate past obstacles like in Fig. 3.4C. Interestingly, the pure neural agent showed gradual learning performance like the actor-critic. This is because when the neural agent is far away from the goal location, its recall value falls below the threshold (Supplementary Figure 3.1B), causing it to suppress the NAVIGATE schema and instead adopting a random policy to navigate past obstacles. When the pure neural agent moved past the obstacle and closer to the goal, its recall value exceeded the threshold, and used the NAVIGATE schema to head directly to the goal. Although this strategy allowed the pure neural agent to navigate past obstacles and solve multiple goals, it failed to learn and navigate to new paired associates after a single trial.

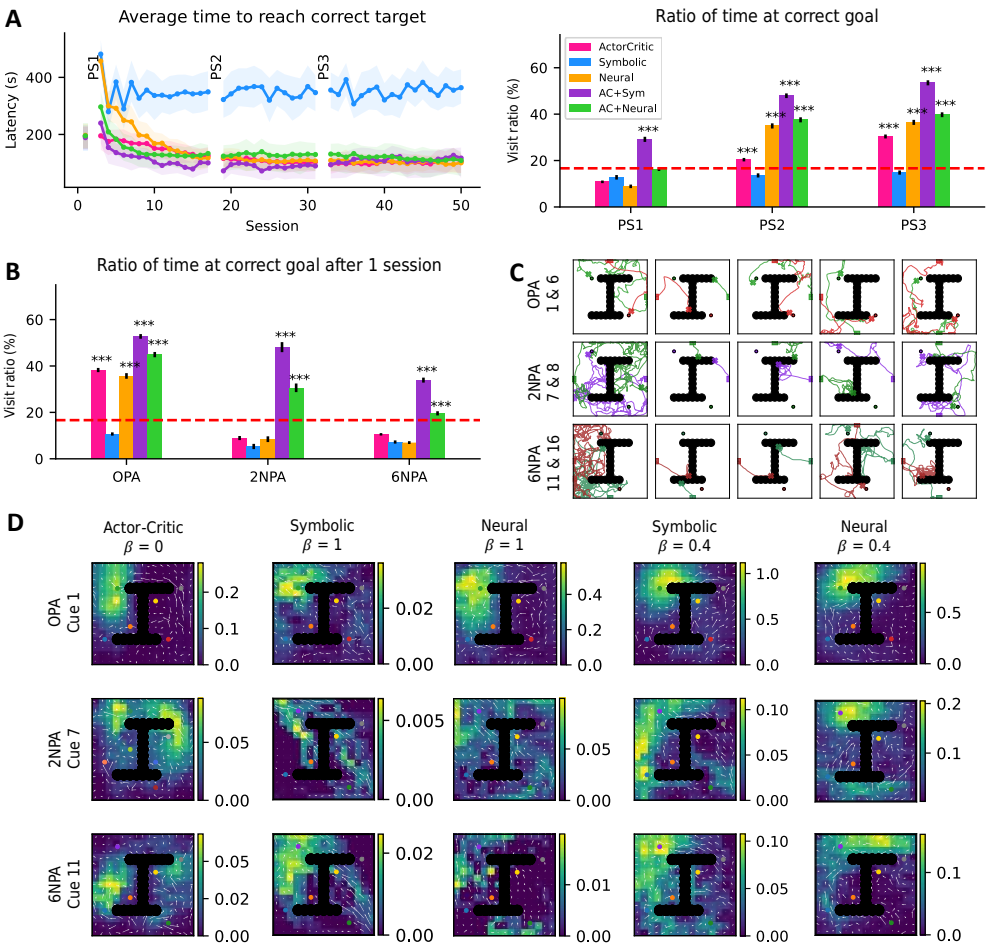


Figure 3.6. One-shot navigation to new paired associates by model-free and schema hybrid agents. A) All agents, except the pure symbolic agent, showed a decrease in average latency to reach six PAs (left) and showed improvement in the visit ratios to the correct goal during the probe sessions (right). B) Both the actor-critic-symbolic and actor-critic-neural agents demonstrated one-shot learning of two (2NPA) and six new PAs (6NPA) ($p < 0.001$) while the actor-critic and pure schema agents showed chance performance of 16.7%. C) Example trajectories to two FLAVOUR-LOCATION pairs by each agent (left to right: Actor-Critic, Symbolic, Neural, Actor-Critic-Symbolic, Actor-Critic-Neural) during the OPA (cue 1 and cue 6), 2NPA (cue 7 and cue 8) and 6NPA (cue 11 and cue 16) probe sessions. Although the Actor-Critic agent can navigate past obstacles, it cannot navigate to the new PAs after a single trial whereas the pure schema agents cannot navigate past obstacles since the NAVIGATE schema only allows for direct heading. Only the hybrid agents use a combination of direct heading and state-based actions to navigate past obstacles and learn new PAs after a single trial. D) Superimposed value (color) and policy (white arrows) maps of all agents during nonrewarded probe sessions (averaged over 24 simulations per agent). Only the hybrid agents show an optimal value and policy maps for new PAs. 196 simulations per agent, shaded area indicates 25th and 75th quantiles while error bars indicate standard error.

Instead, the actor-critic-schema agents gradually learned a mixed policy to navigate away from the obstacles while using the NAVIGATE schema to directly head to the goals when possible (Fig. 3.6C–D). This strategy enabled the hybrid agents to achieve above chance visit ratios for 2NPA and 6NPA configurations in an arena with obstacles (Fig. 3.6B) to demonstrate one-shot navigation.

3.3.6 Learning to gate working memory generalises to new paired associates

In the previous sections, sensory cues were presented throughout the trial for all tasks. In this section, we used the same task as in Figure 3.5 but with the cue presented only at the start of the trial for 2 seconds, to simulate the same task conditions as the biological experiment (Tse et al. 2007). Only the hybrid actor-critic-neural schema agent with $\beta^{control} = 0.9$ was studied for this section.

Kumar et al. (2022) demonstrated that adding a bump attractor to the reservoir-actor-critic agent endowed it with working memory to persistently maintain the transient sensory cue. This enabled the agent to gradually learn the multiple paired association task. Similarly, we added a bump attractor that took in sensory cues as inputs using loading weights such that each cue caused different subpopulations within the bump attractor to persistently maintain activity throughout the trial. The bump attractor activity was passed to the reservoir as an additional input (Fig. 3.7A).

The LEARN FLAVOUR-LOCATION schema associated sensory cues to goal coordinates. We wanted to study if agents could learn to selectively attend to the task relevant cues and ignore distractors to form the correct flavour-location association. To do this, either cue 17 or 18 was randomly presented three seconds after the presentation of task relevant cue, at a mean rate of $0.1\,Hz$ for a duration of one second. The distractor caused a different subpopulation of the bump attractor to be excited. The distractor was presented either once or twice within a trial such that the bump attractor

maintained wrong cue representations that fluctuated throughout the trial (Fig. 3.7B). When the distractor is maintained in working memory, agent will associates the wrong sensory cue with the goal coordinates, hindering its ability to solve the MPA task.

To attend to task relevant cues, we added a gating mechanism that controlled the information flow from the sensory cue to the bump attractor (Fig. 3.7A). The gate performed two actions, either update the working memory by allowing sensory information to flow to the bump attractor or maintain the working memory by restricting information being passed to the bump attractor (Lloyd et al. 2012; O'Reilly and Frank 2006; Todd et al. 2009)

An optimal gating policy is to update the working memory with the task relevant cue and maintain it throughout the trial to ignore distractors (Fig. 3.7C) so that the agent can form the correct FLAVOUR-LOCATION association. Dopamine, modelled here as the temporal difference error $\delta^{DA}(t)$ computed by the critic (Eq. 14), has been shown to encode the presentation of task relevant information (P. Read Montague et al. 1996; Niv 2009; W. Schultz, Dayan, and Montague 1997) and can be used to learn an optimal gating policy by modulating the Hebbian plasticity rule with presynaptic reservoir activity and postsynaptic gating activity (Eq. 45).

With the synaptic plasticity for the gating mechanism switched off and no distractors presented (Fig. 3.7D green), the neural schema agent with the bump attractor and random gating policy showed gradual decrease in latency to $51\,s \pm 43\,(SD)$ and above chance visit ratio performance during the probe sessions PS1 to PS3 and OPA ($p < 0.001$). The agent also achieved above chance visit ratios for the 2NPA ($t = 4.5, p < 0.001$) and 6NPA ($t = 13.5, p < 0.001$) conditions after one session of training (Fig. 3.7E).

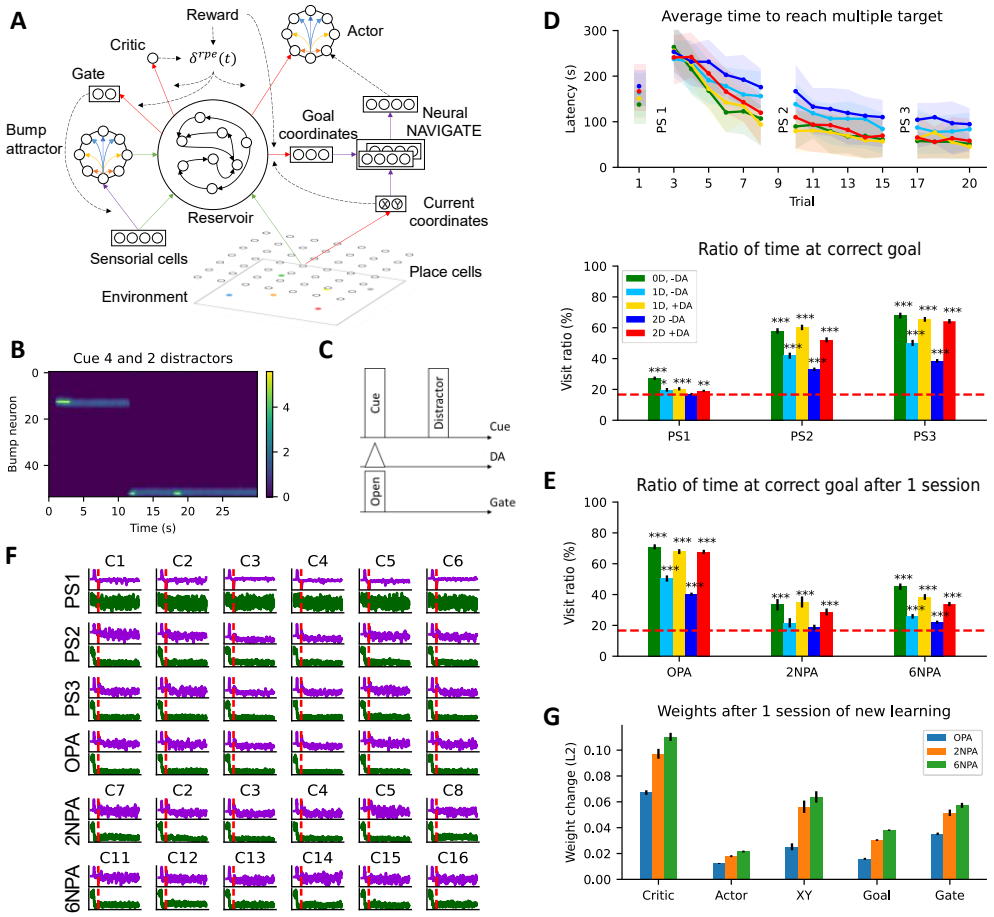


Figure 3.7. Learning to gate working memory from distractors generalises to new paired associates. A) The task relevant sensory cue is given only at the start of the trial. Distractors are given as the agent navigates through the maze. A bump attractor and gating mechanism is added to the neural schema agent. Sensory cues synapse to the bump attractor neurons while the gating mechanism either updates or maintains working memory by opening or closing the inputs to the bump neurons. The synapses from the reservoir to the gate are learned using the temporal difference error modulated Hebbian rule, similar to the learning rule used by the Actor and Critic. B) When cue 4 is presented, a subpopulation of bump attractor neurons selective for cue 4 become persistently activated till distractor cue 17 or 18 is presented. This causes the bump activity to shift to another subpopulation. C) In Tse et al., 2007, the FLAVOUR cue presented at the start of the trial signals the corresponding target LOCATION. Similarly, in this simulation, only the cue presented at the start of the trial is indicative of the goal location while subsequent cues serve as distractors. Reward prediction error signal encodes task relevant information and can be used to learn a gating policy to attend to task relevant cues and ignore distractors. D) When zero (green), one (light blue) or two (dark blue) distractors were presented, agents correspondingly took a longer amount of time to reach the correct goal (top) and spent a significantly lower amount of time at the target during probes PS2 and PS3 (bottom). When the plasticity for the gate was switched on, agents performed as well as when there were no distractors by navigating faster and spending a significantly higher amount of time at the goal compared to when plasticity was switched off ($p < 0.001$). E) Agents with

synaptic plasticity switched on demonstrated higher visit ratios for 2NPA and 6NPA compared to agents without plasticity to learn a gating policy, indicating improved one-shot learning of two and six PAs. F) The temporal difference error (purple) signals the presentation of the cue at the start of the trial while the gating policy (green) learns to open when the task relevant cue is presented (before red dashed line) and remains closed throughout the probe trial to prevent distractors disrupting working memory. The gradually learned gating policy (PS1 to PS2) applied to new cues, even though there were only presented for a single trial, suggesting that the gate learned a generalisable policy to open only at the start of the trial and remain closed subsequently despite the cue presented. Row shows the change in temporal difference error and gating policy as learning progressed while columns show the activity when different cues were presented during probe sessions PS1-PS3, OPA, 2NPA and 6NPA. 144 simulations per agent, shaded area indicates 25th and 75th quantiles while error bars indicate standard error.

However, when one or two distractors (Fig. 3.7D one distractor – light blue, two distractors – dark blue) were presented to the agent, the latency decreased to $84\ s\ \pm 54\ (SD)$ for one distractor and $95\ s\ \pm 63\ (SD)$ for two distractors while visit ratio performance during PS1, PS2, PS3 and OPA decreased significantly compared to when no distractors were presented ($p < 0.001$). Visit ratios for 2NPA decreased to chance performance (one distractor: $t = 1.6, p = 0.0571$, two distractors: $t = 1.2, p = 0.115$) while visit ratios for 6NPA (one distractor: $t = -8.6, p < 0.001$, two distractors: $t = -23.8, p < 0.001$) significantly decreased compared to when no distractors were presented (Fig. 3.7E), demonstrating the ability for distractors to affect the agent's one-shot learning performance.

When synaptic plasticity to the gating mechanism was switched on, agents gradually learned to mitigate the effects of distractors. Agents showed a decrease in latency to $45 \pm 45\ (SD)$ for one distractor and $58 \pm 49\ (SD)$ for two distractors while showing improvements in visit ratios during probe sessions PS1 to PS3 and OPA ($p < 0.001$) compared to when synaptic plasticity was switched off.

More importantly, visit ratios for 2NPA were above chance performance (one distractor: $t = 5.1, p < 0.001$, two distractors: $t = 5.4, p < 0.001$) and visit ratios for

6NPA was significantly higher compared to when synaptic plasticity was switched off (one distractor: $t = 5.3, p < 0.001$, two distractors: $t = 7.0, p < 0.001$). This suggests that the gating policy generalised to new flavour-location PAs (cues 7, 8, 11-16) that the agent had not seen before, demonstrating the gating mechanism's ability to learn a generalizable rule to ignore distractors and associate task relevant cues for one-shot learning of multiple new PAs.

