One-shot learning of paired associations by a reservoir computing model with Hebbian plasticity

Anonymous Author(s)
Affiliation
Address
email

Abstract

One-shot learning can be achieved by algorithms and animals, but how the latter do it is poorly understood as most of the algorithms are not biologically plausible. Experiments studying one-shot learning in rodents have shown that after initial gradual learning of associations between cues and locations, new associations can be learned with just a single exposure to each new cue-location pair. Foster, Morris and Dayan (2000) developed a hybrid temporal difference - symbolic model that exhibited one-shot learning for dead reckoning to displaced single locations. While the temporal difference rule for learning the agent's actual coordinates was biologically plausible, the model's symbolic mechanism for learning target coordinates was not, and one-shot learning for multiple target locations was not addressed. Here we extend the model by replacing the symbolic mechanism with a reservoir of recurrently connected neurons resembling cortical microcircuitry. Biologically plausible learning of target coordinates was achieved by subjecting the reservoir's output weights to synaptic plasticity governed by a novel 4-factor variant of the exploratory Hebbian (EH) rule. As with rodents, the reservoir model exhibited one-shot learning for multiple paired associations.

1 Introduction

Algorithms for one-shot learning in classification [1–4] and navigation [5–7] have recently been developed. Animals are also capable of one-shot learning. In delayed matching to place (DMP) experiments, rats are rewarded for finding a target that remains in the same location for several trials per day, but that is moved to a new location each day. For the first few days, the time taken to find the target gradually decreases with successive trials. But after several days, rats exhibit one-shot learning with near asymptotic performance by the second trial of the day [8]. More recently, Tse et al [9] developed a two-part experiment to study rodent one-shot learning of multiple paired associations. In the first part, rats gradually learn to associate each of several flavor cues with one of several reward locations. In the second part, rats learn new cue-location pairs after a single encounter with each pair. Such biological one-shot learning has been attributed to learning and utilizing schemas [10–13]. As most algorithms for one-shot learning are not biologically plausible, the computations involved in one-shot learning by animals remains poorly understood.

Foster et al [14] developed a hybrid temporal difference - symbolic model of one-shot learning in the DMP task. The agent had a preexisting schema for representing its current location with coordinates, with the correspondence between location and coordinates learned by a biologically plausible generalized temporal difference rule. It also used non-biologically plausible, symbolic computation to learn target coordinates, which were stored in a memory bank. Vector subtraction between current and target coordinates enabled dead reckoning to the target location. Besides the non-biologically plausible method of learning target coordinates, one-shot learning of multiple paired associations was not demonstrated.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Here we build on Foster and colleagues' work to demonstrate several types of agents, varying in biological plausibility, that exhibit one-shot learning in the multiple paired association task. In the first type of agent, the hybrid temporal difference - symbolic spirit is retained, but with a memory bank that uses a non-biologically plausible, symbolic computation to learn cue and location information and an attention-based recall mechanism. The second type of agent uses a reservoir of recurrently-connected neurons to store and recall coordinates while using a non-neural motor controller to reach target locations. In one subtype of reservoir agent, the reservoir's output weights are subject to synaptic plasticity governed by the perceptron rule, which is Hebbian, but also requires preexisting, highly specific connectivity between neurons representing current and target coordinates. The main point of this paper is demonstrated by a second subtype of reservoir agent, in which the reservoir's output weights synaptic plasticity is governed by a novel 4-factor variant of the exploratory Hebbian (EH) rule [15, 16]. The latter is a biologically plausible reinforcement learning agent whose one-shot learning of target coordinates resembles that displayed by rodents in Tse and colleagues' multiple paired association task.

2 Methods

2.1 Tasks

In the DMP task, the agent was required to find a single goal location, randomly chosen from 49 possible locations in an open arena (Fig. 1a), where it would receive reward. The goal location was displaced every five trials, with the last of each group of five trials being an unrewarded probe trial.

Tse et al's multiple paired association task had 2 parts. The first part (Original Paired Associates, OPA) required learning of 6 cue-location associations over 20 sessions (Fig. 1b). Cue information was available to the agent throughout each trial. In the second part (New Paired Associates, NPA), 2 (Fig. 1c), 6 (Fig. 1d) or 12 random cue-location combinations (Fig. 1a) were introduced in a session with a single trial per cue, followed by a probe session.

In both tasks, agents randomly started from midpoints of the north, south, east, or west walls. The arena was 1.6 m x 1.6 m. Reward locations were 3 cm in radius. Trials ended after 600 s, Time steps in simulations represented 100 ms.

2.2 Agents

The first agent type retains the hybrid temporal difference - symbolic spirit of Foster et al. [14]. It has three major modules – episodic memory selection, learning of self-position, and a symbolic motor controller – which are detailed below.

The second agent type uses a reservoir of recurrently connected neurons [16, 17] in place of the episodic memory selection module. Reservoir outputs are trained by either the perceptron rule or a novel 4-factor variant of the exploratory Hebbian (EH) rule [15, 16] detailed below. The symbolic motor controller is retained, which remains a part of this agent that is not neurally implemented.

For comparison, we also used a variant Advantage Actor Critic (A2C) agent [18, 19] (see supplementary material).

An agent's position in the arena was a 2D coordinate vector which was transformed into place cell activity as in [14]. Each agent received cue and place cell activity as input and had 40 actor units representing different output directions. Agent hyperparameters (e.g. learning rate, number of units, reward discount factor) were optimized based on the agent's learning performance in the OPA task.

