Hippocampal networks support reinforcement learning in partially observable environments

Dabal Pedamonti[1,2*], **Samia Mohinta**[1,6*], **Martin V. Dimitrov**[1], **Hugo Malagon-Vina**[5], **Stephane Ciocchi**[4], **Rui Ponte Costa**[1,2,3]

[1]Computational Neuroscience Unit, Intelligent Systems Labs Faculty of Engineering, University of Bristol, Bristol, United Kingdom; [2]Centre for Neural Circuits and Behaviour, Department of Physiology, Anatomy and Genetics, University of Oxford, Oxford, United Kingdom; [4]Laboratory of Systems Neuroscience, Department of Physiology, University of Bern, Bern, Switzerland; [3]Department of Physiology, University of Bern, Bern, Switzerland; [5]Center for Brain Research, Division of Cognitive Neurobiology, Medical University of Vienna, Vienna, Austria; [6]Department of Physiology, Development and Neuroscience, University of Cambridge, United Kingdom

Abstract Mastering navigation in environments with limited visibility is crucial for survival. While the hippocampus has been associated with goal-oriented navigation, its specific role in real-world behaviour, particularly in scenarios with partial observability, remains elusive. To investigate this, we combined deep reinforcement learning (RL) modelling with behavioural and neural data analysis. First, we trained RL agents to perform reward-based navigational tasks in partially observable environments. We show that agents equipped with recurrent hippocampal circuitry, as opposed to a purely feedforward network, successfully learned the tasks, resembling animal behaviour. By employing neural dimensionality reduction, our models predicted reward, strategy and temporal representations, which we validated using large-scale hippocampal neuronal recordings. Moreover, hippocampal RL agents predicted state-specific trajectories and action certainty, which mirror empirical findings. In contrast, agents trained in fully observable environments failed to capture experimental data, suggesting that partial observability is implicit in goal-driven tasks. Finally, we show that hippocampal-like RL agents demonstrated improved generalisation across novel task conditions. In summary, our findings suggest a key role of hippocampal networks in facilitating learning in naturalistic environments.

Introduction

As we navigate new environments, we must learn to integrate incomplete sensory information towards desired goals. How biological neural networks perform this feat is not fully understood.

The hippocampus is classically associated with building a cognitive map of the environment and the storage of episodic memories [1–3]. However, growing evidence suggests that the hippocampus also supports goal-driven behaviour [4–8]. For example, Wikenheiser and Redish [4] showed that the hippocampus is indeed involved in planning routes towards desired goals. Moreover, their work suggests that hippocampal sequence events, known as "replay", serve as a mechanism for goal-directed navigation, facilitating memory-based trajectory planning and guiding subsequent navigational behaviour. Other studies have shown that the hippocampus, and the hippocampal CA3 region in particular, is involved in maintaining information in working memory that is needed during navigational tasks when sensory cues are no longer present [9–11]. Given that in most naturalistic conditions animals do not have contin-

*These authors contributed equally to this work.

For correspondence: rui.costa@dpag.ox.ac.uk (RPC)

uous access to the full environment, we postulate that the hippocampus may have evolved to support goal-driven navigation in environments in which sensory information is not always present.

The hippocampus has been traditionally conceptualised using Hopfield neural networks, known for their capacity for autoassociative memory storage [12, 13]. More recent studies have demonstrated that recurrent neural network models of the hippocampus, trained for navigation tasks, exhibit specific cell types tuned to spatial information [14–17], including the commonly observed place cells and grid cells [18, 19]. Furthermore, some of these models have also considered the role of hippocampal networks in reward-based navigation tasks [6, 7, 20]. However, how hippocampal networks contribute to navigating goal-oriented environments under realistic conditions and what are its implications for our understanding of animal behaviour and the underlying neural substrates has remained unclear.