Figure 3.7F shows the temporal difference error (purple), averaged across agents, encoding the presentation of task relevant cues (across columns) which was presented only at the start of the probe trial (red dashed line is when task cue presentation ended and navigation started) and showing little to no selectivity to distractors for the rest of the trial across probe sessions PS1 to PS3 as well as OPA, 2NPA and 6NPA (down the rows), validating the theoretical scheme in Figure 3.7C. The probability of the gating mechanism updating the working memory (value of $\chi(t)$ in Eq. 44 being 1) was initially random (dark green) throughout the probe trial during PS1. As learning progressed, the gating policy updated working memory with a higher probability at the start of the trial when the task relevant cue was presented (before the red dashed line) and maintained a lower probability of update for the rest of the trial period. This suggests that the gating mechanism did learn the optimal gating policy in Figure 3.7C to only allow task relevant cues to be fed into the bump attractor and remain closed thereafter to maintain the working memory information by preventing disruptions by distractors.

Agents showed significant ($F = 74.6, p < 0.001$) and monotonic increase across all synaptic weight change when learning a greater number of novel paired associations compared to the single OPA session. Since we did not simulate a change in environmental cues, the amount of synaptic weight change in a new environment (New Maze condition in Tse et al. (2007)) was not assessed. Nevertheless, the model predicts that when learning new PAs using the LEARN METRIC REPRESENTATION and

LEARN FLAVOUR-LOCATION schemas, there is a greater amount of synaptic weight change, consistent with the memory schema consolidation theory (McClelland 2013) and the increase in immediate early gene levels in the prelimbic region during NPA conditions compared to the OPA condition (Tse et al. 2011).

3.4 Discussion

We have shown that an almost fully neural reinforcement learning agent with three schemas and biologically plausible synaptic plasticity demonstrates one-shot learning of multiple new paired associations. The only caveat to the agent being fully neural is the symbolic specification to associate flavour-location pairs when a positive reward is disbursed, and to forget a previously stored paired associate if the trial ends with no reward. However, once the decision is made to associate or forget, the association or forget operations are neurally implemented.

Model-free reinforcement learning agents such as the actor-critic can navigate past obstacles but do not show one-shot learning. Instead, a combination of actor-critic and schema agents perform one-shot navigation to multiple new pairs. Furthermore, we demonstrate a biologically plausible working memory gating mechanism that gradually learns to attend to task relevant cues and generalises to new pairs of FLAVOUR-LOCATION associations.

We verified that the actor-critic trained using biologically plausible learning rules (Kumar et al. 2022) learns to navigate past obstacles to multiple paired associates but could not learn new PAs after a single trial. Although Foster and colleagues developed an agent that could learn new PAs in one shot by learning a metric representation using biologically plausible synaptic rules, its LEARN GOAL COORDINATE and NAVIGATE schemas were symbolic and one-shot learning was demonstrated only for single goals (Foster et al. 2000). We refined the LEARN METRIC REPRESENTATION schema, and implemented the NAVIGATE schema neurally by

using backpropagation to train a neural network that approximates the computation performed by symbolic version of NAVIGATE. More importantly, we have shown that the LEARN FLAVOUR-LOCATION association schema can be neurally implemented as a reservoir with readout units trained by the reward-modulated Exploratory Hebbian rule to associate multiple sensory cues to goal coordinates after a single trial per paired associate, to replicate the one-shot learning rodent result of Tse et al. (2007).

Behavioural evidence demonstrates animals perform vector-based navigation to goals from any arbitrary location (Etienne et al. 1998; Müller and Wehner 1988). To achieve this, we hypothesize the brain contains three schemas 1) the ability to self-localize by learning a metric representation of the environment 2) the ability to perform one-shot association to store and recall goal coordinates 3) the ability to navigate to the recalled goal location from an arbitrary location using the shortest path. Although the schemas we have proposed are inspired by theoretical accounts, there is experimental evidence suggesting that these schemas could exist in the brain.

Place cells in the hippocampus (Moser et al. 2015; Sosa and Giocomo 2021) and grid cells in the entorhinal cortex (Behrens et al. 2018; McNaughton et al. 2006) have been shown to code an allocentric representation of an animal's position to perform goal directed navigation. However, several proposals suggest that animals use these binned representations as an error correcting mechanism while additional neural circuitry is needed to transform the binned representation into a continuous metric representation of the environment to perform vector-based navigation (Bush et al. 2015; Fiete et al. 2008; Widloski and Fiete 2014). Metric representations may be present in the cortex (Ito et al. 2015; Salinas and Abbott 2001) and other brain regions (Hulse et al. 2021; Yang et al. 2021). The LEARN METRIC REPRESENTATION schema learns a Cartesian metric representation, however, whether the brain learns a Cartesian, polar or other metric representation needs to be experimentally studied (Bush et al. 2015).

One-trial association of goal location is canonically attributed to the hippocampal CA3 auto-associative system due to its highly recurrent architecture implementing attractor dynamics (Pfeiffer and Foster 2015; Rolls 2007, 2013). The Hopfield network has a similar architecture and uses the Hebbian rule (Hopfield 1982; Whittington et al. 2020) to form and recall associations after one trial. However, Hopfield networks suffer from at least two problems. The first problem is that Hopfield networks are not suitable architectures for continual learning because once a set of patterns are stored, additional patterns cannot be stored without disrupting previous associations (Fusi 2021; Parisi et al. 2019). Additional techniques such as pseudo-rehearsals are needed to store new patterns simultaneously with the old patterns (Frean and Robins 1999; Robins 2004). Although replay of previous episodes (Carr, Jadhav, and Frank 2011; Ji and Wilson 2007; Karlsson and Frank 2009) could be a biological mechanism for pseudo-rehearsals, this requires additional neural circuitry to separately store and recall episodic memory (Nicola and Clopath 2019; van de Ven, Siegelmann, and Tolias 2020). The second problem is that additional neural circuitry is needed to transform the Hopfield representations for vector-based navigation. For example, the flavour cue vector and place cell activity at the goal location needs to be transformed into a suitable vector representation to perform auto association. Subsequently, an inverse transformation is needed to retrieve the goal coordinates for vector navigation. Neural circuitry to perform these transformations can be obtained by training neural networks using backpropagation (Banino et al. 2018; Limbacher and Legenstein 2020; Whittington et al. 2020), though the computation performed by the network becomes difficult to interpret. Instead, our proposed reservoir-based LEARN FLAVOUR-LOCATION association schema does not suffer from the continual learning problem nor does it require additional mechanisms to recall the flavour cue-associated goal coordinates.

While goal information is usually thought to be represented in the hippocampus (Hok et al. 2007; Ormond and Keefe 2022; Pfeiffer and Foster 2013; Sarel et al. 2017), the prefrontal cortex recalls goal information too (Hok et al. 2005; Tse et al. 2011). Accordingly, the LEARN FLAVOUR-LOCATION association schema may be jointly implemented in the hippocampus and prefrontal cortex. Other biologically plausible implementations of one-shot association that may comprise CA3's autoassociative architecture may yet to be discovered.

Vector-based navigation has been a dominant proposal for one-shot navigation (Banino et al. 2018; Howard et al. 2014; Ito et al. 2015; Lyu, Abbott, and Maimon 2022). Egocentric-based navigation proposals do not rely on vector computation (Ethier et al. 2001; Fouquet et al. 2013; Rich and Shapiro 2009) but its suitability for one-shot navigation is unclear. Hence, the NAVIGATE schema performs vector subtraction between the animal's current location and goal to compute the distance and direction vector for vector-based navigation.

However, our NAVIGATE schema only affords direct heading. Combining the actor-critic with the NAVIGATE schema was motivated by work that compared model-free and model-based reinforcement learning (Daw et al. 2011; Gläscher et al. 2010). Although this method surprisingly allowed agents to navigate past obstacles and demonstrate one-shot learning of new PAs, a more elegant NAVIGATE schema remains to be developed that allows agents to identify trajectories that navigate past obstacles and towards goals. For example by either using options (Botvinick 2012), subgoals (McGovern and Barto 2001), successor representations (Dayan 1993; Gardner et al. 2018; Stachenfeld, Botvinick, and Gershman 2017) or path planning algorithms (Stentz 1997). Perhaps the readout units of the reservoir can be trained to learn and recall sequential goal coordinates (Cazin et al. 2019) in a reinforcement learning paradigm (Miconi 2017; Murray 2019).

The handcrafted aspects of the network such as outputs that are goal coordinates, and synaptic plasticity gated by a scalar error that is computed using vector subtraction as an intermediate step, must arise via processes during development or prior experience that we do not model. Perhaps a similar solution could have been obtained by training a network via backpropagation (Banino et al. 2018; Cueva and Wei 2018; Whittington et al. 2020). However, the computation performed by the handcrafted structure is more interpretable.

We also implemented a neural network-based solution to gate working memory so that the agent associated task relevant cues with goal coordinates instead of distractor cues. Although the initial proposal was that the basal ganglia gated working memory in the prefrontal cortex (O'Reilly and Frank 2006), there is evidence to suggest that the thalamus also gates relevant information to learn tasks with uncertainty (Mukherjee et al. 2021; Rikhye, Gilra, and Halassa 2018). Exactly which brain regions exert top-down control over working memory has yet to be established.

The plasticity rules used by LEARN METRIC REPRESENTATION and LEARN FLAVOUR–LOCATION schemas are biologically plausible in that they use local information such as the presynaptic activity, postsynaptic activity and global neuromodulatory factors. Learning a metric representation requires an error term to be computed for each axis and presynaptic activity in the form of an eligibility trace encapsulating place cell activity. This takes the form of a non-Hebbian two-factor learning rule in which plasticity depends on presynaptic activity and other factors, but not postsynaptic activity; this resembles learning rules in the cerebellum (Hausknecht et al. 2017; Medina and Mauk 1999; Piochon et al. 2013) and the amygdala (Humeau et al. 2003). Prior work suggests a three-factor Hebbian rule could succeed as well as the non-Hebbian two-factor rule (Frémaux et al. 2013). Plasticity at the goal synapses requires all three factors, and an additional reward modulation factor, expanding on the

commonly described formulation of the neuromodulated Hebbian rule (Frémaux and Gerstner 2016; Hoerzer et al. 2012) into a 4-factor Hebbian rule.

To further advance the biological plausibility of our one-shot learning schema agents, future computational modelling may explore other agent architectures, using a spiking neuronal model, spike time dependent plasticity rules and effects of neuromodulators (Brzosko et al. 2017; Frémaux et al. 2013; Zannone et al. 2018).

While we have proposed anatomical mappings of the schemas to brain regions, we have yet to explore how the circuitry could account for experimental results such as the similarity in neural representation between familiar and novel information (Baraduc et al. 2019), or encode other types of schemas (McKenzie et al. 2014; Zhou et al. 2020). We have also not modelled the memory consolidation (Alvarez and Squire 1994; Kumaran et al. 2016; McClelland 2013) result by Tse et al. (2007) in which the recall of flavour-location association pairs becomes hippocampus-independent.

CHAPTER 4 Conclusion

Animals can solve a novel task after a single attempt by using their prior knowledge. This phenomenon is called one-shot learning. Artificial neural networks, on the other hand, require millions of training examples and parameters while using biologically implausible learning rules to demonstrate generalizable behaviour. Schemas are a tantalizing theory of how animals learn efficiently. However, how schemas are represented in the brain and how they facilitate one-shot learning has remained elusive.

Hence, our research question was, what are the neural circuitry and biological learning rules required to learn new information after a single trial to solve novel problems?

To answer this question, my thesis has been to develop a biologically plausible reinforcement learning agent that replicated the two-part rodent experiment by Tse et al. (2007). The first part was to replicate the gradual learning of multiple flavour cue and goal location paired associates (PA) and the second part was to replicate the learning of multiple new PAs after a single trial.

4.1 Summary of contributions

Chapter 1 reviewed the computational problem of one-shot learning, several symbolic and neural algorithms to solve new variants of a task after a single trial and experimental work characterizing the neural computations involved in one-shot learning. Schemas can facilitate rapid integration of new information to enable one-shot learning. However, there were no demonstrations of neural network-based models that use local synaptic information and global neuromodulatory factors to implement schemas for one-shot learning.

Chapter 2 demonstrated a biologically plausible reinforcement learning (RL) agent that gradually learned multiple flavour cue and goal location PAs as in the first part of Tse et al. (2007). Learning was performed solely by the actor and critic components, which

is similar to stimulus-response learning performed by the dorsolateral and ventral striatum in the basal ganglia (Barto 1995; Houk et al. 1994; Joel et al. 2002; Niv 2009).

The synapses to the actor and critic were updated using the biologically plausible temporal difference error modulated Hebbian rule which used only the presynaptic activity, postsynaptic activity, and the reward prediction error as a global neuromodulatory factor. Although classical RL agents (Foster et al. 2000; Frémaux et al. 2013), with synapses from the place cells and sensory cue to the actor and critic, gradually learned single goals, they could not learn distinct policies to solve multiple PAs. Instead, if place cells and sensory cues were pre-processed by a nonlinear hidden layer or reservoir of recurrently connected units, and synaptic plasticity was between the hidden layer and actor-critic, these agents learned distinct value and policy maps to solve up to 16 PAs (Kumar et al. 2022).

Based on our computational model, place cell and sensory cue information need to be first pre-processed by a downstream cortical layer before passing the information to the striatum, suggesting a hippocampal-cortico-striatal pathway (De Bruin et al. 1997; Kolb et al. 1994; Negrón-Oyarzo et al. 2018; Whitlock et al. 2008) to learn multiple PAs.

Chapter 3 demonstrated that although the biologically plausible reservoir-actor-critic gradually learned multiple PAs, it failed to subsequently demonstrate one-shot learning of new PAs. Similarly, an actor-critic trained using backpropagation could not learn new PAs after a single trial (Kumar et al. 2021). Instead, we proposed the need for three schemas to replicate the second part of Tse et al. (2007).

The first schema was to learn a continuous metric representation of the environment (LEARN METRIC REPRESENTATION). The schema used place cell activity as inputs and synapses to X and Y coordinate cells were updated using a path integration derived temporal difference error modulated Hebbian plasticity rule (Foster et al.

2000). The error was computed by integrating the agent's self-motion cues with path integration estimations. This additional neural circuitry has been suggested to be necessary for animals to self-localize and perform vector navigation, which affords a more efficient goal directed strategy (Bush et al. 2015; Fiete et al. 2008).