2.3 Episodic memory

The memory bank comprises of an 18 X 67 Key matrix K and a 18 X 2 Value matrix V. During a trial, the agent receives as input 49 place cell activity and a sensory cue represented by a one-hot vector of size 18. The place cell activity and cue vector are encoded as a query vector $Q(t)$ of dimension 1 X 67. When the agent reaches the correct reward location, the $Q(t)$ is stored in a cue indexed row in K

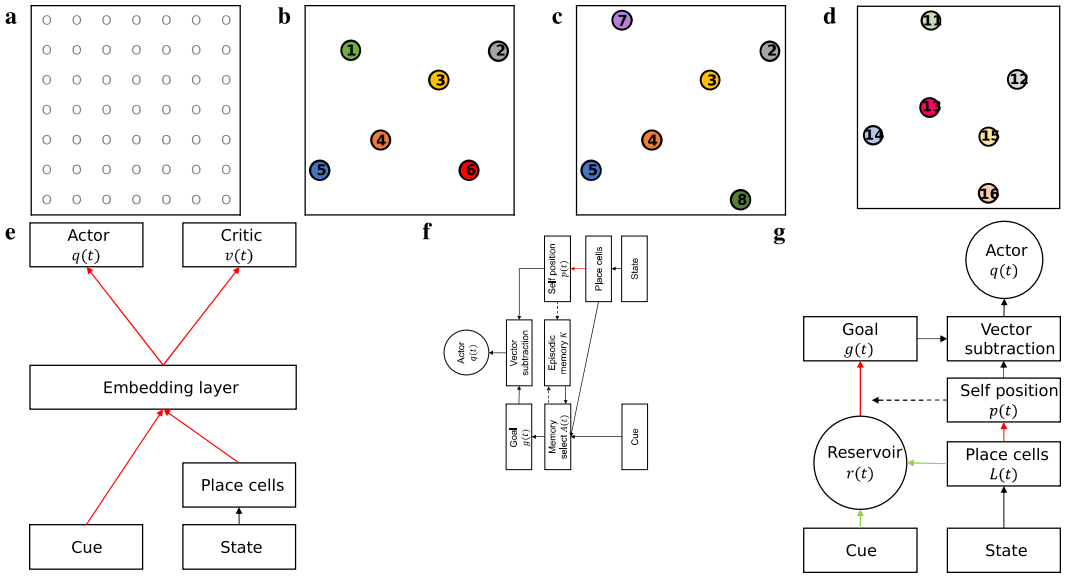

Figure 1: **Multiple paired association task and agent architectures.** a) Open arena with 49 possible reward locations. b-d) Cue-reward location association task as in Tse et al (2007). b) Original Paired Associate (OPA) task with Cues 1 to 6. c) Two New Paired Associate (2NPA) task with Cues 7 and 8 (which were in slightly displaced positions from Cues 1 and 6, respectively) replacing Cues 1 and 6. d) 6NPA task in which six cue-location associations were changed completely to Cues 11 to 16. e-g) Three types of agent architecture. Red arrows indicate synaptic weights that are trained during a task. e) A2C agent with a nonlinear hidden layer trained by backpropagation. f) Symbolic memory and motor controller agent with synaptic weights of the self-position network learnt during training. Black dashed arrows indicate information is stored in the memory, gated by the presence of a reward. g) Reservoir agent with synaptic weights of current and goal coordinates learnt during training.

i.e. Cue 1 - row 1, Cue 2 - row 2, ...) and the learnt position coordinates are stored in the same row in the V.

$$A(t) = \text{softmax}(\beta Q(t) K^T) \tag{1}$$

During the course of a trial, the probability that the current $Q(t)$ is similar to past experiences is represented by $A(t)$, which is determined by taking the dot product between the $Q(t)$ and K followed by a softmax with $\beta = 1$. A dot product between the probability vector A and V is performed to retrieve the coordinates where a reward was previously obtained for the given $Q(t)$.

$$g(t) = A(t)V \tag{2}$$

At the beginning of the simulation, when the agent has not yet reached reward locations associated to each cue, $A(t)$ would be a vector of equal probability. This would return a zero $g(t)$ vector that indicates no goal location was recalled.

This form of episodic memory selection has been previously shown [4, 7] to be effective for reinforcement learning and has been attributed to attention-based networks trained by backpropagation [20, 21].

2.4 Learning self-position coordinates

Foster and colleagues [14] have shown that navigation in openfields can be achieved by learning a coordinate system that is stable throughout trials. This coordinate system, learnt by dead reckoning, allows an agent to estimate its current position while using this as a system to tag goal locations. An agent can then performing vector subtraction between the goal $g(t)$ and current position $p(t)$

coordinates to determine the direction of movement towards the goal. Such a coordinate system can be learnt by minimising Foster's [14] formulation of a general temporal difference error that uses the agent's self-motion estimates $\Delta a(t)$ as the target, current position estimation $p(t)$ and filtered place cell activity $\widehat{L}(t)$. The temporal difference error is computed as

$$\delta(t) = -\Delta a(t) + p(t) - p(t-1) \tag{3}$$

where Δa represents the self-motion estimates for the horizontal and vertical axes. Next, a low pass filter of the place cell activity $\widehat{L}$ is computed

$$\widehat{L}(t) = (1-\alpha)\widehat{L}(t-1) + \alpha L(t) \tag{4}$$

Where $\alpha = \frac{100}{150} ms = 0.67$ controls the filter's smoothing function. The self-position coordinate network comprises of two output units to represent the X and Y coordinates and is directly connected to the place cells (Fig. 1f, 1g). The weights of the self-position network are updated by taking the outer dot product between the low pass filtered place cell activity and the temporal difference error.

$$\Delta W^{selfpos}(t) = \widehat{L}(t)\delta(t) \tag{5}$$

Learning rate for self-position coordinate network was optimised to 0.015.

2.5 Motor controller

The symbolic motor controller performs a vector subtraction between the goal coordinate and the agent's current position coordinate. Using the same attention mechanism in Eq. 1 with β = 4, we use the vector subtraction as the query to determine the action to take out of 40 possible actions to move to the goal location. The computed action is passed to a population of actor units that contains a ring dynamics and noise for exploration. In the event the actor has no goal coordinates, the motor controller output is suppressed allowing the agent to explore the maze freely. (See supplementary material for details).