Here, we show that the hippocampal circuitry is well placed to deal with environments with realistic conditions, such as limited visibility and cue uncertainty. To that end, we combine behavioural and neural data analysis together with deep reinforcement learning (RL) modelling on similar task setups. Both animals and agents were trained to perform ego-allocentric strategies on a T-maze. Our models consist of a neural network with three-layered hippocampal-like structure trained in a reinforcement learning setting. By contrasting experimental observations with the model we show that hippocampal networks trained in partial, but not fully observable environments, provide a good match of neuronal and behavioural observations. Using task-relevant dimensionality reduction we show that hippocampal neurons encode decision, strategy and temporal population activity that can only be explained by a model with CA3-like recurrence. Moreover, our modelling shows that CA3 recurrence also captures key behavioural features commonly observed in animals and humans, and that it generalises to different task conditions. This is in contrast with non-recurrent models, which failed to capture experimental observations. In addition, our work shows that agents trained in fully observable environments also do not capture experimental observations, thus suggesting the need to reevaluate previous experimental findings that may have implicitly assumed full observability.

Our work suggests that recurrent hippocampal networks underlie the ability of animals to learn to navigate environments with real-world conditions.

Results

We were inspired by a behavioural setup in which animals were trained on a Plus-maze (Fig. 1A) to perform a goal-driven navigational task while following two strategies, egocentric and allocentric [5]. In the egocentric (self-centred) rule, the reward was always positioned in the same location with respect to the animal, i.e. regardless of the north or south starting location. For the allocentric (world-centred) rule, the reward is at the same location irrespective of the animal's starting position, and the animal needs to turn left or right depending on whether they start from a north or south position. Training was conducted in a block-wise fashion with interleaved blocks of allocentric and egocentric tasks, each one with sub-blocks corresponding to different starting locations (i.e., north versus south). Despite the relatively complex nature of these tasks with multiple rules, rats achieve a good performance (Fig. 1B).

Next, we aimed to study the underlying architectural principles that enable such goal-driven navigation. To this end, we contrast animal behaviour and (hippocampal) neural data with artificial reinforcement learning (RL) agents with a hippocampal-like architecture. To mimic the experimental setup described above, we simulated a 2D minigrid environment [21], which consists of a starting state and two terminal states: rewarded and non-rewarded (Fig. 1C). To capture both north and south starting states, we use different sensory cues (Fig. 1D). Together, this task setup results in four sub-tasks (or rules): allocentric north and south, and egocentric north and south (illustrated in Fig. 1E).

Agents with CA3-like recurrence learn ego-allocentric goal-driven tasks

Our hippocampal RL models are based on standard deep reinforcement learning models, specifically, Deep Q-networks (DQN) [22] as outlined in the Methods section. These models feature a three-layer hippocampal-like structure: the input layer emulates entorhinal input to the Dentate Gyrus (DG), the first layer represents CA3, the second CA1, and the output layer encodes action-state values, denoted as $Q(a, s)$ (Fig. 2A,B). The entorhinal cortex (EC) is known to supply the hippocampus with a spatial map of the environment [19], which we approximate using the 2D top-down spatial map from the minigrid environment within our model (Fig. 1C). Additionally, the output layer captures state-action Q values, serving as an abstraction of hippocampal-to-striatum functional connectivity [23].

Motivated by the existence of recurrent connectivity in CA3 [24] and in line with previous work in which brain areas are modelled as gated recurrent neural networks [25, 26] we model CA3 using a Gated Recurrent Unit (GRU)