The second schema was to learn multiple flavour cue and goal coordinate associations after a single trial (LEARN FLAVOUR-LOCATION). Autoassociative attractor networks require additional complex circuitry to transform goal information into goal directed behaviour (Banino et al. 2018; Limbacher and Legenstein 2020; Whittington et al. 2020) and they do not perform well in tasks that require continual learning unless additional mechanisms are incorporated (Frean and Robins 1999; Robins 2004). Hence, we implemented the one-shot association system using a reservoir that took in flavour cues as inputs and readout units representing goal coordinates as output, with only the readout synapses trained using a reward-modulated Exploratory Hebbian rule (Hoerzer et al. 2012). This biologically plausible system solved the two problems suffered by the autoassociative network. At the same time, specific associations could be stored or deleted like the symbolic key-value matrix used in a neural Turing machine (Graves et al. 2014; Santoro et al. 2016). Although we did not propose an anatomical mapping of this association schema, the micro-cortical inspired reservoir architecture may be implemented by the prefrontal cortex and hippocampus given that the prefrontal cortex is needed to encode and recall flavour associated goal coordinates collaboratively with the hippocampus (Gilboa and Marlatte 2017; Ito et al. 2015; Tse et al. 2011).

The third schema performed direct heading to a defined goal from arbitrary locations. This was by first performing vector subtraction between an agent's current and goal coordinates and subsequently choosing the direction towards the goal (NAVIGATE). Since we only used these two computations to train a network by backpropagation and subsequently fixed the weights during learning, the one-shot learning demonstrated by

this schema agent is biologically plausible. We assumed that development or learning of vector subtraction occurred during development.

We demonstrated that an agent with these three schemas – LEARN METRIC REPRESENTATION, LEARN FLAVOUR-LOCATION, NAVIGATE – gradually learned six PAs and subsequently learned up to eight new PAs after one trial, replicating the one-shot learning behaviour seen in the second part of Tse et al. (2007).

Furthermore, an agent that combined an actor-critic with the schemas navigated past obstacles and demonstrated one-shot learning of new PAs. Lastly, the schema agent used the reward prediction error to gradually learn a working memory gating policy to ignore distractors and generalised to associate new flavour cues to goal coordinates.

Most of the synaptic weights in the neural schema agent were trained using a modulated Hebbian rule, which obeys similar weight update principles as the experimentally demonstrated the spike time-dependent plasticity (STDP) rule (Markram et al. 1997). Furthermore, the rate-coded neuron model and temporal difference errors obey biologically plausible time constants. Hence, the actor-critic-neural schema agent is biologically plausible.

4.2 Limitations and future directions

Despite the ability to demonstrate one-shot learning of paired associations, the biological plausible schema agent has several limitations.

Biological plausibility. As our model uses leaky rate-coded neurons and Hebbian plasticity, a refinement would be to extend the schema agent with spiking neurons and modify synapses using spike time dependent plasticity rules (Frémaux et al. 2013). Other modifications are to impose biological constraints on neuromodulators representing reward prediction errors and path integration errors.

Can there be an agent that does not need to rely on all three schemas to perform one-shot learning? The successor representation is a hybrid reinforcement learning

algorithm that combines the computational efficiency of model-free algorithm and the planning flexibility offered by model-based algorithms to navigate past obstacles (Akam and Walton 2021; Dayan 1993; Gardner et al. 2018). Can agents that employ successor representations learn multiple new PAs after a single trial while navigating past obstacles? If not, what other schemas might be needed?

Generalizability to other cognitive tasks. The robustness of a proposed model depends on its ability to generalize to a variety of tasks. Although our fully neural schema agent can demonstrate one-shot learning, its capability has only been demonstrated on a particular spatial navigation task. The model should be further tested on its ability to learn sequences (Cazin et al. 2019; Han, Doya, and Tani 2019; Zhou et al. 2020), hierarchical associations (McKenzie et al. 2014; Ribas-Fernandes et al. 2011) or other forms of cognitive control tasks such as task switching (Hoerzer et al. 2012; Mante et al. 2013; Miconi 2017) and reversal learning (Harlow 1949; Wang et al. 2018; Zhang et al. 2018). Furthermore, our proposed schema agent could probably be straightforwardly extended to solve one-shot navigation in a three dimensional environment, for example, to control a robotic arm.

Anatomical mapping. In schema dependent tasks, the hippocampus was required to learn paired associates, both gradually and after a single trial (Tse et al. 2007) while the prefrontal cortex was needed to encode and recall PAs after consolidation has occurred (Tse et al. 2011). How do the computations differ between these two regions for one-shot learning? It has been proposed that the prefrontal cortex learns the abstract task schema to guide information encoding in the hippocampus (Bernardi et al. 2020; Miller and Cohen 2001). This translates to consistent hippocampal CA1 and CA3 neural firing activity for familiar and novel tasks that share the same task schema (Baraduc et al. 2019; McKenzie et al. 2013, 2014). Our current model describes the computations and learning processes required for one-shot learning of PAs but does not propose a distinct anatomical mapping to the hippocampus, prefrontal cortex

(Gilboa and Marlatte 2017), and other brain regions such as the entorhinal cortex (Whittington et al. 2020). This might make it harder to test hypotheses on the computations performed by distinct brain regions, but there is also insufficient experimental evidence about the computations performed by specific brain regions. For example, visual integration happens in the hindbrain systems (Grill and Hayes 2012; Yang et al. 2021) and path integration in the entorhinal cortex (Fuhs and Touretzky 2006; Hafting et al. 2005). Hence, neural integration computations or those required for goal directed vector-based navigation can be found in other brain regions. More work, likely including additional experimental data, is needed to understand the potential links between various theoretical proposals, such as those presented in this chapter, and where they might be implemented in the brain.

Memory consolidation. Tse et al. (2007) demonstrated that the memories of the FLAVOUR-LOCATION PAs were plausibly transferred from the hippocampus to the prefrontal cortex after 24 hours through a memory consolidation process. The complementary learning systems theory postulates that individual episodic memories are gradually consolidated and transferred to the cortex in the form of an abstract task schema (Kumaran et al. 2016; McClelland 2013). A recent computational model proposed a hippocampus to prefrontal cortex consolidation mechanism while replicating the one-shot learning (Hwu and Krichmar 2020) as in Tse et al. (2007), although they did not replicate the experimental conditions and employed supervised learning algorithms. Nevertheless, our current reinforcement learning agent should be expanded to include a more biologically plausible memory consolidation process (Alvarez and Squire 1994; Tomé, Sadeh, and Clopath 2022) to test hypotheses such as the complementary learning system theory in acquiring schemas.

This thesis has presented an example of how the use of theoretical schemas for one-shot learning can be implemented using biologically plausible neural networks and learning rules. These biologically plausible schemas replicate both the gradual and one-

shot learning behaviour exhibited by rodents in the multiple paired associations task by Tse et al. (2007). The next steps are to further the biological plausibility and explain the neural representations seen in the schema literature. I believe more effort to develop biologically plausible models will enable us to predict the learning processes in biological neural circuits to improve education outcomes, alleviate learning disabilities, and may improve artificial intelligence learning algorithms.

Bibliography

Akam, Thomas, Ines Rodrigues-Vaz, Ivo Marcelo, Xiangyu Zhang, Michael Pereira, Rodrigo Freire Oliveira, Peter Dayan, and Rui M. Costa. 2021. "The Anterior Cingulate Cortex Predicts Future States to Mediate Model-Based Action Selection." *Neuron* 109(1):149-163.e7.

Akam, Thomas, and Mark E. Walton. 2021. "What Is Dopamine Doing in Model-Based Reinforcement Learning?" *Current Opinion in Behavioral Sciences* 38:74–82.

Albus, S. 1971. "A Theory of Cerebellar Function." *Mathematical Biosciences* 10:25–61.

Alvarez, Pablo, and Larry R. Squire. 1994. "Memory Consolidation and the Medial Temporal Lobe: A Simple Network Model." *Proceedings of the National Academy of Sciences of the United States of America* 91(15):7041–45.

An, Guozhong. 1996. "The Effects of Adding Noise during Backpropagation Training on a Generalization Performance." *Neural Computation* 8(3):643--674.

Anderson, John R. 2013. *The Architecture of Cognition*. Psychology Press.

Anderson, Richard C., and James W. Pichert. 1978. "Recall of Previously Unrecallable Information Following a Shift in Perspective." *Journal of Verbal Learning and Verbal Behavior* 17(1):1–12.

Arkin, Ronald C. 1989. "Motor Schema — Based Mobile Robot Navigation." *The International Journal of Robotics Research* 8(4):92–112.

Arleo, Angelo, and Wulfram Gerstner. 2000. "Spatial Cognition and Neuro-Mimetic Navigation: A Model of Hippocampal Place Cell Activity." *Biological Cybernetics* 83(3):287–99.

Asabuki, Toshitake, Naoki Hiratani, and Tomoki Fukai. 2018. "Interactive Reservoir Computing for Chunking Information Streams." *PLoS Computational Biology* 14(10):1–21.

Baddeley, Alan. 2012. "Working Memory: Theories, Models, and Controversies." *Annual Review of Psychology* 63(1):1–29.

Bakin, Jonathan S., and Norman M. Weinberger. 1996. "Induction of a Physiological Memory in the Cerebral Cortex by Stimulation of the Nucleus Basalis." *Proceedings of the National Academy of Sciences of the United States of America* 93(20):11219–24.

Balleine, Bernard W., and Anthony Dickinson. 1998. "Goal-Directed Instrumental Action: Contingency and Incentive Learning and Their Cortical Substrates." *Neuropharmacology* 37(4–5):407–19.

Banerjee, Abhishek, Giuseppe Parente, Jasper Teutsch, Christopher Lewis, Fabian F. Voigt, and Fritjof Helmchen. 2020. "Value-Guided Remapping of Sensory Cortex by Lateral Orbitofrontal Cortex." *Nature* 585(7824):245–50.

Banino, Andrea, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, Martin J. Chadwick, Thomas Degris, Joseph Modayil, Greg Wayne, Hubert Soyer, Fabio Viola, Brian Zhang, Ross Goroshin, Neil Rabinowitz, Razvan Pascanu, Charlie Beattie, Stig Petersen, Amir Sadik, Stephen Gaffney, Helen King, Koray Kavukcuoglu, Demis Hassabis, Raia Hadsell, and Dharshan Kumaran. 2018. "Vector-Based Navigation Using Grid-like Representations in Artificial Agents." *Nature* 557(7705):429–33.

Baraduc, P., J. R. Duhamel, and S. Wirth. 2019. "Schema Cells in the Macaque Hippocampus." *Science* 363(6427):635–39.

Barak, Omri, Mattia Rigotti, and Stefano Fusi. 2013. "The Sparseness of Mixed Selectivity Neurons Controls the Generalization-Discrimination Trade-Off." *Journal of Neuroscience* 33(9):3844–56.

Baram, Alon Boaz, Timothy Howard Muller, Hamed Nili, Mona Maria Garvert, and Timothy Edward John Behrens. 2021. "Entorhinal and Ventromedial Prefrontal Cortices Abstract and Generalize the Structure of Reinforcement Learning Problems." *Neuron* 109(4):713-723.e7.

Baras, Dorit, and Ron Meir. 2007. "Reinforcement Learning, Spike-Time-Dependent Plasticity, and the BCM Rule." *Neural Computation* 19(8):2245–79.

Barnes, C. A. 1979. "Memory Deficits Associated with Senescence: A Neurophysiological and Behavioral Study in the Rat." *Journal of Comparative and Physiological Psychology* 93(1):74–104.

Bartlett, F. C., and Cyril Burt. 1932. "Remembering: A Study in Experimental and Social Psychology." *British Journal of Educational Psychology* 3(2):187–92.

Barto, Andrew G. 1995. "Adaptive Critics and the Basal Ganglia." in *Models of Information Processing in the Basal Ganglia*. The MIT Press.

Barto, Andrew G., Richard S. Sutton, and Charles W. Anderson. 1983. "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems." *IEEE Transactions on Systems, Man, and Cybernetics* SMC-13(5):834–46.

Bayer, Hannah M., and Paul W. Glimcher. 2005. "Midbrain Dopamine Neurons Encode a Quantitative Reward Prediction Error Signal." *Neuron* 47(1):129–41.

Behrens, Timothy E. J., Timothy H. Muller, James C. R. Whittington, Shirley Mark, Alon B. Baram, Kimberly L. Stachenfeld, and Zeb Kurth-nelson. 2018. "Perspective What Is a Cognitive Map ? Organizing Knowledge for Flexible Behavior." *Neuron* 100(2):490–509.

Bellec, Guillaume, Franz Scherr, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. 2019. "Biologically Inspired Alternatives to Backpropagation through Time for Learning in Recurrent Neural Nets." 1–37.

Bellec, Guillaume, Franz Scherr, Anand Subramoney, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. 2020. "A Solution to the Learning Dilemma for Recurrent Networks of Spiking Neurons." *Nature Communications* 11(1):1–15.

Bellman, Richard. 1954. "The Theory of Dynamic Programming." *Bulletin of the*

American Mathematical Society 60(6):503–15.

Bernardi, Silvia, Marcus K. Benna, Mattia Rigotti, Jérôme Munuera, Stefano Fusi, and C. Daniel Salzman. 2020. "The Geometry of Abstraction in the Hippocampus and Prefrontal Cortex." *Cell* 954–67.

Bertsekas, Dimitri P., and John N. Tsitsiklis. 1996. *Neuro-Dynamic Programming*. Belmont, MA.

Bethus, Ingrid, Dorothy Tse, and Richard G. M. Morris. 2010. "Dopamine and Memory: Modulation of the Persistence of Memory for Novel Hippocampal NMDA Receptor-Dependent Paired Associates." *Journal of Neuroscience* 30(5):1610–18.

Borst, Jelmer P., Menno Nijboer, Niels A. Taatgen, Hedderik Van Rijn, and John R. Anderson. 2015. "Using Data-Driven Model-Brain Mappings to Constrain Formal Models of Cognition." *PLoS ONE* 10(3):5–7.

Botvinick, Mathew, Sam Ritter, Jane X. Wang, Zeb Kurth-Nelson, Charles Blundell, and Demis Hassabis. 2019. "Reinforcement Learning, Fast and Slow." *Trends in Cognitive Sciences* 23(5):408–22.

Botvinick, Matthew Michael. 2012. "Hierarchical Reinforcement Learning and Decision Making." *Current Opinion in Neurobiology* 22(6):956–62.

Botvinick, Matthew, Leigh E. Nystrom, Kate Fissell, Cameron S. Carter, and Jonathan D. Cohen. 1999. "Conflict Monitoring versus Selection For-Action in Anterior Cingulate Cortex." *Nature* 402(6758):179–81.

Botvinick, Matthew, Jane X. Wang, Will Dabney, Kevin J. Miller, and Zeb Kurth-Nelson. 2020. "Deep Reinforcement Learning and Its Neuroscientific Implications." *Neuron* 107(4):603–16.

Brea, Johanni, Walter Senn, and Jean Pascal Pfister. 2013. "Matching Recall and Storage in Sequence Learning with Spiking Neural Networks." *Journal of Neuroscience* 33(23):9565–75.

Brewer, William F., and James C. Treyens. 1981. "Role of Schemata in Memory for Places." *Cognitive Psychology* 13(2):207–30.

Brown, Michael A., and Patricia E. Sharp. 1995. "Simulation of Spatial Learning in the Morris Water Maze by a Neural Network Model of the Hippocampal Formation and Nucleus Accumbens." *Hippocampus* 5(3):171–88.

De Bruin, J. P. C., W. A. M. Swinkels, and J. M. De Brabander. 1997. "Response Learning of Rats in a Morris Water Maze: Involvement of the Medial Prefrontal Cortex." *Behavioural Brain Research* 85(1):47–55.