2.6 Reservoir model and plasticity rules

The concatenated place cell activity and cue vector $Q(t)$ is passed to a reservoir of recurrently connected neurons whose firing rates are given by $r(t) = \tanh[x(t)]$ and the membrane potential $x(t)$ described by

$$x(t) = (1-\alpha)x(t-1) + \alpha\left(\lambda W^{rec} r(t-1) + W^{in} Q(t-1) + \frac{\sigma_{res}}{\sqrt{\alpha}} N(0,1)\right) \tag{6}$$

with λ = 1.5, and σ_{res} = 0.025. The synaptic weights W^{in} are drawn from a uniform distribution between [-1, 1]; W^{rec} are drawn from a Gaussian distribution with mean 0 and variance $1/pN$ with connection probability p = 0.1. These synaptic weights are not subject to synaptic plasticity. Rather, only the synaptic weights from the reservoir to the goal coordinate units W^{out} are subject to synaptic plasticity.

$$g(t) = W^{out} r(t) \tag{7}$$

All trainable parameters are initialized to zero before the onset of learning. Two forms of Hebbian plasticity rules were explored. The first is the perceptron learning rule which takes the difference between the learnt position estimate $p(t)$ and the predicted goal coordinates $g(t)$, as the target to learn, followed by the outer product between the reservoir neuron firing rate $r(t)$ and the computed target

$$\Delta W^{out} = r(t) \cdot (p(t) - g(t)) \cdot R(t) \tag{8}$$

It is important to note that this trace needs to be gated by the reward R(t) that is disbursed when the agent reaches the correct reward location and stays there for approximately 2 seconds or 20

timesteps. This provides the necessary time for the agent to associate the current position $p(t)$ as the goal location $g(t)$ for a given condition using a target modulated Hebbian plasticity rule.

Alternatively, the sparse learning signal from Hoerzer et al. [16] carries much less information about the self learnt target information and was used to determine if one-shot learning could still be achieved even with a sparse learning signal. Firstly, white noise is added to the goal readout units

$$g^{noisy}(t) = g(t) + \frac{\sigma_{sls}}{\sqrt{\alpha}} N(0,1) \tag{9}$$

σ_{sls} = 0.25. Following which a performance measure $P(t)$ is computed to determine if the estimated goal with noise is similar to the position estimate

$$P(t) = -\sum \left(p(t) - g^{noisy}(t)\right)^2 \tag{10}$$

Two low pass filters $\widehat{P}(t)$ and $\widehat{g}(t)$ were instantiated at 0 at the start of each trial for the performance measure and the noisy goal estimate respectively. The low pass filters are formulated similar to the filtered place cell activity in Eq. 4. Next, a modulatory factor $M(t)$ is computed

$$M(t) = \begin{cases} 1, & \text{if } P(t) > \widehat{P}(t) \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

such that if the current performance $P(t)$ is higher than the low pass filtered value $\widehat{P}(t)$, the modulatory factor carries a value of 1 and 0 otherwise. The outer dot product between the reservoir neuron firing rate and the normalized goal readout activity is then taken while gated by both the modulatory factor and the disbursement of the reward

$$\Delta W^{out} = r(t) \cdot (g^{noisy}(t) - \widehat{g}(t)) \cdot R \cdot M(t) \tag{12}$$

to form the novel 4 factor exploratory Hebbian rule. Again, the inclusion of the reward gates the update of the synapses to store the current position as the goal while having a global but sparse learning signal. Due to the intermittent nature of the modulatory factor, the readout units require a longer reward disbursement period of up to 20 seconds or 200 timesteps to learn the target output. Learning rates for both plasticity rules were optimised to 0.05. All simulations ran on the Institute's High Performance Computing CPU Cluster over three days.

3 Results

3.1 Learning displaced single locations

We begin by verifying the ability of four agents, the A2C (Advantage Actor-Critic), Symbolic, Res. Pc. (Reservoir trained by Perceptron rule) and Res. SpLs. (Reservoir trained by sparse learning signal) to learn the displaced single location task. Agents were exposed to a single reward location for four trials followed by a probe trial where plasticity was turned off; thereafter, agents were exposed to a new displaced location randomly chosen out of the remaining 48 locations. All agents, except the A2C agent, reached the newly displaced location significantly faster (i.e. on average, 26.1 ± 2.4 (Symbolic) , 22.8 ± 4.8 (Res. Pc.), and 24.3 ± 3.7 (Res. SpLs.) seconds faster in the second trial compared to the first after the 3rd session), showing one-shot learning of displaced locations (Fig. 2a). We observe the one-shot learning behaviour gradually emerging as the agents took about 12 trials to learn their self-position and in turn, learn and recall the new goal locations after a single trial. Comparatively, the A2C agent showed the opposite trend of reaching newly displaced locations slower as sessions progressed (i.e. on average, A2C agents were 5.0 ± 0.9, 1.6 ± 0.81 and 0.02 ± 0.83 seconds faster in the first, fifth and ninth session respectively to reach the displaced location in the second trial compared to the first).

These results are mirrored during the probe trial as both the symbolic and the reservoir agents show a monotonic increase in time spent at displaced locations. As the agents become more accurate in

