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

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

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.

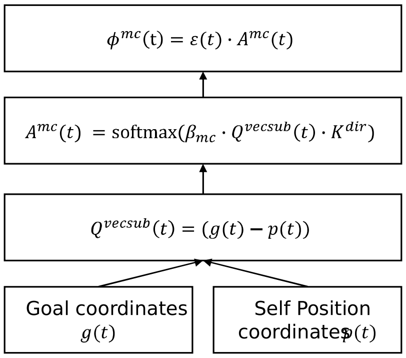
$\phi^{mc}(t) = \varepsilon(t) \cdot A^{mc}(t)$
$A^{mc}(t) = \text{softmax}(\beta_{mc} \cdot Q^{vecsub}(t) \cdot K^{dir})$
$Q^{vecsub}(t) = (g(t) - p(t))$
Goal coordinates $g(t)$
Self Position coordinates $p(t)$

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 ± 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

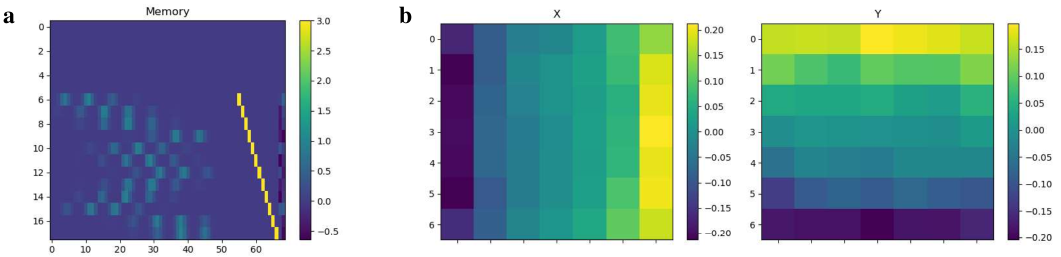

Figure 5: **Episodic memory bank and self-position network weights.** a) 2D episodic memory bank with each row storing 49 place cell activity, cue vector of size 18 and 2D goal coordinates. Example Cue 11 to Cue 18 information are stored in the memory bank. b) Synaptic weights of self-position coordinate 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

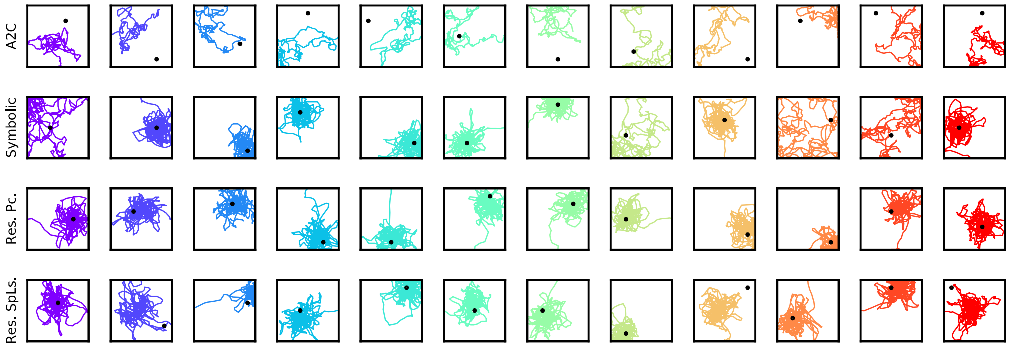

Figure 6: **Example trajectory of agents solving 12 new paired associations (12NPA).** Circles and squares without border indicate start and end positions respectively. A2C agent navigates to five out of 12 locations but achieves low visit ratio at each paired associate. Comparatively, the Symbolic and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