Brun, Vegard H., Mona K. Otnæss, Sturla Molden, Hill Aina Steffenach, Menno P. Witter, May Britt Moser, and Edvard I. Moser. 2002. "Place Cells and Place Recognition Maintained by Direct Entorhinal-Hippocampal Circuitry." *Science* 296(5576):2243–46.

Brzosko, Zuzanna, Susanna B. Mierau, and Ole Paulsen. 2019. "Neuromodulation of Spike-Timing-Dependent Plasticity: Past, Present, and Future." *Neuron*

103(4):563–81.

Brzosko, Zuzanna, Wolfram Schultz, and Ole Paulsen. 2015. “Retroactive Modulation of Spike Timing-Dependent Plasticity by Dopamine.” *ELife* 4(4):1–13.

Brzosko, Zuzanna, Sara Zannone, Wolfram Schultz, Claudia Clopath, and Ole Paulsen. 2017. “Sequential Neuromodulation of Hebbian Plasticity Offers Mechanism for Effective Reward-Based Navigation.” *ELife* 6:1–18.

Buonomano, Dean V., and Wolfgang Maass. 2009. “State-Dependent Computations: Spatiotemporal Processing in Cortical Networks.” *Nature Reviews Neuroscience* 10(2):113–25.

Burak, Yoram, and Ila R. Fiete. 2009. “Accurate Path Integration in Continuous Attractor Network Models of Grid Cells.” *PLoS Computational Biology* 5(2).

Bush, Daniel, Caswell Barry, Daniel Manson, and Neil Burgess. 2015. “Using Grid Cells for Navigation.” *Neuron* 87(3):507–20.

Bush, George, Phan Luu, and Michael I. Posner. 2000. “Cognitive and Emotional Influences in Anterior Cingulate Cortex.” *Trends in Cognitive Sciences* 4(6):215–22.

Van Buuren, Mariët, Marijn C. W. Kroes, Isabella C. Wagner, Lisa Genzel, Richard G. M. Morris, and Guillén Fernández. 2014. “Initial Investigation of the Effects of an Experimentally Learned Schema on Spatial Associative Memory in Humans.” *Journal of Neuroscience* 34(50):16662–70.

Caporale, Natalia, and Yang Dan. 2008. “Spike Timing–Dependent Plasticity: A Hebbian Learning Rule.” *Annual Review of Neuroscience* 31(1):25–46.

Carr, Margaret F., Shantanu P. Jadhav, and Loren M. Frank. 2011. “Hippocampal Replay in the Awake State: A Potential Substrate for Memory Consolidation and Retrieval.” *Nature Neuroscience* 14(2):147–53.

Carrell, Patricia L., and Joan C. Eisterhold. 1983. “Schema Theory and ESL Reading Pedagogy.” *TESOL Quarterly* 17(4):553.

Carter, Cameron S., Todd S. Braver, Deanna M. Barch, Matthew M. Botvinick, Douglas Noll, and Jonathan D. Cohen. 1998. “Anterior Cingulate Cortex, Error Detection, and the Online Monitoring of Performance.” *Science* 280(5364):747–49.

Cayco-Gajic, N. Alex, Claudia Clopath, and R. Angus Silver. 2017. “Sparse Synaptic Connectivity Is Required for Decorrelation and Pattern Separation in Feedforward Networks.” *Nature Communications* 8(1):1–11.

Cayco-Gajic, N. Alex, and R. Angus Silver. 2019. “Re-Evaluating Circuit Mechanisms Underlying Pattern Separation.” *Neuron* 101(4):584–602.

Cazin, Nicolas, Martin Llofriu Alonso, Pablo Scleidorovich Chiodi, Tatiana Pelc, Bruce Harland, Alfredo Weitzenfeld, Jean-Marc Fellous, and Peter Ford Dominey. 2019. “Reservoir Computing Model of Prefrontal Cortex Creates Novel Combinations of Previous Navigation Sequences from Hippocampal Place-Cell Replay with Spatial Reward Propagation” edited by C. Inman. *PLOS*

Computational Biology 15(7):e1006624.

Clevert, Djork Arné, Thomas Unterthiner, and Sepp Hochreiter. 2016. "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)." Pp. 1–14 in *4th International Conference on Learning Representations, ICLR 2016*. San Juan, Puerto Rico.

Cueva, Christopher J., and Xue Xin Wei. 2018. "Emergence of Grid-like Representations by Training Recurrent Neural Networks to Perform Spatial Localization." Pp. 1–19 in *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*.

Curtis, Clayton E., and Mark D'Esposito. 2003. "Persistent Activity in the Prefrontal Cortex during Working Memory." *Trends in Cognitive Sciences* 7(9):415–23.

D'amour, James A., and Robert C. Froemke. 2015. "Inhibitory and Excitatory Spike-Timing-Dependent Plasticity in the Auditory Cortex." *Neuron* 86(2):514–28.

Dabney, Will, Zeb Kurth-Nelson, Naoshige Uchida, Clara Kwon Starkweather, Demis Hassabis, Rémi Munos, and Matthew Botvinick. 2020. "A Distributional Code for Value in Dopamine-Based Reinforcement Learning." *Nature* 577(7792):671–75.

Darley, John M., and Paget H. Gross. 1983. "A Hypothesis-Confirming Bias in Labeling Effects." *Journal of Personality and Social Psychology* 44(1):20–33.

Daw, Nathaniel D., Samuel J. Gershman, Ben Seymour, Peter Dayan, and Raymond J. Dolan. 2011. "Model-Based Influences on Humans' Choices and Striatal Prediction Errors." *Neuron* 69(6):1204–15.

Day, M., R. Langston, and R. G. M. Morris. 2003. "Glutamate-Receptor-Mediated Encoding and Retrieval of Paired-Associate Learning." *Nature* 424(6945):205–9.

Dayan, Peter. 1993. "Improving Generalization for Temporal Difference Learning: The Successor Representation." *Neural Computation* 5(4):613–24.

Dennis, Siobhan H., Francesca Pasqui, Ellen M. Colvin, Helen Sanger, Adrian J. Mogg, Christian C. Felder, Lisa M. Broad, Steve M. Fitzjohn, John T. R. Isaac, and Jack R. Mellor. 2016. "Activation of Muscarinic M1 Acetylcholine Receptors Induces Long-Term Potentiation in the Hippocampus." *Cerebral Cortex* 26(1):414–26.

Devinsky, Orrin, Martha J. Morrell, and Brent A. Vogt. 1995. "Contributions of Anterior Cingulate Cortex to Behaviour." *Brain* 118(1):279–306.

DiMaggio, Paul. 1997. "Culture and Cognition." *Annual Review of Sociology* 23(1):263–87.

Dolan, Ray J., and Peter Dayan. 2013. "Goals and Habits in the Brain." *Neuron* 80(2):312–25.

Doya, Kenji. 2000. "Reinforcement Learning in Continuous Time and Space." *Neural Computation* 12(1):219–45.

Dragoi, George, and György Buzsáki. 2006. "Temporal Encoding of Place Sequences

by Hippocampal Cell Assemblies." *Neuron* 50(1):145–57.

Dragoi, George, and Susumu Tonegawa. 2011. "Preplay of Future Place Cell Sequences by Hippocampal Cellular Assemblies." *Nature* 469(7330):397–401.

Ehrlich, Daniel B., and John D. Murray. 2021. "Geometry of Neural Computation Unifies Working Memory and Planning." *BioRxiv* 2021.02.01.429156.

Eichenbaum, Howard. 2004. "Hippocampus: Cognitive Processes and Neural Representations That Underlie Declarative Memory." *Neuron* 44(1):109–20.

Enel, Pierre, Emmanuel Procyk, René Quilodran, and Peter Ford Dominey. 2016. "Reservoir Computing Properties of Neural Dynamics in Prefrontal Cortex." *PLoS Computational Biology* 12(6):1–35.

Engineer, Crystal T., Seth A. Hays, and Michael P. Kilgard. 2017. "Vagus Nerve Stimulation as a Potential Adjuvant to Behavioral Therapy for Autism and Other Neurodevelopmental Disorders." *Journal of Neurodevelopmental Disorders* 9(1):1–8.

Ethier, Katia, Nathalie Le Marec, Pierre Paul Rompré, and Roger Godbout. 2001. "Spatial Strategy Elaboration in Egocentric and Allocentric Tasks Following Medial Prefrontal Cortex Lesions in the Rat." *Brain and Cognition* 46(1–2):134–35.

Etienne, A. S., R. Maurer, J. Berlie, B. Reverdin, T. Rowe, J. Georgakopoulos, and V. Séguinot. 1998. "Navigation through Vector Addition." *Nature* 396(6707):161–64.

Everitt, Barry J., and Trevor W. Robbins. 2016. "Drug Addiction: Updating Actions to Habits to Compulsions Ten Years On." *Annual Review of Psychology* 67:23–50.

Farries, Michael A., and Adrienne L. Fairhall. 2007. "Reinforcement Learning with Modulated Spike Timing-Dependent Synaptic Plasticity." *Journal of Neurophysiology* 98(6):3648–65.

Fiete, Ila R., Yoram Burak, and Ted Brookings. 2008. "What Grid Cells Convey about Rat Location." 28(27):6858–71.

Fiete, Ila R., and H. Sebastian Seung. 2006. "Gradient Learning in Spiking Neural Networks by Dynamic Perturbation of Conductances." *Physical Review Letters* 97(4):048104.

Foster, D. J., R. G. Morris, and Peter Dayan. 2000. "A Model of Hippocampally Dependent Navigation, Using the Temporal Difference Learning Rule." *Hippocampus* 10(1):1–16.

Fouquet, Céline, Bénédicte M. Babayan, Aurélie Watilliaux, Bruno Bontempi, Christine Tobin, and Laure Rondi-Reig. 2013. "Complementary Roles of the Hippocampus and the Dorsomedial Striatum during Spatial and Sequence-Based Navigation Behavior." *PLoS ONE* 8(6).

Frean, Marcus, and Anthony Robins. 1999. "Catastrophic Forgetting in Simple Networks: An Analysis of the Pseudorehearsal Solution." *Network: Computation in Neural Systems* 10(3):227–36.

Frémaux, Nicolas, and Wulfram Gerstner. 2016. “Neuromodulated Spike-Timing-Dependent Plasticity, and Theory of Three-Factor Learning Rules.” *Frontiers in Neural Circuits* 9:85.

Frémaux, Nicolas, Henning Sprekeler, and Wulfram Gerstner. 2013. “Reinforcement Learning Using a Continuous Time Actor-Critic Framework with Spiking Neurons.” *PLoS Computational Biology* 9(4).

Froemke, Robert C., Michael M. Merzenich, and Christoph E. Schreiner. 2007. “A Synaptic Memory Trace for Cortical Receptive Field Plasticity.” *Nature* 450(7168):425–29.

Fu, Justin, Sergey Levine, and Pieter Abbeel. 2016. “One-Shot Learning of Manipulation Skills with Online Dynamics Adaptation and Neural Network Priors.” *IEEE International Conference on Intelligent Robots and Systems* 2016-Novem:4019–26.

Fu, Wai Tat, and John R. Anderson. 2006. “From Recurrent Choice to Skill Learning: A Reinforcement-Learning Model.” *Journal of Experimental Psychology: General* 135(2):184–206.

Fuhs, Mark C., and David S. Touretzky. 2006. “A Spin Glass Model of Path Integration in Rat Medial Entorhinal Cortex.” *Journal of Neuroscience* 26(16):4266–76.

Fusi, Stefano. 2021. “Memory Capacity of Neural Network Models.” *ArXiv Preprint ArXiv:2108.07839* 1–29.

Fusi, Stefano, Earl K. Miller, and Mattia Rigotti. 2016. “Why Neurons Mix: High Dimensionality for Higher Cognition.” *Current Opinion in Neurobiology* 37:66–74.

Fyhn, Marianne, Torkel Hafting, Alessandro Treves, May Britt Moser, and Edvard I. Moser. 2007. “Hippocampal Remapping and Grid Realignment in Entorhinal Cortex.” *Nature* 446(7132):190–94.

Gardner, Matthew P. H., Geoffrey Schoenbaum, and Samuel J. Gershman. 2018. “Rethinking Dopamine as Generalized Prediction Error.” *Proceedings of the Royal Society B: Biological Sciences* 285(1891).

Gershman, Samuel J. 2018. “The Successor Representation: Its Computational Logic and Neural Substrates.” *Journal of Neuroscience* 38(33):7193–7200.

Gilboa, Asaf, and Hannah Marlatte. 2017. “Neurobiology of Schemas and Schema-Mediated Memory.” *Trends in Cognitive Sciences* 21(8):618–31.

Giocomo, Lisa M., May Britt Moser, and Edvard I. Moser. 2011. “Computational Models of Grid Cells.” *Neuron* 71(4):589–603.

Gläscher, Jan, Nathaniel Daw, Peter Dayan, and John P. O’Doherty. 2010. “States versus Rewards: Dissociable Neural Prediction Error Signals Underlying Model-Based and Model-Free Reinforcement Learning.” *Neuron* 66(4):585–95.

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio. 2011. “Deep Sparse Rectifier Neural Networks.” Pp. 315–23 in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Vol. 15. Fort Lauderdale, FL:

PMLR.

Graesser, Arthur C., and Glenn V. Nakamura. 1982. *The Impact of a Schema on Comprehension and Memory*. Vol. 16.

Graves, Alex, Greg Wayne, and Ivo Danihelka. 2014. "Neural Turing Machines." *ArXiv Preprint ArXiv:1410.5401* 1–26.

Graybiel, Ann M. 2008. "Habits, Rituals, and the Evaluative Brain." *Annual Review of Neuroscience* 31:359–87.

Grill, Harvey J., and Matthew R. Hayes. 2012. "Hindbrain Neurons as an Essential Hub in the Neuroanatomically Distributed Control of Energy Balance." *Cell Metabolism* 16(3):296–309.

Grondman, Ivo, Lucian Busoniu, Gabriel A. D. Lopes, and Robert Babuška. 2012. "A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients." *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews* 42(6):1291–1307.

Gu, Hongjing, Qi Chen, Guan Yang, Lei He, Hang Fan, Yong Qiang Deng, Yanxiao Wang, Yue Teng, Zhongpeng Zhao, Yujun Cui, Yuchang Li, Xiao Feng Li, Jiangfan Li, Na Na Zhang, Xiaolan Yang, Shaolong Chen, Yan Guo, Guangyu Zhao, Xiliang Wang, De Yan Luo, Hui Wang, Xiao Yang, Yan Li, Gencheng Han, Yuxian He, Xiaojun Zhou, Shusheng Geng, Xiaoli Sheng, Shibo Jiang, Shihui Sun, Cheng Feng Qin, and Yusen Zhou. 2020. "Adaptation of SARS-CoV-2 in BALB/c Mice for Testing Vaccine Efficacy." *Science* 369(6511):1603–7.

Gulli, Roberto Adamo, Lyndon Duong, Benjamin Whelehan Corrigan, Guillaume Doucet, Sylvain Williams, Stefano Fusi, and Julio Cesar Martinez-Trujillo. 2020. "Context-Dependent Representations of Objects and Space in the Primate Hippocampus during Virtual Navigation." *Nature Neuroscience* 23(January):295774.