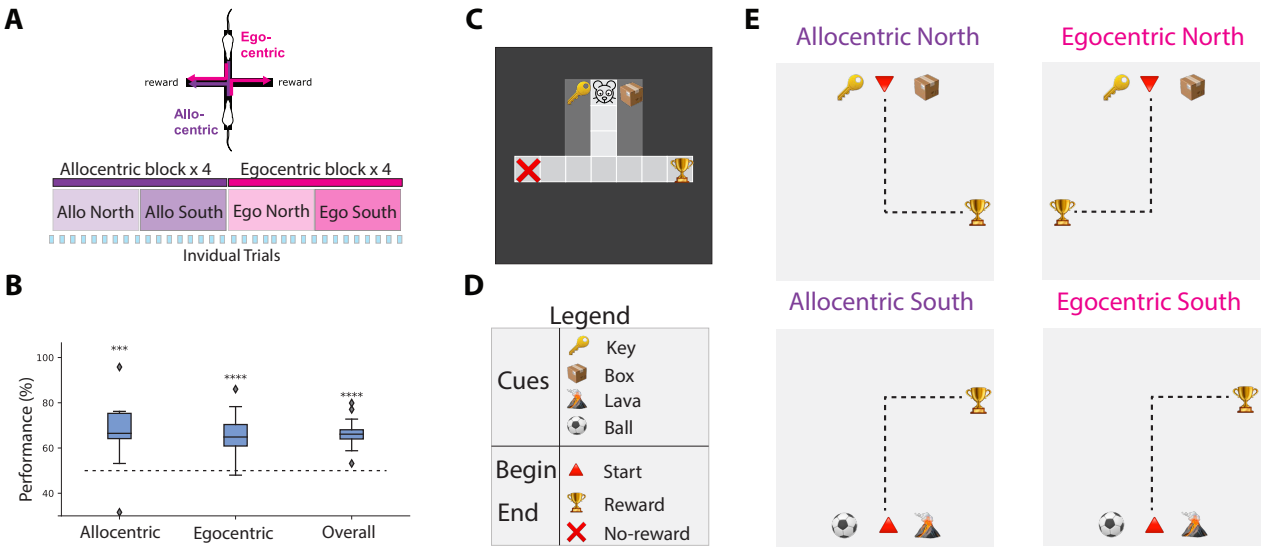


Figure 1. Ego-allocentric task setup in animals and reinforcement learning agents. (**A**) Top, experimental setup in which rats were placed in a plus-shaped maze [5]. The task consists of reaching the reward at the end of one of two arms following either egocentric (pink) or allocentric (purple) rules. Bottom: both animal and artificial agents were trained by interleaving blocks of allocentric and egocentric (see main text). (**B**) Animal performance on allocentric and egocentric tasks following the setup shown in (A) across 5 animals. (**C**) The experimental setup in (A) was simulated using a grid world environment. This setup was then used to train reinforcement learning agents in ego- and allocentric tasks. Environment observability was modelled by defining a visible range around the animal (light gray box; see main text for details), which is limited to the current cell alone when the agent enters the terminal arms. A total of four cues (cf. (D)) are placed in the environment, two for north starting state and two for the south starting state. Trophy and red cross represent rewarding and non-rewarding terminal states, respectively (not made visible to the agent). (**D,E**) Schematic showing the cues used (D) and the four possible allocentric and egocentric rules (E). Dotted line represents the ideal path towards the reward starting north/south. Cues are presented near the starting positions where key/box refer to the north starting point and lava/ball to the south one. In the allocentric task, the reward is always on the same side regardless of the starting position. In egocentric task, the reward is always on the right side of the starting position.

network [27], which we denote as hippocampal deep recurrent Q-Network (hcDRQN). In addition, we contrast this network with three other networks: a purely feedforward hippocampal Deep Q-Network (hcDQN) and two hcDQNs augmented with artificial continual learning algorithms. We considered two continual learning models to contrast our results with modern deep learning solutions to similar multi-task learning problems. In particular, we included two of the most popular methods: Elastic Weight Consolidation (ML-EWC; [28]) and Synaptic Intelligence (ML-SI; [29]). Motivated by the lack of evidence suggesting that replay of previous memories from the hippocampus to itself, we did not use the experience replay buffer in our model. One output Q-value head is not sufficient to solve all the tasks we consider (see Fig. S1A), even in the presence of a replay buffer. Therefore, to ensure that the network could solve the two tasks (ego and allocentric) we use task-specific heads at the output (see Methods), which in biological networks could be implemented through task-switching contextual signals [30].