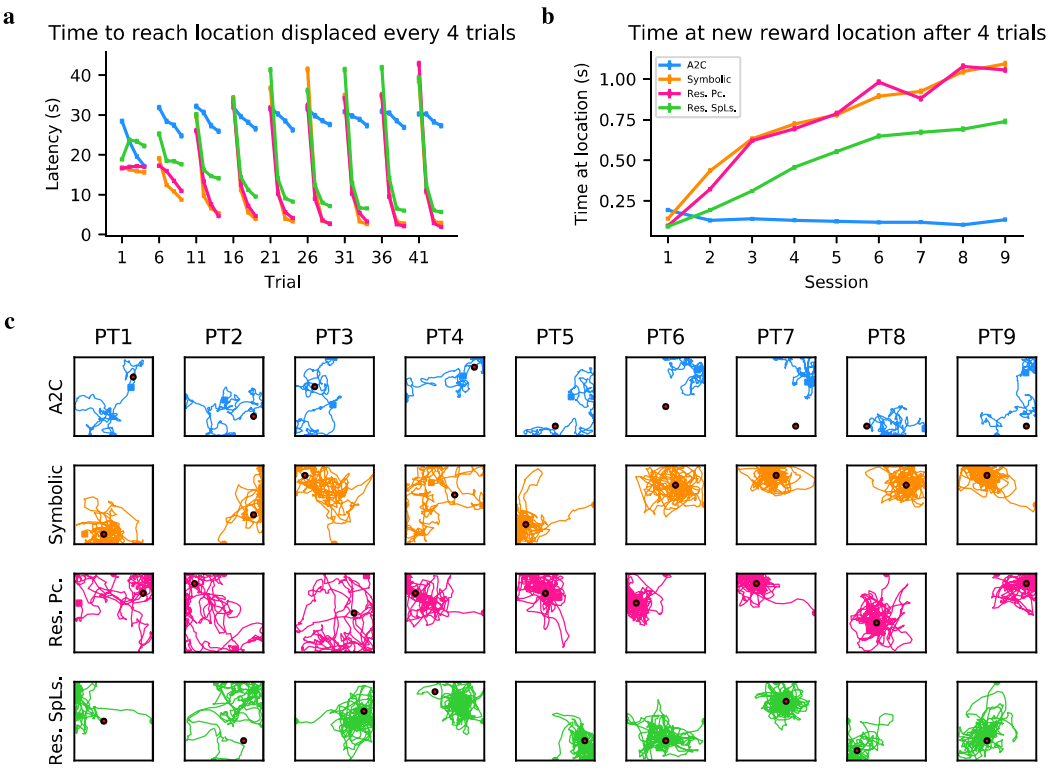

Figure 2: **Learning single locations displaced every 4 trials.** a) Latency to reach single reward locations that are displaced every 4 trials. Agents include Actor Critic agent with hidden layer trained by backpropagation through time (blue), Symbolic agent with episodic memory and motor controller (orange), Reservoir trained by perceptron rule (pink), Reservoir trained by sparse teaching signal (green). b) Amount of time spent at reward location out of 60 seconds. c) Physical trajectories of each agent (row) during the probe trial conducted after 4 training trials as the location changes over 9 epochs (column). Error bars indicate standard error.

estimating their self-position, their goal location estimate becomes more accurate in turn, allowing the agents to quickly and accurately hit the target locations compared to freely moving around in the maze during the probe trials (Fig. 2b).

Instead, A2C agents spent decreasing amount of time at the newly displaced locations as sessions progressed.

This is because incremental learning by backpropagation causes A2C agent's motor policy to converge to a particular location, hence, these agents struggle to find newly displaced locations quickly (Fig. 2c top row). This is why the rate of change in latency to reach the target becomes more gradual from the first session compared to subsequent sessions. Both the symbolic and reservoir agents can overcome this issue by reinitialising the memory entries K,V and the synaptic weights W^{out} to zero respectively if they do not find the target location and the trial ends. These agents can then explore the maze freely to find and learn the new goal locations within the next trial (Fig. 2c second to fourth rows).

3.2 Learning multiple paired associations

With the ability to learn newly displaced single location within one trial, we next investigated the ability of the agents to solve the multiple paired association task. Within each training session, agents were exposed to all six cue-location combinations in a random order. 1b,1c,1d). Agents would only be rewarded if they moved to the location corresponding to the cue before the trial ends. All agents, including the A2C agent, showed a gradual decrease in the average latency to reach all