Guzman, Segundo Jose, Alois Schlögl, Michael Frotscher, and Peter Jonas. 2016. "Synaptic Mechanisms of Pattern Completion in the Hippocampal CA3 Network." *Science* 353(6304):1117–23.

Hafting, Torkel, Marianne Fyhn, Sturla Molden, May Britt Moser, and Edvard I. Moser. 2005. "Microstructure of a Spatial Map in the Entorhinal Cortex." *Nature* 436(7052):801–6.

Han, Dongqi, Kenji Doya, and Jun Tani. 2019. "Self-Organization of Action Hierarchy and Compositionality by Reinforcement Learning with Recurrent Neural Networks." *Neural Networks* 129:149–62.

Hangya, Balázs, Sachin P. Ranade, Maja Lorenc, and Adam Kepecs. 2015. "Central Cholinergic Neurons Are Rapidly Recruited by Reinforcement Feedback." *Cell* 162(5):1155–68.

Harlow, Harry F. 1949. "The Formation of Learning Sets." *Psychological Review* 56(1):51–65.

Hassabis, Demis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. 2017. "Neuroscience-Inspired Artificial Intelligence." *Neuron*

95(2):245–58.

Hausknecht, Matthew, Wen Ke Li, Michael Mauk, and Peter Stone. 2017. “Machine Learning Capabilities of a Simulated Cerebellum.” *IEEE Transactions on Neural Networks and Learning Systems* 28(3):510–22.

He, Kaiwen, Marco Huertas, Su Z. Hong, Xiao Xiu Tie, Johannes W. Hell, Harel Shouval, and Alfredo Kirkwood. 2015. “Distinct Eligibility Traces for LTP and LTD in Cortical Synapses.” *Neuron* 88(3):528–38.

Healy, S. D., and T. A. Hurly. 1995. “Spatial Memory in Rufous Hummingbirds (Selasphorus Rufus): A Field Test.” *Animal Learning & Behavior* 23(1):63–68.

Hoerzer, Gregor M., Robert Legenstein, and Wolfgang Maass. 2012. “Emergence of Complex Computational Structures from Chaotic Neural Networks through Reward-Modulated Hebbian Learning.” *Cerebral Cortex* 24(3):677–90.

Hok, V., E. Save, and B. Poucet. 2005. “Coding for Spatial Goals in the Prelimbic-Infralimbic.” *Proceedings of the National Academy of Sciences (PNAS)* 102(12):4602–7.

Hok, Vincent, P. P. Lenck-Santini, Sébastien Roux, Etienne Save, Robert U. Muller, and Bruno Poucet. 2007. “Goal-Related Activity in Hippocampal Place Cells.” *Journal of Neuroscience* 27(3):472–82.

Hopfield, J. J. 1982. “Neural Networks and Physical Systems with Emergent Collective Computational Abilities.” *Proceedings of the National Academy of Sciences of the United States of America* 79(8):2554–58.

Hopfield, J. J. 1984. “Neurons with Graded Response Have Collective Computational Properties like Those of Two-State Neurons.” *Proceedings of the National Academy of Sciences of the United States of America* 81(10 I):3088–92.

Hoshi, Eiji, and Jun Tanji. 2004. “Area-Selective Neuronal Activity in the Dorsolateral Prefrontal Cortex for Information Retrieval and Action Planning.” *Journal of Neurophysiology* 91(6):2707–22.

Houk, James C., James L. Adams, and Andrew G. Barto. 1994. “A Model of How the Basal Ganglia Generate and Use Neural Signals That Predict Reinforcement.” Pp. 249–79 in *Models of Information Processing in the Basal Ganglia*. Cambridge, MA: The MIT Press.

Howard, Lorelei R., Amir Homayoun Javadi, Yichao Yu, Ravi D. Mill, Laura C. Morrison, Rebecca Knight, Michelle M. Loftus, Laura Staskute, and Hugo J. Spiers. 2014. “The Hippocampus and Entorhinal Cortex Encode the Path and Euclidean Distances to Goals during Navigation.” *Current Biology* 24(12):1331–40.

Høydal, Øyvind Arne, Emilie Ranheim Skytøen, Sebastian Ola Andersson, May Britt Moser, and Edvard I. Moser. 2019. “Object-Vector Coding in the Medial Entorhinal Cortex.” *Nature* 568(7752):400–404.

Hulse, Brad K., Hannah Haberkern, Romain Franconville, Daniel B. Turner-Evans, Shin Ya Takemura, Tanya Wolff, Marcella Noorman, Marisa Dreher, Chuntao Dan, Ruchi Parekh, Ann M. Hermundstad, Gerald M. Rubin, and Vivek

Jayaraman. 2021. “A Connectome of the Drosophila Central Complex Reveals Network Motifs Suitable for Flexible Navigation and Context-Dependent Action Selection.” *ELife* 10:1–180.

Humeau, Yann, Hamdy Shaban, Stephanie Bissière, and Andreas Lüthi. 2003. “Presynaptic Induction of Heterosynaptic Associative Plasticity in the Mammalian Brain.” *Nature* 426(6968):841–45.

Hunsberger, Eric, Jeff Orchard, and Alexander Wong. 2017. “Spiking Deep Neural Networks : Engineered and Biological Approaches to Object Recognition By.”

Hunt, L. T., N. D. Daw, P. Kaanders, M. A. MacIver, U. Mugan, E. Procyk, A. D. Redish, E. Russo, J. Scholl, K. Stachenfeld, C. R. E. Wilson, and N. Kolling. 2021. “Formalizing Planning and Information Search in Naturalistic Decision-Making.” *Nature Neuroscience* 24(August).

Hwu, Tiffany, and Jeffrey L. Krichmar. 2020. “A Neural Model of Schemas and Memory Encoding.” *Biological Cybernetics* 114(2):169–86.

Ito, Hiroshi T., Sheng Jia Zhang, Menno P. Witter, Edvard I. Moser, and May Britt Moser. 2015. “A Prefrontal-Thalamo-Hippocampal Circuit for Goal-Directed Spatial Navigation.” *Nature* 522(7554):50–55.

Izhikevich, Eugene M. 2007. “Solving the Distal Reward Problem through Linkage of STDP and Dopamine Signaling.” *Cerebral Cortex* 17(10):2443–52.

Jackson, Jadin C., Adam Johnson, and A. David Redish. 2006. “Hippocampal Sharp Waves and Reactivation during Awake States Depend on Repeated Sequential Experience.” *Journal of Neuroscience* 26(48):12415–26.

Jackson, Jadin, and A. David Redish. 2007. “Network Dynamics of Hippocampal Cell-Assemblies Resemble Multiple Spatial Maps Within Single Tasks.” *Hippocampus* 17:1209–29.

Ji, Daoyun, and Matthew A. Wilson. 2007. “Coordinated Memory Replay in the Visual Cortex and Hippocampus during Sleep.” *Nature Neuroscience* 10(1):100–107.

Jitendra, Asha K., and Jon R. Star. 2011. “Meeting the Needs of Students with Learning Disabilities in Inclusive Mathematics Classrooms: The Role of Schema-Based Instruction on Mathematical Problem-Solving.” *Theory into Practice* 50(1):12–19.

Joel, Daphna, Yael Niv, and Eytan Ruppin. 2002. “Actor-Critic Models of the Basal Ganglia: New Anatomical and Computational Perspectives.” *Neural Networks* 15(4–6):535–47.

Johansen, Jennifer. 1997. “Instructional Design Models for Well-Structured and III-Structured Problem-Solving Learning Outcomes.” *Educational Technology Research and Development* 45(1):65--94.

Jordan, Jakob, Philipp Weidel, and Abigail Morrison. 2019. “A Closed-Loop Toolchain for Neural Network Simulations of Learning Autonomous Agents.” *Frontiers in Computational Neuroscience* 13(August):1–11.

Kakeyama, Masaki, Toshihiro Endo, Yan Zhang, Wataru Miyazaki, and Chiharu

Tohyama. 2014. “Disruption of Paired-Associate Learning in Rat Offspring Perinatally Exposed to Dioxins.” *Archives of Toxicology* 88(3):789–98.

Kane, Michael J., and Randall W. Engle. 2002. “The Role of Prefrontal Cortex in Working-Memory Capacity, Executive Attention, and General Fluid Intelligence: An Individual-Differences Perspective.” *Psychonomic Bulletin and Review* 9(4):637–71.

Kansky, Ken, Tom Silver, David A. Mély, Mohamed Eldawy, Miguel Lázaro-Gredilla, Xinghua Lou, Nimrod Dorfman, Szymon Sidor, Scott Phoenix, and Dileep George. 2017. “Schema Networks: Zero-Shot Transfer with a Generative Causal Model of Intuitive Physics.”

Kar, Kohitij, Jonas Kubilius, Kailyn Schmidt, Elias B. Issa, and James J. DiCarlo. 2019. “Evidence That Recurrent Circuits Are Critical to the Ventral Stream’s Execution of Core Object Recognition Behavior.” *Nature Neuroscience* 22(6):974–83.

Karachot, Laddawan, Yoshinori Shirai, Réjan Vigot, Tetsuo Yamamori, and Masao Ito. 2001. “Induction of Long-Term Depression in Cerebellar Purkinje Cells Requires a Rapidly Turned over Protein.” *Journal of Neurophysiology* 86(1):280–89.

Karlsson, Mattias P., and Loren M. Frank. 2009. “Awake Replay of Remote Experiences in the Hippocampus.” *Nature Neuroscience* 12(7):913–18.

Kennerley, Steven W., Mark E. Walton, Timothy E. J. Behrens, Mark J. Buckley, and Matthew F. S. Rushworth. 2006. “Optimal Decision Making and the Anterior Cingulate Cortex.” *Nature Neuroscience* 9(7):940–47.

Kesner, Raymond P., Guy Farnsworth, and Bruce V. DiMattia. 1989. “Double Dissociation of Egocentric and Allocentric Space Following Medial Prefrontal and Parietal Cortex Lesions in the Rat.” *Behavioral Neuroscience* 103(5):956–61.

Kesner, Raymond P., Michael R. Hunsaker, and Matthew W. Warthen. 2008. “The CA3 Subregion of the Hippocampus Is Critical for Episodic Memory Processing by Means of Relational Encoding in Rats.” *Behavioral Neuroscience* 122(6):1217–25.

Van Kesteren, Marlieke T. R., Mark Rijpkema, Dirk J. Ruiter, and Guillén Fernández. 2010. “Retrieval of Associative Information Congruent with Prior Knowledge Is Related to Increased Medial Prefrontal Activity and Connectivity.” *Journal of Neuroscience* 30(47):15888–94.

Van Kesteren, Marlieke T. R., Dirk J. Ruiter, Guillén Fernández, and Richard N. Henson. 2012. “How Schema and Novelty Augment Memory Formation.” *Trends in Neurosciences* 35(4):211–19.

Kilgard, Michael P. 1998. “Cortical Map Reorganization Enabled by Nucleus Basalis Activity.” *Science* 279(5357):1714–18.

Kimura, Haijme, and Shigenobu Kobayashi. 1998. “An Analysis of Actor / Critic Algorithms Using Eligibility Traces.” *International Conference on Machine Learning*.

Kolb, Bryan, Kristin Buhrmann, Robert Mcdonald, and Robert J. Sutherland. 1994.

"Dissociation of the Medial Prefrontal, Posterior Parietal, and Posterior Temporal Cortex for Spatial Navigation and Recognition Memory in the Rat." *Cerebral Cortex* 4(6):664–80.

Kumar, M. Ganesh, Cheston Tan, Camilo Libedinsky, Shih-Cheng Yen, and Andrew Y. Y. Tan. 2022. "A Nonlinear Hidden Layer Enables Actor–Critic Agents to Learn Multiple Paired Association Navigation." *Cerebral Cortex* 1–20.

Kumar, M. Ganesh, Cheston Tan, Camilo Libedinsky, Shih-Cheng Yen, and Andrew Yong-Yi Tan. 2021. "One-Shot Learning of Paired Associations by a Reservoir Computing Model with Hebbian Plasticity." *ArXiv Preprint ArXiv:2106.03580*.

Kumaran, Dharshan, Demis Hassabis, and James L. McClelland. 2016. "What Learning Systems Do Intelligent Agents Need? Complementary Learning Systems Theory Updated." *Trends in Cognitive Sciences* 20(7):512–34.

Laird, John. 2021. "An Analysis and Comparison of ACT-R and Soar." *Proceedings of the Ninth Annual Conference on Advances in Cognitive Systems*.

Laird, John E., Allen Newell, and Paul S. Rosenbloom. 1987. "SOAR: An Architecture for General Intelligence." *Artificial Intelligence* 33(1):1–64.

Lebiere, Christian, and John R. Anderson. 1993. "A Connectionist Implementation of the ACT-R Production System." Pp. 635–40 in *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*.

Lechner, Hilde A., and John H. Byrne. 1998. "New Perspectives on Classical Conditioning: A Synthesis of Hebbian and Non-Hebbian Mechanisms." *Neuron* 20(3):355–58.

Legenstein, R., S. M. Chase, A. B. Schwartz, and W. Maass. 2010. "A Reward-Modulated Hebbian Learning Rule Can Explain Experimentally Observed Network Reorganization in a Brain Control Task." *Journal of Neuroscience* 30(25):8400–8410.

Legenstein, Robert, Dejan Pecevski, and Wolfgang Maass. 2008. "A Learning Theory for Reward-Modulated Spike-Timing-Dependent Plasticity with Application to Biofeedback." *PLoS Computational Biology* 4(10).

Lillicrap, Timothy P., Daniel Cownden, Douglas B. Tweed, and Colin J. Akerman. 2016. "Random Synaptic Feedback Weights Support Error Backpropagation for Deep Learning." *Nature Communications* 7(1):13276.

Lillicrap, Timothy P., Adam Santoro, Luke Marris, Colin J. Akerman, and Geoffrey Hinton. 2020. "Backpropagation and the Brain." *Nature Reviews Neuroscience* 21(6):335–46.

Limbacher, Thomas, and Robert Legenstein. 2020. "H-Mem: Harnessing Synaptic Plasticity with Hebbian Memory Networks." in *34th Conference on Neural Information Processing Systems*.

Lindsay, Grace W., Mattia Rigotti, Melissa R. Warden, Earl K. Miller, and Stefano Fusi. 2017. "Hebbian Learning in a Random Network Captures Selectivity Properties of the Prefrontal Cortex." *Journal of Neuroscience* 37(45):11021–36.

Lipton, David M., Ben J. Gonzales, and Ami Citri. 2019. “Dorsal Striatal Circuits for Habits, Compulsions and Addictions.” *Frontiers in Systems Neuroscience* 13(July):1–14.

Litwin-Kumar, Ashok, Kameron Decker Harris, Richard Axel, Haim Sompolinsky, and L. F. Abbott. 2017a. “Optimal Degrees of Synaptic Connectivity.” *Neuron* 93(5):1153-1164.e7.

Litwin-Kumar, Ashok, Kameron Decker Harris, Richard Axel, Haim Sompolinsky, and L. F. Abbott. 2017b. “Optimal Degrees of Synaptic Connectivity.” *Neuron* 93(5):1153-1164.e7.