First, we compare the hippocampal deep reinforcement learning models (Fig. 2B) with animals by contrasting their task performance (Fig. 1B). We trained the models using the grid environment described above and following a similar training procedure (trial-by-trial) used to train animals with blocks of allocentric trials alternated with blocks of egocentric trials (Fig. 1A, bottom). Within each block we alternate the two starting (north/south) positions. Our results show that the hippocampal-like network, hcDRQN, can successfully learn multiple tasks (Fig. 2D). The hcDRQN model not only yields the best performance on both tasks but is also the only model that can learn allocentric tasks while other models perform around chance level. This is because models that fail to truly learn the tasks will default to memorising to always turn right at the decision point as this behaviour will work 3 of the 4 sub-tasks (allo-south, ego-north and ego-south; see more details in Fig. 4). This is in line with the performance of the animals, showing that animals can learn both strategies (Fig. 1B). In addition, our results show that a non-plastic recurrent CA3 is not sufficient to learn all tasks (fixed hcDRQN model). Studying how the performance evolves over trials within each allocentric and egocentric block, shows that allocentric performance drops for hcDQN after each switch between

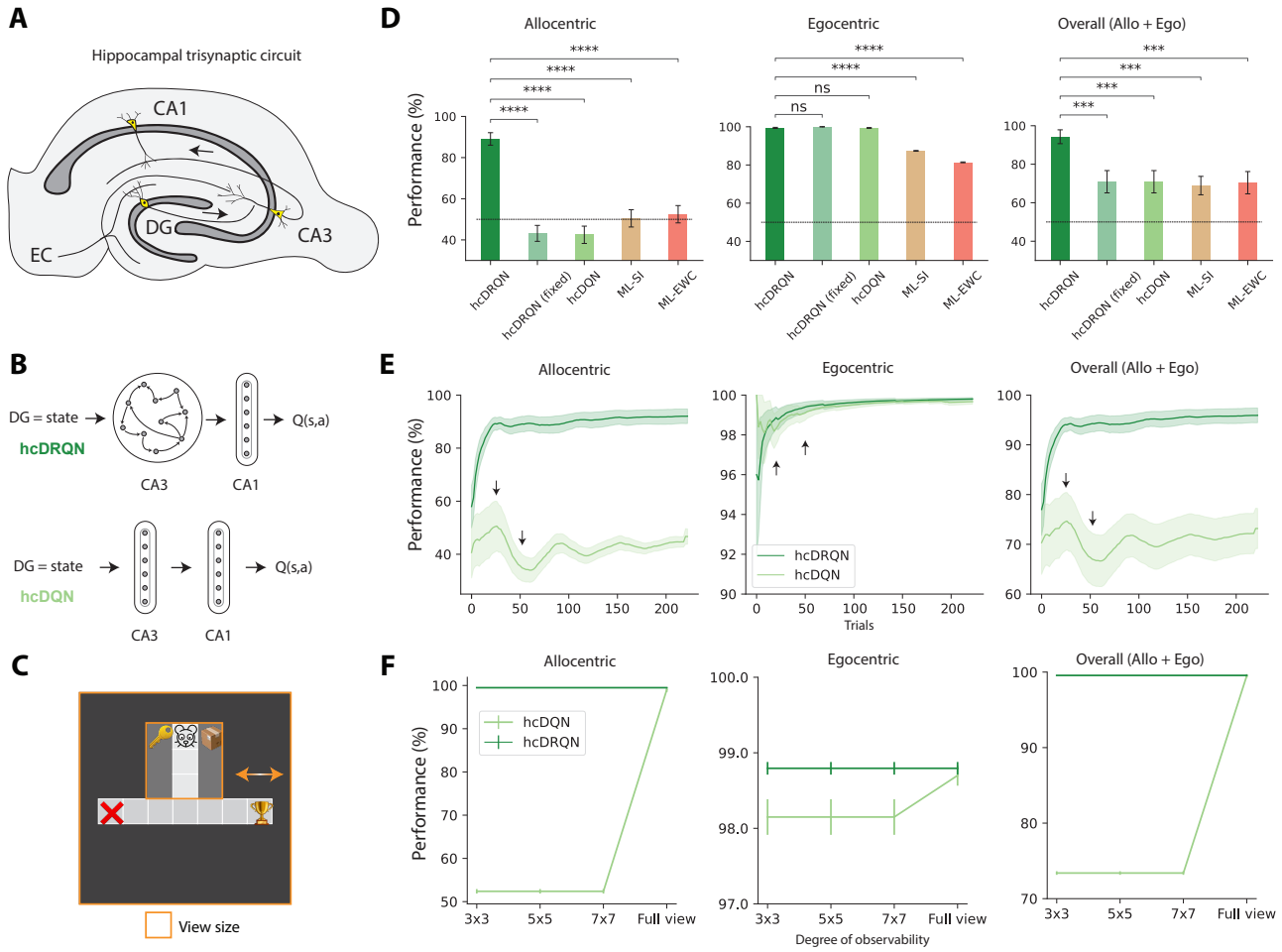


Figure 2. Reinforcement learning agents with CA3 recurrence jointly learn ego and allocentric tasks. (**A**) Classical hippocampal trisynaptic circuitry: entorhinal cortex (EC), dentate gyrus (DG), and hippocampus CA3 and CA1 layers. (**B**) Schematics of reinforcement learning (RL) agents with hippocampal-like architecture modelled as deep-Q-networks (DQN) used to learn the goal-driven tasks described in Fig. 1. In our models the DG receives a simplified (partially observable) map of the environment which is processed by the CA3-CA1 pathway and then CA1 projects to the reward system to compute the Q-value of state-action pairs, Q(s,a). We