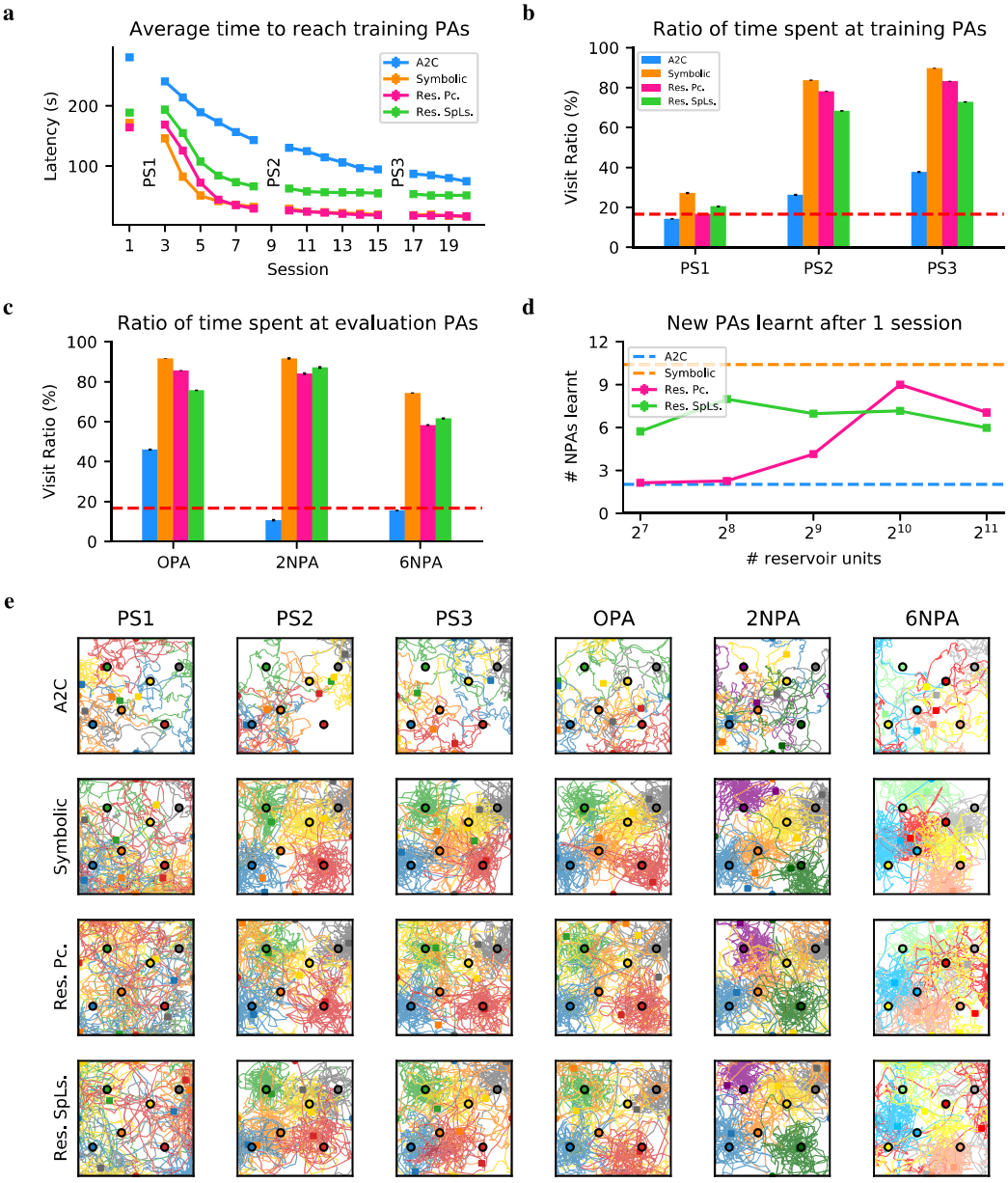

Figure 3: **Learning multiple paired associations and one-shot learning of new paired associates (NPA).** a) Average latency to reach 6 cued locations. b-c) Proportion of time spent at the correct cue-location compared to the wrong location (Visit ratio) during training probe session 1 to 3 (b) and evaluation probe sessions (c). c) Agents were exposed to a single training session with two new paired associations (2NPA) or all six new cue-location combinations (6NPA) after training for 20 sessions on the OPA task. Thereafter an unrewarded probe session was conducted for the new cue-location combinations. Symbolic and reservoir agents show above chance visit ratio performance to 2NPA and 6NPA while the A2C agent shows chance performance for the new conditions. d) Number of new paired associations out of 12 combinations learnt by reservoir agents after 1 training trial and 20 OPA training sessions. e) Physical trajectories of each agent (row) during probe sessions PS1, PS2, PS3 and evaluation on OPA, 2NPA and 6NPA tasks (column). Error bars indicate standard error.

six cue-location combination in Fig. 3a. The Symbolic, Reservoir trained by perceptron rule and Reservoir trained by sparse learning signal showed a faster decrease in latency than the A2C agent before plateauing to a latency of 17, 20 and 48 seconds respectively while the A2C agent took 75 seconds on average in the last session. The Reservoir agent trained by sparse learning signal took 28 seconds longer than its counterpart due to the increase in reward disbursement duration.

During the probe sessions, agents' plasticity is turned off and they would need to spend as much time at the correct location compared to the other locations; this is termed as the visit ratio. A visit ratio of 16.7% indicates chance performance as the agent visits all six locations equally or visits a particular location regardless of the cue presented. All agents showed gradual increase in visit ratios from PS 1 to PS3 (Fig. 3b) with above chance visit ratio performance at PS3 ($p < 0.0001$). It should be noted that, when the trained reservoir was queried with same cue but changing place cell activity $Q(t)$ as the agent moved around the maze, the reservoir was able to maintain its recall of the correct goal coordinates, acting as a content addressable memory system. However, if the cue was not presented throughout the trial, the goal location recalled by the reservoir fluctuates.

After 20 sessions of learning the OPA maze configuration, agents synaptic weights and memory keys were copied and trained on one session of the original configuration (OPA), two new cue-location pairs (2NPA) or six new cue-location pairs (6NPA), followed by a probe session. The New Paired Association condition comprised of two new cue-location pairs (Cue 7 and Cue 8) while keeping cues-location pairs 2 to 5(Fig. 1c). The New Maze condition comprised of six new cue-location pairs (Cues 11 to Cues 16) with different goal locations (Fig. 1d). Unlike in Tse's task [9], the contextual cues such as landmarks, external room cues were not included. All agents, except the A2C agent, showed above chance visit ratios for the two new pairs in NPA condition ($p < 0.0001$) and the six new pairs in 6NPA condition ($p < 0.0001$) demonstrating that the agents can learn two and six new paired associations after just one trial of learning (Fig. 3c). Figure 3e shows the example physical trajectories of all four agents across probe sessions, colour coded according to the cue-location pair.

This prompted the question of the agent's capacity to learn new associations within a single trial. Agents were then trained on OPA for 20 sessions followed by a single training session with 12 new cue-location pairs. The new cues (Cues 7 to Cue 18) and 12 locations were drawn randomly from the remaining 43 possible locations for each agent, minus the locations taken up in OPA. Hence, there was no overlap between the 12 new cue-location pairs and the 6 original pairs. A visit ratio of 8.3% indicates chance performance but if the agent achieves a visit ratio of greater than 16.7%, that cue-location pair was considered to been learnt. For this series of simulations, the total exploration time was increased to 1000 seconds to provide more time for the agents to find the new locations. Out of 200 simulations, the symbolic agent was able to learn 10.4 ± 1.1 new cue-location pairs on average after just 1 trial of learning. The number of pairs learnt decreased to 8.5 ± 2.2 and 5.5 ± 1.9 if the exploration time was reduced from 1000 seconds to 600 and 100 seconds, respectively. With a greater exploration time, the symbolic agent should be able to learn all 12 cue-location pairs. Conversely, the reservoir agent with 1024 recurrent units was able to learn 8.5 ± 2.4 pairs when trained with the perceptron rule and 6.6 ± 2.5 pairs when trained with the sparse learning signal (Fig 3d). The maximum number of pairs that could be learnt within a single session increased when the number of units in the reservoir trained by the perceptron plasticity rule increased, indicating the size of the reservoir does affect the agent's one-shot learning capacity. Interestingly, the reservoir agent trained by the sparse learning signal with only 256 recurrent units was able to learn 8.7 ± 2.1 new pairs after a single trial, indicating a better one-learning potential of multiple pairs compared to the Reservoir agent trained by the perceptron rule. Nevertheless, we have shown that a reservoir agent trained by a sparse learning signal has the capacity to learn up to 84% of the new cue-location pairs a symbolic agent is able to learn in one-shot given the environment and training constraints.

4 Discussion

Our main result is built upon Foster and colleagues' work to demonstrate that an agent with a reservoir model, whose output weights are trained by a novel 4-factor variant of the exploratory Hebbian (EH) rule [15, 16], can learn target coordinates after a single trial of training. This is a biologically plausible reinforcement learning agent whose one-shot learning of target coordinates resembles that displayed by rodents in Tse and colleagues' multiple paired association task.

Although deep reinforcement learning agents such as the A2C showed initial learning, the synapses in these agents are adjusted incrementally, causing them to slowly converge to a single environment [22]. This form of learning however, restricts their ability to store new information quickly to adapt to dynamic environments, a deed that Hebbian plasticity based models seem to be able to achieve [2, 3]. However, instead of using Hebbian plasticity to store and recall input patterns in Hopfield networks for a reinforcement learning paradigm, we show that a similar outcome can be achieved by training only the readout synapses of a reservoir model.