Lloyd, Kevin, Nadine Becker, Matthew W. Jones, and Rafal Bogacz. 2012. “Learning to Use Working Memory: A Reinforcement Learning Gating Model of Rule Acquisition in Rats.” *Frontiers in Computational Neuroscience* 6(October):1–10.

Lyu, Cheng, L. F. Abbott, and Gaby Maimon. 2022. “Building an Allocentric Travelling Direction Signal via Vector Computation.” *Nature* 601(7891):92–97.

Maas, Andrew L., Awni Y. Hannun, and Andrew Y. Ng. 2013. “Rectifier Nonlinearities Improve Neural Network Acoustic Models.” P. 3 in *Proceedings of the 30th International Conference on Machine Learning*. Vol. 30. Atlanta, Georgia.

Maass, Wolfgang, Prashant Joshi, and Eduardo D. Sontag. 2007. “Computational Aspects of Feedback in Neural Circuits.” *PLoS Computational Biology* 3(1):0015–0034.

Maass, Wolfgang, Thomas Natschläger, and Henry Markram. 2002. “Real-Time Computing without Stable States: A New Framework for Neural Computation Based on Perturbations.” *Neural Computation* 14(11):2531–60.

Mack, Michael L., Alison R. Preston, and Bradley C. Love. 2020. “Ventromedial Prefrontal Cortex Compression during Concept Learning.” *Nature Communications* 11(1):1–11.

Maisson, David J. N., Tyler V. Cash-Padgett, Maya Z. Wang, Benjamin Y. Hayden, Sarah R. Heilbronner, and Jan Zimmermann. 2021. “Choice-Relevant Information Transformation along a Ventrodorsal Axis in the Medial Prefrontal Cortex.” *Nature Communications* 12(1):1–14.

Mansouri, Farshad Alizadeh, David J. Freedman, and Mark J. Buckley. 2020. “Emergence of Abstract Rules in the Primate Brain.” *Nature Reviews Neuroscience* 21(11):595–610.

Mante, Valerio, David Sussillo, Krishna V. Shenoy, and William T. Newsome. 2013. “Context-Dependent Computation by Recurrent Dynamics in Prefrontal Cortex.” *Nature* 503(7474):78–84.

Markram, Henry, Joachim Lübke, Michael Frotscher, and Bert Sakmann. 1997. “Regulation of Synaptic Efficacy by Coincidence of Postsynaptic APs and EPSPs.” *Science* 275(5297):213–15.

Marr, David. 1969. “A Theory of Cerebellar Cortex.” *The Journal of Physiology* 202(2):437–70.

Marr, David. 2010. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* Vol. 80. The MIT Press.

McClelland, James L. 2013. "Incorporating Rapid Neocortical Learning of New Schema-Consistent Information into Complementary Learning Systems Theory." *Journal of Experimental Psychology: General* 142(4):1190–1210.

McGovern, Amy, and Andrew G. Barto. 2001. "Automatic Discovery of Subgoals in Reinforcement Learning Using Diverse Density." *Proceedings of the Eighteenth International Conference on Machine Learning* 361–68.

McKenzie, Sam, and Howard Eichenbaum. 2011. "Consolidation and Reconsolidation: Two Lives of Memories?" *Neuron* 71(2):224–33.

McKenzie, Sam, Andrea J. Frank, Nathaniel R. Kinsky, Blake Porter, Pamela D. Rivière, and Howard Eichenbaum. 2014. "Hippocampal Representation of Related and Opposing Memories Develop within Distinct, Hierarchically Organized Neural Schemas." *Neuron* 83(1):202–15.

McKenzie, Sam, Nick T. M. Robinson, Lauren Herrera, Jordana C. Churchill, and Howard Eichenbaum. 2013. "Learning Causes Reorganization of Neuronal Firing Patterns to Represent Related Experiences within a Hippocampal Schema." *Journal of Neuroscience* 33(25):10243–56.

McNaughton, B. L., and R. G. M. Morris. 1987. "Hippocampal Synaptic Enhancement and Information Storage within a Distributed Memory System." *Trends in Neurosciences* 10(10):408–15.

McNaughton, Bruce L., Francesco P. Battaglia, Ole Jensen, Edvard I. Moser, and May Britt Moser. 2006. "Path Integration and the Neural Basis of the 'Cognitive Map.'" *Nature Reviews Neuroscience* 7(8):663–78.

McVee, Mary B., Kailonnie Dunsmore, and James R. Gavelek. 2005. "Schema Theory Revisited." *Review of Educational Research* 75(4):531–66.

Medina, Javier F., Keith S. Garcia, William L. Nores, Nichole M. Taylor, and Michael D. Mauk. 2000. "Timing Mechanisms in the Cerebellum: Testing Predictions of a Large- Scale Computer Simulation." *Journal of Neuroscience* 20(14):5516–25.

Medina, Javier F., and Michael D. Mauk. 1999. "Simulations of Cerebellar Motor Learning: Computational Analysis of Plasticity at the Mossy Fiber to Deep Nucleus Synapse." *The Journal of Neuroscience* 19(16):7140–51.

Menzel, R., and U. Müller. 1996. "Learning and Memory in Honeybees: From Behavior to Neural Substrates." *Annual Review of Neuroscience* 19:379–404.

Miconi, Thomas. 2017. "Biologically Plausible Learning in Recurrent Neural Networks Reproduces Neural Dynamics Observed during Cognitive Tasks." *ELife* 6:1–24.

Miller, Earl K. 2000. "The Prefrontal Cortex and Cogntive Control." *Nature Reviews Neuroscience* 1(October):13–15.

Miller, Earl K., and Jonathan D. Cohen. 2001. "An Integrative Theory of Prefrontal Cortex Function." *Annual Review of Neuroscience* 167–202.

Minsky, Marvin. 1974. “A Framework for Representing Knowledge.” Pp. 1–116 in *The Psychology of Computer Vision*, edited by P. Winston.

Mnih, Volodymyr, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. “Asynchronous Methods for Deep Reinforcement Learning.” Pp. 1928–37 in *Proceedings of the 33rd International Conference on Machine Learning*. Vol. 48. New York, NY.

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. “Human-Level Control through Deep Reinforcement Learning.” *Nature* 518(7540).

Momennejad, I., E. M. Russek, J. H. Cheong, M. M. Botvinick, N. D. Daw, and S. J. Gershman. 2017. “The Successor Representation in Human Reinforcement Learning.” *Nature Human Behaviour* 1(9):680–92.

Monsell, Stephen. 2003. “Task Switching.” *Trends in Cognitive Sciences* 7(3):134–40.

Montague, P Read, Peter Dayan, and Terrence J. Sejnowski. 1996. “A Framework for Mesencephalic Dopamine Systems Based on Predictive Hebbian Learning.” *Journal of Neuroscience* 16(5):1936–47.

Montague, P. Read, Peter Dayan, and Terrence J. Sejnowski. 1996. “A Framework for Mesencephalic Dopamine Systems Based on Predictive Hebbian Learning.” *Journal of Neuroscience* 16(5):1936–47.

Morris, R. G. M., P. Garrud, J. N. P. Rawlins, and J. O’Keefe. 1982. “Place Navigation Impaired in Rats with Hippocampal Lesions.” *Nature* 297(5868):681–83.

Moser, Edvard I., Emilio Kropff, and May Britt Moser. 2008. “Place Cells, Grid Cells, and the Brain’s Spatial Representation System.” *Annual Review of Neuroscience* 31:69–89.

Moser, May Britt, David C. Rowland, and Edvard I. Moser. 2015. “Place Cells, Grid Cells, and Memory.” *Cold Spring Harbor Perspectives in Biology* 7(2):a021808.

Mukherjee, Arghya, Norman H. Lam, Ralf D. Wimmer, and Michael M. Halassa. 2021. “Thalamic Circuits for Independent Control of Prefrontal Signal and Noise.” *Nature* 600(7887):100–104.

Müller, Martin, and Rüdiger Wehner. 1988. “ Path Integration in Desert Ants, Cataglyphis Fortis .” *Proceedings of the National Academy of Sciences* 85(14):5287–90.

Muller, Robert. 1996. “A Quarter of a Century of Place Cells.” *Neuron* 17(5):813–22.

Murray, James M. 2019. “Local Online Learning in Recurrent Networks with Random Feedback.” *ELife* 8:1–25.

Neelakantan, Arvind, Luke Vilnis, Quoc V. Le, Ilya Sutskever, Lukasz Kaiser, Karol Kurach, and James Martens. 2015. “Adding Gradient Noise Improves Learning

for Very Deep Networks.” 1–11.

Negrón-Oyarzo, Ignacio, Nelson Espinosa, Marcelo Aguilar-Rivera, Marco Fuenzalida, Francisco Aboitiz, and Pablo Fuentealba. 2018. “Coordinated Prefrontal–Hippocampal Activity and Navigation Strategy-Related Prefrontal Firing during Spatial Memory Formation.” *Proceedings of the National Academy of Sciences* 115(27):7123–28.

Nemeroff, Charles B., Helen S. Mayberg, Scott E. Krahl, James McNamara, Alan Frazer, Thomas R. Henry, Mark S. George, Dennis S. Charney, and Stephen K. Brannan. 2006. “VNS Therapy in Treatment-Resistant Depression: Clinical Evidence and Putative Neurobiological Mechanisms.” *Neuropsychopharmacology* 31(7):1345–55.

Neunuebel, Joshua P., and James J. Knierim. 2014. “CA3 Retrieves Coherent Representations from Degraded Input: Direct Evidence for CA3 Pattern Completion and Dentate Gyrus Pattern Separation.” *Neuron* 81(2):416–27.

Nicola, Wilten, and Claudia Clopath. 2019. “A Diversity of Interneurons and Hebbian Plasticity Facilitate Rapid Compressible Learning in the Hippocampus.” *Nature Neuroscience* 22(7):1168–81.

Niv, Yael. 2009. “Reinforcement Learning in the Brain.” *Journal of Mathematical Psychology* 53(3):139–54.

O’Doherty, John, Peter Dayan, Johannes Schultz, Ralf Deichmann, Karl Friston, and Raymond J. Dolan. 2004. “Dissociable Roles of Ventral and Dorsal Striatum in Instrumental Conditioning.” *Science* 304(5669):452–54.

O’Keefe, J., and N. Burgess. 1996. “Geometric Determinants of the Neurons.” *Nature* 381(May):425–28.

O’Keefe, J., and J. Dostrovsky. 1971. “The Hippocampus as a Spatial Map . Preliminary Evidence from Unit Activity in the Freely-Moving Rat.” *Brain Research* 34(1):171–75.

O’Reilly, Randall C., and Michael J. Frank. 2006. *Making Working Memory Work: A Computational Model of Learning in the Prefrontal Cortex and Basal Ganglia*. Vol. 18.

O’Reilly, Randall C., Michael J. Frank, Thomas E. Hazy, and Brandon Watz. 2007. “PVLV: The Primary Value and Learned Value Pavlovian Learning Algorithm.” *Behavioral Neuroscience* 121(1):31–49.

Ohmae, Shogo, and Javier F. Medina. 2015. “Climbing Fibers Encode a Temporal-Difference Prediction Error during Cerebellar Learning in Mice.” *Nature Neuroscience* 18(12):1798–1803.

Ormond, Jake, and John O. Keefe. 2022. “Hippocampal Place Cells Have Goal-Oriented Vector Fields during Navigation.” 607(March 2021).

Packard, Mark G., and James L. McGaugh. 1996. “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.

Palacios-Filardo, Jon, and Jack R. Mellor. 2019. “Neuromodulation of Hippocampal Long-Term Synaptic Plasticity.” *Current Opinion in Neurobiology* 54:37–43.

Palmer, Stephen E. 1975. “Visual Perception and World Knowledge: Notes on a Model of Sensory-Cognitive Interaction.” *Studies of Visual Perception and Problem Solving* (May):279–307.

Parisi, German I., Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. 2019. “Continual Lifelong Learning with Neural Networks: A Review.” *Neural Networks* 113:54–71.

Parthasarathy, Aishwarya, Roger Herikstad, Jit Hon Bong, Felipe Salvador Medina, Camilo Libedinsky, and Shih Cheng Yen. 2017. “Mixed Selectivity Morphs Population Codes in Prefrontal Cortex.” *Nature Neuroscience* 20(12):1770–79.

Parthasarathy, Aishwarya, Cheng Tang, Roger Herikstad, Loong Fah Cheong, Shih-Cheng Yen, and Camilo Libedinsky. 2019. “Time-Invariant Working Memory Representations in the Presence of Code-Morphing in the Lateral Prefrontal Cortex.” *Nature Communications* 10(1):4995.

Pawlak, Verena, Jeffery R. Wickens, Alfredo Kirkwood, and Jason N. D. Kerr. 2010. “Timing Is Not Everything: Neuromodulation Opens the STDP Gate.” *Frontiers in Synaptic Neuroscience* 2(OCT):1–14.

Pfeiffer, Brad E., and David J. Foster. 2013. “Hippocampal Place-Cell Sequences Depict Future Paths to Remembered Goals.” *Nature* 497(7447):74–79.

Pfeiffer, Brad E., and David J. Foster. 2015. “Autoassociative Dynamics in the Generation of Sequences of Hippocampal Place Cells.” *Science* 349(6244):180–83.

Pfister, Jean Pascal, Taro Toyoizumi, David Barber, and Wulfram Gerstner. 2006. “Optimal Spike-Timing-Dependent Plasticity for Precise Action Potential Firing in Supervised Learning.” *Neural Computation* 18(6):1318–48.

Piaget, Jean, Barbel Inhelder, and Harlod Chipman. 1976. *Piaget and His School*. edited by B. Inhelder and H. Chipman. Springer.

Piochon, Claire, Peter Kruskal, Jason MacLean, and Christian Hansel. 2013. “Non-Hebbian Spike-Timing-Dependent Plasticity in Cerebellar Circuits.” *Frontiers in Neural Circuits* 6(DEC):1–8.

Piray, Payam, and Nathaniel D. Daw. 2021. “Linear Reinforcement Learning in Planning, Grid Fields, and Cognitive Control.” *Nature Communications* 12(1):1–20.

Potjans, Wiebke, Markus Diesmann, and Abigail Morrison. 2011. “An Imperfect Dopaminergic Error Signal Can Drive Temporal-Difference Learning” edited by T. Behrens. *PLoS Computational Biology* 7(5):e1001133.

Potjans, Wiebke, Abigail Morrison, and Markus Diesmann. 2009. “A Spiking Neural Network Model of an Actor-Critic Learning Agent.” *Neural Computation* 21(2):301–39.

Poucet, B., and V. Hok. 2017. “Remembering Goal Locations.” *Current Opinion in*

Behavioral Sciences 17:51–56.

Preston, Alison R., and Howard Eichenbaum. 2013. “Interplay of Hippocampus and Prefrontal Cortex in Memory.” *Current Biology* 23(17):R764–73.

Rajalingham, Rishi, Elias B. Issa, Pouya Bashivan, Kohitij Kar, Kailyn Schmidt, and James J. DiCarlo. 2018. “Large-Scale, High-Resolution Comparison of the Core Visual Object Recognition Behavior of Humans, Monkeys, and State-of-the-Art Deep Artificial Neural Networks.” *Journal of Neuroscience* 38(33):7255–69.