There are several limitations to our current work. Firstly, Tse's multiple paired association task requires rodents to figure out the Flavour – Location schema i.e. each odour cue given is associated to a specific reward location whereas, this schema is hand crafted for the symbolic and reservoir agents. This requires the development of a schema learning model that has to figure out which information to gate before the reservoir or memory bank stores it. However, assigning these computations to anatomical structures such as the prefrontal cortex [23] or the hippocampus [24, 25] is still premature. Secondly, these agents can only take direct paths towards its goal locations using a symbolic motor controller that suppresses its activity if the goal coordinate is unspecified. In addition, since the symbolic motor controller uses dead reckoning to reach the goal location, the agent might not be able to navigate past obstacles. Hence, we have yet to formulate a fully neural implementation of an agent that is able to to navigate to new locations, past obstacles as this would more closely resemble the one-shot navigation behaviour observed in animals. Perhaps a hybrid actor-critic variant with a reservoir to store episodic memories could address this limitation.

References

[1] Jane X Wang, Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Demis Hassabis, and Matthew Botvinick. Prefrontal cortex as a meta-reinforcement learning system. *Nature neuroscience*, 21(6):860–868, 2018.

[2] James CR Whittington, Timothy H Muller, Shirley Mark, Guifen Chen, Caswell Barry, Neil Burgess, and Timothy EJ Behrens. The tolman-eichenbaum machine: Unifying space and relational memory through generalization in the hippocampal formation. *Cell*, 183(5):1249–1263, 2020.

[3] Thomas Limbacher and Robert Legenstein. H-mem: Harnessing synaptic plasticity with hebbian memory networks. *bioRxiv*, 2020.

[4] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR, 2016.

[5] Andrea Banino, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, Martin J Chadwick, Thomas Degris, Joseph Modayil, et al. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705):429–433, 2018.

[6] Greg Wayne, Chia-Chun Hung, David Amos, Mehdi Mirza, Arun Ahuja, Agnieszka Grabska-Barwinska, Jack Rae, Piotr Mirowski, Joel Z Leibo, Adam Santoro, et al. Unsupervised predictive memory in a goal-directed agent. *arXiv preprint arXiv:1803.10760*, 2018.

[7] Samuel Ritter, Jane Wang, Zeb Kurth-Nelson, Siddhant Jayakumar, Charles Blundell, Razvan Pascanu, and Matthew Botvinick. Been there, done that: Meta-learning with episodic recall. In *International Conference on Machine Learning*, pages 4354–4363. PMLR, 2018.

[8] RJ Steele and RGM Morris. Delay-dependent impairment of a matching-to-place task with chronic and intrahippocampal infusion of the nmda-antagonist d-ap5. *Hippocampus*, 9(2): 118–136, 1999.

[9] Dorothy Tse, Rosamund F Langston, Masaki Kakeyama, Ingrid Bethus, Patrick A Spooner, Emma R Wood, Menno P Witter, and Richard GM Morris. Schemas and memory consolidation. *Science*, 316(5821):76–82, 2007.

[10] James L McClelland. Incorporating rapid neocortical learning of new schema-consistent information into complementary learning systems theory. *Journal of Experimental Psychology: General*, 142(4):1190, 2013.

[11] Asaf Gilboa and Hannah Marlatte. Neurobiology of schemas and schema-mediated memory. *Trends in cognitive sciences*, 21(8):618–631, 2017.

[12] Marlieke TR Van Kesteren, Dirk J Ruiter, Guillén Fernández, and Richard N Henson. How schema and novelty augment memory formation. *Trends in neurosciences*, 35(4):211–219, 2012.

[13] Guillén Fernández and Richard GM Morris. Memory, novelty and prior knowledge. *Trends in Neurosciences*, 41(10):654–659, 2018.

[14] David J Foster, Richard GM Morris, and Peter Dayan. A model of hippocampally dependent navigation, using the temporal difference learning rule. *Hippocampus*, 10(1):1–16, 2000.

[15] Robert Legenstein, Steven M Chase, Andrew B Schwartz, and Wolfgang Maass. A reward-modulated hebbian learning rule can explain experimentally observed network reorganization in a brain control task. *Journal of Neuroscience*, 30(25):8400–8410, 2010.

[16] Gregor M Hoerzer, Robert Legenstein, and Wolfgang Maass. Emergence of complex computational structures from chaotic neural networks through reward-modulated hebbian learning. *Cerebral cortex*, 24(3):677–690, 2014.

[17] David Sussillo and Larry F Abbott. Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557, 2009.

[18] Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.

[19] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[21] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.

[22] Matthew Botvinick, Sam Ritter, Jane X Wang, Zeb Kurth-Nelson, Charles Blundell, and Demis Hassabis. Reinforcement learning, fast and slow. *Trends in cognitive sciences*, 23(5):408–422, 2019.

[23] Jingfeng Zhou, Chunying Jia, Marlian Montesinos-Cartagena, Matthew PH Gardner, Wenhui Zong, and Geoffrey Schoenbaum. Evolving schema representations in orbitofrontal ensembles during learning. *Nature*, 590(7847):606–611, 2021.

[24] Pierre Baraduc, J-R Duhamel, and Sylvia Wirth. Schema cells in the macaque hippocampus. *Science*, 363(6427):635–639, 2019.

[25] Sam McKenzie, Andrea J Frank, Nathaniel R Kinsky, Blake Porter, Pamela D Rivière, and Howard Eichenbaum. Hippocampal representation of related and opposing memories develop within distinct, hierarchically organized neural schemas. *Neuron*, 83(1):202–215, 2014.

[26] Nicolas Frémaux, Henning Sprekeler, and Wulfram Gerstner. Reinforcement learning using a continuous time actor-critic framework with spiking neurons. *PLoS Comput Biol*, 9(4): e1003024, 2013.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No] We do not forsee negative societal impact due to this work.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Specific training details will be included as supplementary information. Code for all experiments will be published with the paper.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

A.1 Place cells

All agents have 49 place cells which perform a Gaussian transformation of the agent's position in the maze according to

$$u^{pc} = e^{-\frac{(x(t)-x_i)^2}{2\sigma_{pc}^2}} \tag{13}$$

Where $x(t)$ indicates the agents position in the square maze with bounded by $x = (0.8m, 0.8m)$. σ_{pc} = 0.2666.. m and place cells are spaced regularly apart in a 7 by 7 grid, while covering the boundary. The sensory cue passed to the agents is encoded by u^{cue}, which is a one-hot encoded vector of length 18 with gain 3. The cue vector is presented to the agent throughout the trial period. The place cell activity and the sensory cue is concatenated to form the input vector $Q(t)$ to all agents

$$Q(t) = [u^{pc}, u^{cue}] \tag{14}$$

A.2 Actor

All agents have M = 40 actor units where each kth unit represents a spatial direction . The firing rate of each actor unit is $\rho(t) = \text{ReLU}[q(t)]$ and the membrane potential q has dynamics

$$q(t) = (1-\alpha)q(t-1) + \alpha\left(\phi_{mc}(t-1) + \sum W^{lateral}\rho(t-1) + \frac{\sigma_{actor}}{\sqrt{\alpha}}N(0,1)\right) \tag{15}$$

with $\alpha = \frac{100}{150}$ = 0.667... and σ_{actor} = 0.25. $\phi(t)$ represents the output from the motor controller module. The lateral synaptic weights is given by

$$W^{lateral} = \frac{w_-}{M} + w_+\frac{f(k,h}{\sum f(k,h} \tag{16}$$

with $f(k,h) = (1-\delta)e^{\psi cos(\theta-\theta)}$, $w_- = -1$, $w_+ = 1$ abd $\psi = 20$, connect the actor units into a ring attractor that smoothens the agent's spatial trajectory. The direction of movment is chosen by

$$a(t) = \frac{1}{M}\sum \rho(t)K^{dir} \tag{17}$$

which is the vector sum of directions weighed by each actor unit's firing rate with $K^{dir} = a_0(\sin\theta, \cos\theta)$ and $a_0 = 0.03$. The Advantage Actor Critic (A2C) agent on the other hand chooses one direction of movement out of K^{dir} based on a stochastic action policy

$$\rho(t) = \text{softmax}(W^{actor}h(t)) \tag{18}$$

where $h(t)$ is the nonlinear hidden layer activity.To match the same speed achieved by the other agents, $a_0 = 0.07$ was chosen to increase the step size of a particular action in K^{dir}. The direction of movement is smoothened using a low pass filter

$$a(t) = (1-\alpha_{a2c})\widehat{a}(t) + \alpha_{a2c}\rho(t) \tag{19}$$

where $\alpha_{a2c} = 0.25$ as in how Foster, Dayan & Morris (2000) [14] smoothened the trajectory of the actor-critic agent that chose discrete actions. The Advantage Actor-Critic reinforcement algorithm was implemented as in [1, 18, 19] where the agent is allowed to run through an entire trial, storing the various state, reward and actions taken before the weights are updated. Instead of the asynchornous method, only one cpu thread was used to run the synchronous method for each agent. The gradient is computed at the end of each trial according to a weighted sum of the policy π, value function V and entropy regularisation term H according to

$$\nabla L = \nabla L_\pi + \nabla L_v + \nabla L_{ent}$$
$$= \frac{\partial \log \pi(a(t)|s(t),\theta)}{\partial \theta}\delta(t) +$$
$$\beta_v \delta(t)\frac{\partial V}{\partial \theta_v} + \beta_e \left(\frac{\partial H(\pi(a(t)|s(t),\theta))}{\partial \theta}\right)$$
$$\delta(t) = [R^{disc}(t) - V(s(t),\theta_v)]$$
$$R^{disc}(t) = \sum_{t=0}^{T} \gamma^{t-1} r(t) \tag{20}$$

where θ and θ_v are the synaptic weight parameters for the policy and value function, and $\gamma = 0.99$ is the reward discount factor. Hyperparameters $\beta_v = 0.5$ and $beta_e = -0.001$ control the value estimate loss and entropy regularisation term contributions to the total loss. Lastly, $\delta(t)$ is the temporal difference error that informs the actor-critic of the advantage incurred. The critic is a single linear unit, actor is softmax activated and the hidden layer uses a ReLU activiation function variant where if the unit activity is above a threshold value of 3, the value is retained otherwise, it is converted to a 0. Using this variant showed a faster convergence in training compared to original ReLU function. Synaptic weights are updated using the RMSprop optimiser with learning rate 0.000035.

A.3 Reward disbursement

Agents are free to explore the arena till the trial ends but if it finds the reward before, the agent remains stationary at the reward location until the trial ends to model consummatory behaviour. After the agent reaches the reward, a total reward value $R = 4$ is disbursed at a reward rate $r(t)$, similar to [26], given by

$$r_a(t) = 1 - \frac{100}{\tau_a} r_a(t), \quad r_b(t) = 1 - \frac{100}{\tau_b} r_b(t),$$
$$r(t) = \frac{r_a(t) - r_b(t)}{\tau_a - \tau_b} \tag{21}$$

with $\tau_a = 120ms$ and $\tau_b = 250ms$ for all agents except the Reservoir agent trained by sparse learning signal where $\tau_b = 2500ms$. When the agent reaches the reward, it is updated according to

$$r_a(t) \to r_a(t) + R \qquad r_b(t) \to r_b(t) + R \tag{22}$$

such that r(t) integrates to R. In trials where the agent does not reach the reward location, no punishment is given, except for the A2C agent where a negative reward $R = -1$ is given to penalise the actions taken, else the agent converges to a stationary action policy.

A.4 Motor Controller

The symbolic motor controller is formulated by taking the goal $g(t)$ and current position $p(t)$ coordinates as inputs and performing vector subtraction. The resultant vector $Q^{vecsub}(t)$ specifies the direction and magnitude in which the agent needs to move in order to reach the goal location.

However, $Q^{vecsub}(t)$ does not specify which action, out of 40, the agent should take. Hence, a dot product is taken between the resultant vector $Q^{vecsub}(t)$ and the 40 possible actions K^{dir} while using $\beta_{mc} = 4$ as a scaling factor before taking a softmax to obtain a firing rate equivalent profile ϕ_{mc} where the combined activity is normalised to one.