Ramsauer, Hubert, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. 2020. “Hopfield Networks Is All You Need.” *ArXiv*.

Reynolds JNJ, Hyland BI, and Wickens JR. 2001. “A Cellular Mechanism of Reward-Related Learning.” *Nature* 413:67–70.

Reynolds, John N. J., and Jeffery R. Wickens. 2002. “Dopamine-Dependent Plasticity of Corticostriatal Synapses.” *Neural Networks* 15(4–6):507–21.

Ribas-Fernandes, José J. F., Alec Solway, Carlos Diuk, Joseph T. McGuire, Andrew G. Barto, Yael Niv, and Matthew M. Botvinick. 2011. “A Neural Signature of Hierarchical Reinforcement Learning.” *Neuron* 71(2):370–79.

Rich, Erin L., and Matthew Shapiro. 2009. “Rat Prefrontal Cortical Neurons Selectively Code Strategy Switches.” *Journal of Neuroscience* 29(22):7208–19.

Rigotti, Mattia, Omri Barak, Melissa R. Warden, Xiao Jing Wang, Nathaniel D. Daw, Earl K. Miller, and Stefano Fusi. 2013. “The Importance of Mixed Selectivity in Complex Cognitive Tasks.” *Nature* 497(7451):585–90.

Rikhye, Rajeev V., Aditya Gilra, and Michael M. Halassa. 2018. “Thalamic Regulation of Switching between Cortical Representations Enables Cognitive Flexibility.” *Nature Neuroscience* 21(12):1753–63.

Riley, Mitchell R., Xue Lian Qi, Xin Zhou, and Christos Constantinidis. 2018. “Anterior-Posterior Gradient of Plasticity in Primate Prefrontal Cortex.” *Nature Communications* 9(1).

Ritter, Samuel, Jane X. Wang, Zeb Kurth-Nelson, Siddhant M. Jayakumar, Charles Blundell, Razvan Pascanu, and Matthew Botvinick. 2018. “Been There, Done That: Meta-Learning with Episodic Recall.” Pp. 6929–38 in *35th International Conference on Machine Learning, ICML 2018*. Vol. 10.

Robins, Anthony. 2004. “Sequential Learning in Neural Networks: A Review and a Discussion of Pseudorehearsal Based Methods.” *Intelligent Data Analysis* 8(3):301–22.

Rolls, Edmund T. 2000. “The Orbitofrontal Cortex and Reward.” *Cerebral Cortex* 10(3):284–94.

Rolls, Edmund T. 2004. “The Functions of the Orbitofrontal Cortex.” *Brain and Cognition* 55(1):11–29.

Rolls, Edmund T. 2007. “An Attractor Network in the Hippocampus: Theory and Neurophysiology.” *Learning and Memory* 14(11):714–31.

Rolls, Edmund T. 2013. “The Mechanisms for Pattern Completion and Pattern Separation in the Hippocampus.” *Frontiers in Systems Neuroscience* 7(OCT):1–21.

Rolls, Edmund T., Wei Cheng, and Jianfeng Feng. 2020. “The Orbitofrontal Cortex: Reward, Emotion and Depression.” *Brain Communications* 2(2).

Rossier, Jérôme, Françoise Schenk, Yulii Kaminsky, and Jan Bures. 2000. “The Place Preference Task: A New Tool for Studying the Relation between Behavior and Place Cell Activity in Rats.” *Behavioral Neuroscience* 114(2):273–84.

Rumelhart, D. E., P. Smolensky, and J. McClelland. 1987. “Schemata and Sequential Thought Processes in PDP Models.” *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Psychological and Biological Models* 2:7–57.

Rumelhart, David E. 1975. *Notes on a Schema for Stories*. ACADEMIC PRESS, INC.

Rumelhart, David E. 1980. “Schemata: The Building Blocks of Cognition.” Pp. 33–58 in *Theoretical Issues in Reading Comprehension*. Routledge.

Rumelhart, David E., and James L. McClelland. 1982. “An Interactive Activation Model of Context Effects in Letter Perception: II. The Contextual Enhancement Effect and Some Tests and Extensions of the Model.” *Psychological Review* 89(1):60–94.

Rumelhart, David E., and Andrew Ortony. 1977. “The Representation of Knowledge in Memory.” *Schooling and the Acquisition of Knowledge* (January 1977):99–135.

Salinas, Emilio, and L. F. Abbott. 2001. “Coordinate Transformations in the Visual System: How to Generate Gain Fields and What to Compute with Them.” *Progress in Brain Research* 130:175–90.

Santoro, Adam, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. “One-Shot Learning with Memory-Augmented Neural Networks.” *33rd International Conference on Machine Learning, ICML 2016* 4:2740–51.

Sarel, Ayelet, Arseny Finkelstein, Liora Las, and Nachum Ulanovsky. 2017. “Vectorial Representation of Spatial Goals in the Hippocampus of Bats.” *Science* 355(6321):176–80.

Scherr, Franz, Christoph Stöckl, and Wolfgang Maass. 2020. “One-Shot Learning with Spiking Neural Networks.” *BioRxiv* 2020.06.17.156513.

Schneider, D. W., and G. D. Logan. 2009. “Task Switching.” Pp. 869–74 in *Encyclopedia of Neuroscience*. Vol. 9. Cambridge, MA: Elsevier.

Schuck, Nicolas W., Ming Bo Cai, Robert C. Wilson, and Yael Niv. 2016. “Human Orbitofrontal Cortex Represents a Cognitive Map of State Space.” *Neuron* 91(6):1402–12.

Schultz, W, P. Dayan, and P. R. Montague. 1997. “A Neural Substrate of Prediction and Reward.” *Science* 275(5306):1593–99.

Schultz, W., P. Dayan, and P. R. Montague. 1997. “A Neural Substrate of Prediction and Reward.” *Science* 275(5306):1593–99.

Seamans, Jeremy K., and Charles R. Yang. 2004. “The Principal Features and Mechanisms of Dopamine Modulation in the Prefrontal Cortex.” *Progress in Neurobiology* 74(1):1–58.

Van Seijen, Harm, A. Rupam Mahmood, Patrick M. Pilarski, Marlos C. Machado, and Richard S. Sutton. 2016. “True Online Temporal-Difference Learning.” *Journal of Machine Learning Research* 17:1–40.

Senn, Walter, and Jean-Pascal Pfister. 2014. “Reinforcement Learning in Cortical Networks.” Pp. 1–9 in *Encyclopedia of Computational Neuroscience*. New York, NY: Springer New York.

Seol, Geun Hee, Jokubas Ziburkus, Shi Yong Huang, Lihua Song, In Tae Kim, Kogo Takamiya, Richard L. Huganir, Hey Kyoung Lee, and Alfredo Kirkwood. 2007. “Neuromodulators Control the Polarity of Spike-Timing-Dependent Synaptic Plasticity.” *Neuron* 55(6):919–29.

Sharma, Sugandha, Sarthak Chandra, and Ila R. Fiete. 2022. “Content Addressable Memory without Catastrophic Forgetting by Heteroassociation with a Fixed Scaffold.”

Sharpe, Melissa J., Chun Yun Chang, Melissa A. Liu, Hannah M. Batchelor, Lauren E. Mueller, Joshua L. Jones, Yael Niv, and Geoffrey Schoenbaum. 2017. “Dopamine Transients Are Sufficient and Necessary for Acquisition of Model-Based Associations.” *Nature Neuroscience* 20(5):735–42.

Sheynikhovich, Denis, Satoru Otani, and Angelo Arleo. 2013. “Dopaminergic Control of Long-Term Depression/Long-Term Potentiation Threshold in Prefrontal Cortex.” *Journal of Neuroscience* 33(34):13914–26.

Skinner, B. F. 1963. “Operant Behavior.” *American Psychologist* 18(8):503–15.

Solstad, Trygve, Charlotte N. Boccara, Emilio Kropff, May-Britt Moser, and Edvard I. Moser. 2008. “Representation of Geometric Borders in the Entorhinal Cortex.” *Science* 322(5909):1865–68.

Song, H. Francis, Guangyu R. Yang, and Xiao Jing Wang. 2017. “Reward-Based Training of Recurrent Neural Networks for Cognitive and Value-Based Tasks.” *ELife* 6:1–24.

Sosa, Marielena, and Lisa M. Giocomo. 2021. “Navigating for Reward.” *Nature Reviews. Neuroscience* 22(August).

Spiers, Hugo J., H. Freyja Olafsdottir, and Colin Lever. 2018. “Hippocampal CA1 Activity Correlated with the Distance to the Goal and Navigation Performance.” *Hippocampus* 28(9):644–58.

Squire, Larry R., Lisa Genzel, John Wixted, and Richard G. M. Morris. 2015. “Memory Consolidation.” *Cold Spring Harbor Perspectives in Biology* 1–22.

Stachenfeld, Kimberly L., Matthew M. Botvinick, and Samuel J. Gershman. 2017. "The Hippocampus as a Predictive Map." *Nature Neuroscience* 20(11):1643–53.

Stalnaker, Thomas A., Nisha K. Cooch, and Geoffrey Schoenbaum. 2015. "What the Orbitofrontal Cortex Does Not Do." *Nature Neuroscience* 18(5):620–27.

Steele, R. J., and R. G. M. Morris. 1999. "Delay-Dependent Impairment of a Matching-to-Place Task with Chronic and Intrahippocampal Infusion of the NMDA-Antagonist D-AP5." *Hippocampus* 9(2):118–36.

Stentz, Anthony. 1997. "Optimal and Efficient Path Planning for Partially Known Environments." Pp. 203–20 in *Intelligent Unmanned Ground Vehicles*. Vol. 1999. Boston, MA: Springer US.

Stokes, Mark G., Makoto Kusunoki, Natasha Sigala, Hamed Nili, David Gaffan, and John Duncan. 2013. "Dynamic Coding for Cognitive Control in Prefrontal Cortex." *Neuron* 78(2):364–75.

Suhaimi, Ahmad, Amos W. H. Lim, Xin Wei Chia, Chunyue Li, and Hiroshi Makino. 2022. "Representation Learning in the Artificial and Biological Neural Networks Underlying Sensorimotor Integration." *Science Advances* 8(22).

Suri, R. E., and W. Schultz. 1999. "A Neural Network Model with Dopamine-like Reinforcement Signal That Learns a Spatial Delayed Response Task." *Neuroscience* 91(3):871–90.

Suri, Roland E., and Wolfram Schultz. 1998. "Learning of Sequential Movements by Neural Network Model with Dopamine-like Reinforcement Signal." *Experimental Brain Research* 121(3):350–54.

Sussillo, David, and L. F. Abbott. 2009. "Generating Coherent Patterns of Activity from Chaotic Neural Networks." *Neuron* 63(4):544–57.

Sutherland, R. J., I. Q. Wishaw, and B. Kolb. 1988. "Contributions of Cingulate Cortex to Two Forms of Spatial Learning and Memory." *Journal of Neuroscience* 8(6):1863–72.

Sutton, Richard S., and Andrew G. Barto. 2020. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.

Tang, Cheng, Roger Herikstad, Aishwarya Parthasarathy, Camilo Libedinsky, and Shih Cheng Yen. 2020. "Minimally Dependent Activity Subspaces for Working Memory and Motor Preparation in the Lateral Prefrontal Cortex." *ELife* 9:1–23.

Team, Open-Ended Learning, Adam Stooke, Anuj Mahajan, Catarina Barros, Charlie Deck, Jakob Bauer, Jakub Sygnowski, Maja Trebacz, Max Jaderberg, Michael Mathieu, Nat McAleese, Nathalie Bradley-Schmieg, Nathaniel Wong, Nicolas Porcel, Roberta Raileanu, Steph Hughes-Fitt, Valentin Dalibard, and Wojciech Marian Czarnecki. 2021. "Open-Ended Learning Leads to Generally Capable Agents."

Todd, Michael T., Yael Niv, and Jonathan D. Cohen. 2009. "Learning to Use Working Memory in Partially Observable Environments through Dopaminergic Reinforcement." *Advances in Neural Information Processing Systems 21 - Proceedings of the 2008 Conference* 1689–96.

Tolman, Edward C. 1948. “Cognitive Maps in Rats and Men.” *Psychological Review* 55(4):189–208.

Tomé, Douglas Feitosa, Sadra Sadeh, and Claudia Clopath. 2022. “Coordinated Hippocampal-Thalamic-Cortical Communication Crucial for Engram Dynamics underneath Systems Consolidation.” *Nature Communications* 13(1):1–18.

Tse, Dorothy, R. F. Langston, Masaki Kakeyama, Ingrid Bethus, Patrick a Spooner, Emma R. Wood, Menno P. Witter, and Richard G. M. Morris. 2007. “Schemas and Memory Consolidation.” *Science* 316(5821):76–82.

Tse, Dorothy, Tomonori Takeuchi, Masaki Kakeyama, Yashushi Kajii, Hiroyuki Okuno, Chiharu Tohyama, and Richard G. M. Morris. 2011. “Schema-Dependent Gene Activation.” *Science* 891(August):891–96.

Tyulmankov, Danil, Ching Fang, Annapurna Vadaparty, and Guangyu Robert Yang. 2021. “Biological Learning in Key-Value Memory Networks.” (NeurIPS).

Urbanczik, Robert, and Walter Senn. 2009. “Reinforcement Learning in Populations of Spiking Neurons.” *Nature Neuroscience* 12(3):250–52.

Vasilaki, Eleni, Nicolas Frémaux, Robert Urbanczik, Walter Senn, and Wulfram Gerstner. 2009. “Spike-Based Reinforcement Learning in Continuous State and Action Space: When Policy Gradient Methods Fail” edited by K. J. Friston. *PLoS Computational Biology* 5(12):e1000586.

van de Ven, Gido M., Hava T. Siegelmann, and Andreas S. Tolias. 2020. “Brain-Inspired Replay for Continual Learning with Artificial Neural Networks.” *Nature Communications* 11(1).

Wallis, Jonathan D., Kathleen C. Anderson, and Earl K. Miller. 2001. “Single Neurons in Prefrontal Cortex Encode Abstract Rules.” *Nature* 411(6840):953–56.

Wang, Jane X., Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z. Leibo, Demis Hassabis, and Matthew Botvinick. 2018. “Prefrontal Cortex as a Meta-Reinforcement Learning System.” *Nature Neuroscience* 21(6):860–68.

Wang, Szu Han, Dorothy Tse, and Richard G. M. Morris. 2012. “Anterior Cingulate Cortex in Schema Assimilation and Expression.” *Learning and Memory* 19(8):315–18.

Wayne, Greg, Chia-Chun Hung, David Amos, Mehdi Mirza, Arun Ahuja, Agnieszka Grabska-Barwinska, Jack Rae, Piotr Mirowski, Joel Z. Leibo, Adam Santoro, Mevlana Gemici, Malcolm Reynolds, Tim Harley, Josh Abramson, Shakir Mohamed, Danilo Rezende, David Saxton, Adam Cain, Chloe Hillier, David Silver, Koray Kavukcuoglu, Matt Botvinick, Demis Hassabis, and Timothy Lillicrap. 2018. “Unsupervised Predictive Memory in a Goal-Directed Agent.” *ArXiv*.