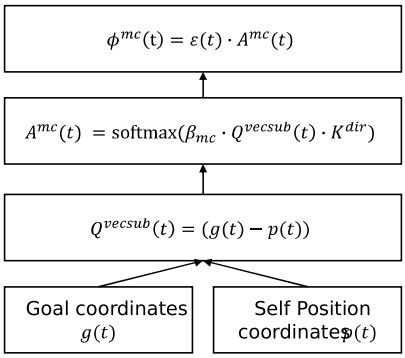

Figure 4: **Symbolic motor controller architecture.** Goal and current position coordinates are taken in as inputs before performing vector subtraction and choosing the direction of movement.

To allow the agent to freely explore the maze and find the target location versus turning on the motor controller for directed movement, a recalled coordinate was considered to be a goal only if the L_2 norm of the goal coordinate was greater than a threshold ω

$$\varepsilon(t) = \begin{cases} 1, & \text{if } \|g(t)\|^2 > \omega \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

where $\omega = 0.15$. If the L_2 norm of the goal coordinate is greater, the output of the motor controller is not suppressed by ε. Instead, if the L_2 norm is lower than the threshold, the output of the motor controller ϕ_{mc} is suppressed by ε.

This is because the episodic memory bank matrix used by the symbolic agent was initialised to 0 and the goal coordinate output by the reservoir was 0 since the synaptic weights W^{out} was initialised as 0. Such a control mechanism is similar to [14] where the agent freely explores the maze if the agent's memory was empty and performs goal directed movement if it was not empty. However, a reservoir network will not be able to output an empty target coordinate. Conversely, using $\omega = 0.15$ meant that the goal location with coordinates (0,0) cannot be considered as a target and the agent will instead explore the maze. This bias against the central goal location is not observed for the other goal locations spread out in the maze.

Instead, the ω threshold should be lowered to value below 0.025 such that the coordinate (0,0) can be considered as goal. This is because the agent only stores the coordinate at which it hits the target, which is $\pm$ 0.03 from the center of the goal coordinate instead of the center of the goal coordinate. Lowering the threshold causes the agent to employ the motor controller and move to the center of the maze when it encounters a new input $Q(t)$. An unforeseen input e.g. Cue 7 to Cue 18 during single session NPA training causes both the attention mechanism in the episodic memory bank and the reservoir to recall a goal coordinate vector with L_2 norm closer to (0,0). This could be a similar strategy animals or humans employ, moving to the center and searching for locations from there. This limits the agent's exploration mostly to the center. However, due to the stochasticity in the ring attractor, the agent can still explore up to the periphery of the maze and find the goal location, but the duration of the trial needs to be increased from 600 seconds to 3600 seconds. This will provide the agent ample time to find goal locations nearer to the boundaries. In subsequent simulations, increasing the time to 3600 seconds allow the Symbolic and Reservoir agents to learn up 12 NPA and 9 NPAs respectively.

Alternatively, the decision to suppress the motor controller's involvement can be computed using a similar metric

$$\varepsilon(t) = \begin{cases} 1, & \text{if } \max A^{mc}(t) > \omega \\ 0, & \text{otherwise.} \end{cases} \tag{24}$$

with $\omega = 0.1$. The motor controller is recruited if the action unit with the highest probability crosses the threshold ω. This happens when the agent is further away from the recalled goal location. When

the agent reaches the vicinity of the goal location, the probability drops below the threshold and the agent switches to an exploration mode. Similar algorithms can be either handcrafted or learnt by the agent to switch between these explore-exploit modalities.

A.5 Symbolic memory & self-position weights

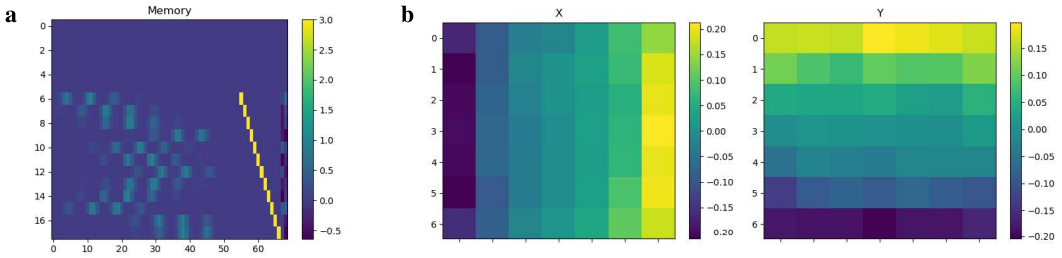

Figure 5: **Episodic memory bank and self-position network weights.** a) 2D episodic memory bank with each row storing 49 place cell activity, cue vector of size 18 and 2D goal coordinates. Example Cue 11 to Cue 18 information are stored in the memory bank. b) Synaptic weights of self-position coordinate network to estimate an agent's 2D coordinate in the square maze. Weights converge to a similar form as in Foster, Morris & Dayan (2000) [14] but for a square maze.

A.6 12NPA trajectory

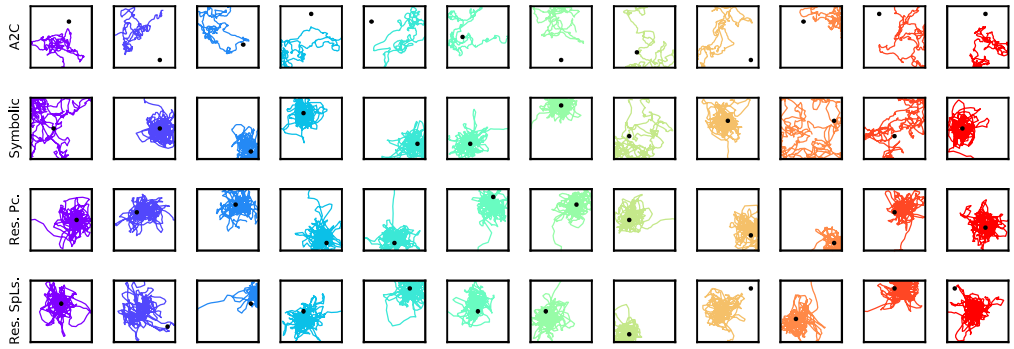

Figure 6: **Example trajectory of agents solving 12 new paired associations (12NPA).** Circles and squares without border indicate start and end positions respectively. A2C agent navigates to five out of 12 locations but achieves low visit ratio at each paired associate. Comparatively, the Symbolic and Reservoir agents spend most of the time at the correct cue-location pairs during each cued probe trial.