Webb, Christina E., and Nancy A. Dennis. 2019. “Memory for the Usual: The Influence of Schemas on Memory for Non-Schematic Information in Younger and Older Adults.” *Cognitive Neuropsychology* 0(0):1–17.

Weingartner, Herbert. 1981. “Cognitive Processes in Depression.” *Archives of General*

Psychiatry 38(1):42.

Whitlock, Jonathan R., Robert J. Sutherland, Menno P. Witter, May Britt Moser, and Edvard I. Moser. 2008. "Navigating from Hippocampus to Parietal Cortex." *Proceedings of the National Academy of Sciences of the United States of America* 105(39):14755–62.

Whittington, James C. R., Timothy H. Muller, Shirley Mark, Guifen Chen, Caswell Barry, Neil Burgess, and Timothy E. J. Behrens. 2020. "The Tolman-Eichenbaum Machine: Unifying Space and Relational Memory through Generalization in the Hippocampal Formation." *Cell* 183(5):1249-1263.e23.

Widloski, John, and Ila R. Fiete. 2014. "A Model of Grid Cell Development through Spatial Exploration and Spike Time-Dependent Plasticity." *Neuron* 83(2):481–95.

Wilson, Robert C., Yuji K. Takahashi, Geoffrey Schoenbaum, and Yael Niv. 2014. "Orbitofrontal Cortex as a Cognitive Map of Task Space." *Neuron* 81(2):267–79.

Wimmer, Klaus, Duane Q. Nykamp, Christos Constantinidis, and Albert Compte. 2014. "Bump Attractor Dynamics in Prefrontal Cortex Explains Behavioral Precision in Spatial Working Memory." *Nature Publishing Group* 17(3):431–39.

Xiao, Zhuocheng, Kevin Lin, and Jean Marc Fellous. 2020. "Conjunctive Reward–Place Coding Properties of Dorsal Distal CA1 Hippocampus Cells." *Biological Cybernetics* 114(2):285–301.

Xie, Xiaohui, and H. Sebastian Seung. 2004. "Learning in Neural Networks by Reinforcement of Irregular Spiking." *Physical Review E* 69(4):041909.

Xiong, Qiaojie, Petr Znamenskiy, and Anthony M. Zador. 2015. "Selective Corticostriatal Plasticity during Acquisition of an Auditory Discrimination Task." *Nature* 521(7552):348–51.

Xu, Zhongwen, Hado Van Hasselt, and David Silver. 2018. "Meta-Gradient Reinforcement Learning." Pp. 2396–2407 in *32nd Conference on Neural Information Processing Systems*. Montreal, Canada.

Yagishita, S., A. Hayashi-Takagi, G. C. R. Ellis-Davies, H. Urakubo, S. Ishii, and H. Kasai. 2014. "A Critical Time Window for Dopamine Actions on the Structural Plasticity of Dendritic Spines." *Science* 345(6204):1616–20.

Yang, En, Maarten F. Zwart, Mikail Rubinov, Benjamin James, Ziqiang Wei, Sujatha Narayan, Nikita Vladimirov, Brett D. Mensh, James E. Fitzgerald, and Misha B. Ahrens. 2021. "A Brainstem Integrator for Self-Localization and Positional Homeostasis." *BioRxiv* 2021.11.26.468907.

Yang, Guangyu Robert, H. Francis Song, William T. Newsome, and Xiao-Jing Wang. 2019. "Clustering and Compositionality of Task Representations in a Neural Network Trained to Perform Many Cognitive Tasks." *Nature Neuroscience* 183632.

Yang, Guangyu Robert, and Xiao Jing Wang. 2020. "Artificial Neural Networks for Neuroscientists: A Primer." *Neuron* 107(6):1048–70.

Yin, Henry H., and Barbara J. Knowlton. 2006. “The Role of the Basal Ganglia in Habit Formation.” *Nature Reviews Neuroscience* 7(6):464–76.

Yin, Henry H., Sean B. Ostlund, Barbara J. Knowlton, and Bernard W. Balleine. 2005. “The Role of the Dorsomedial Striatum in Instrumental Conditioning.” *European Journal of Neuroscience* 22(2):513–23.

Yonelinas, Andrew P., Charan Ranganath, Arne D. Ekstrom, and Brian J. Wiltgen. 2019. “A Contextual Binding Theory of Episodic Memory: Systems Consolidation Reconsidered.” *Nature Reviews Neuroscience* 20(6):364–75.

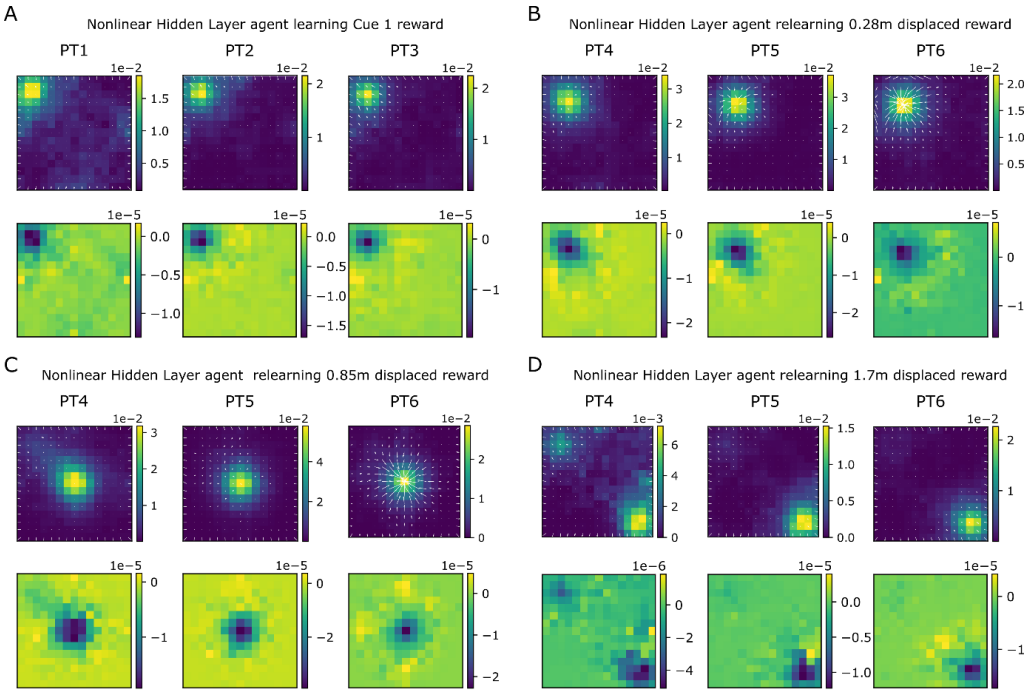
Zannone, Sara, Zuzanna Brzosko, Ole Paulsen, and Claudia Clopath. 2018. “Acetylcholine-Modulated Plasticity in Reward-Driven Navigation: A Computational Study.” *Scientific Reports* 8(1):9486.

Zhang, Zhewei, Zhenbo Cheng, Zhongqiao Lin, Chechang Nie, and Tianming Yang. 2018. “A Neural Network Model for the Orbitofrontal Cortex and Task Space Acquisition during Reinforcement Learning” edited by S. J. Gershman. *PLOS Computational Biology* 14(1):e1005925.

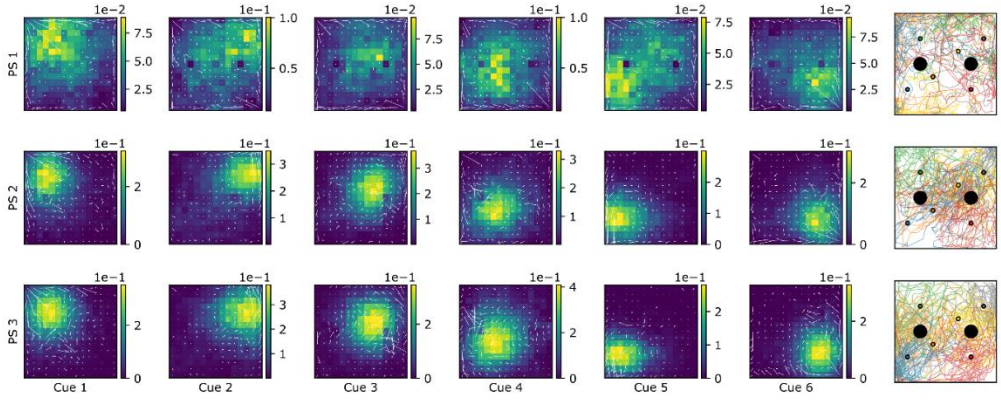
Zhiqing, Zhang. 2015. “Assimilation, Accommodation, and Equilibration: A Schema-Based Perspective on Translation as Process and as Product.” *International Forum of Teaching and Studies* 11(12):84–89.

Zhou, Jingfeng, Chunying Jia, Marlian Montesinos-Cartagena, Matthew P. H. Gardner, Wenhui Zong, and Geoffrey Schoenbaum. 2020. “Evolving Schema Representations in Orbitofrontal Ensembles during Learning.” *Nature* 590(March 2020).

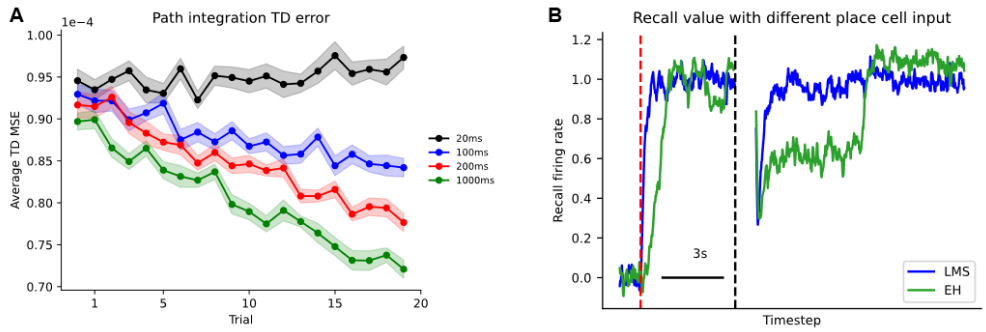
APPENDICES



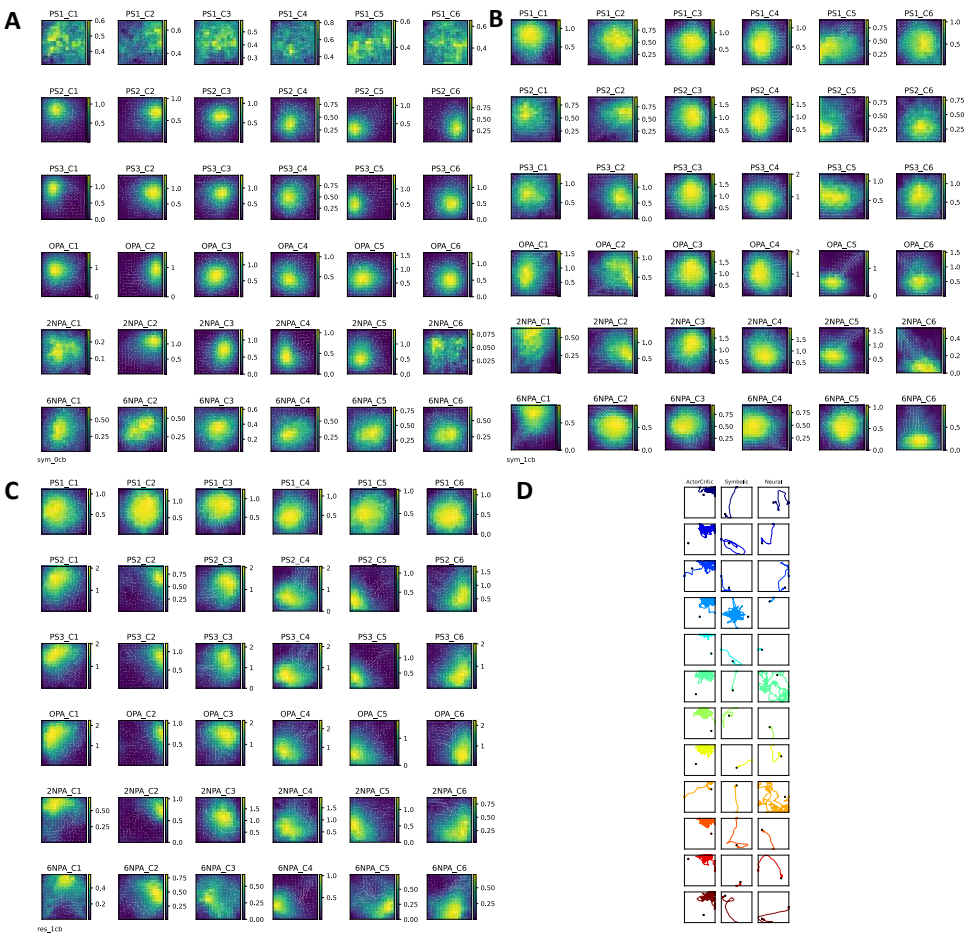
Supplementary Figure 2.1. Policy, value and TD error maps when relearning new rewards. (A) Value (colour) and policy (white arrows) maps (top) and TD error maps (bottom) in PT1, PT2 & PT3 during learning of the original reward location, **(B)** the displaced reward location at 0.28 m from the original reward location, **(C)** the displaced reward location at 0.85 m from the original reward location, and **(D)** the displaced reward location at 1.7 m from the original reward location.



Supplementary Figure 2.2. Value and policy maps for the reservoir agent. Each row shows value and policy maps and example full trajectories for each of the 6 cues in a probe session; top, middle and bottom rows respectively show PS1, PS2 and PS3.



Supplementary Figure 3.1. Learning LEARN METRIC REPRESENTATION and FLAVOUR-LOCATION association. A) Path integration temporal difference error decreases over 20 trials as an agent performs random foraging in an open arena for 300 seconds. The rate of learning depends on the eligibility trace time constant. B) During navigation, the reservoir receives both sensory cue and place cells as input. When an agent receives a reward, plasticity is switched on and the agent's movement is restricted to associate the agent's current coordinates with the reservoir activity. During this period, place cell activity remains constant as the agent is static in the maze. When plasticity is switched on (red to black dashed lines), the recall value reaches 1 when either least mean square or exploratory Hebbian rule are used. However, in the following trial when the agent is moving around the arena, causing the place cell activity to change, the recall value for the synapses trained using the LMS rule averages around 0.95 despite the place cell activity whereas the recall value for the synapses trained by the EH rule averages around 0.61 when the place cell activity is different compared to during the association period but increases to 1.08 when the place cell activity is similar to the association period. Hence, the reservoir trained using the EH rule activates the NAVIGATE schema when the agent gets closer to the goal while the reservoir trained using the LMS rule performs direct heading to demonstrate similar performance as the symbolic agent.



Supplementary Figure 3.2. Value and policy maps for PS1, PS2, PS3, OPA, 2NPA and 6NPA in an open maze arena for A) actor-critic, B) symbolic and C) neural agents. D) Example trajectories by actor-critic, symbolic and neural agents solving the 12NPA task