consider two main models: (i) with CA3 recurrence (hcDRQN, top) or (ii) with CA3 as a feedforward network (hcDQN, bottom). Both models consist of two hidden layers (CA3 and CA1). (**C**) Minigrid environment showing 3x3 and full view size (orange outline). (**D**) Performance of all models for allocentric (left), egocentric (middle) or both (right) tasks. For comparison with modern machine learning solutions to multi-task learning we also consider two popular algorithms: elastic weight consolidation (ML-EWC) and synaptic intelligence (ML-SI). (**E**) Learning curves for both hcDQN and hcDRQN, showing that the former fails to learn allocentric tasks. Arrows represent switching points. (**F**) Task performance of RL agents as environment observability is progressively incremented. Both models achieve the same performance under full observability whereas only the hcDRQN agent can learn tasks under non-full observability. Error bars represent standard error of the mean over 5 different initial conditions.

north vs. south scenarios (Fig. 2E). Although we report only hcDQN compared to hcDRQN, the other two models (ML-SI and ML-EWC) present the same behaviours as hcDQN and cannot solve allocentric tasks (see Fig. S1B).

CA3 recurrence is needed in partially observable environments

Next, we aimed to show that CA3 recurrence is indeed required for partial, but not full environmental observability. To demonstrate this, we tested hcDRQN and hcDQN in environments with different degrees of visibility (3x3, 5x5, 7x7, and full view; Fig. 2C). We expected a model without CA3 recurrence (i.e. DQN) to be able to solve all tasks in environments with full observability (i.e. all information continuously available). Our results show that the non-recurrent model, hcDQN, only succeeds to learn both allo and egocentric tasks when the full view is provided (Fig. 2F). We expected that models learn to solve the task in these conditions by continuously rely on having access to the

task-specific cues. To test this continuous reliance on sensory cues we removed the cues after the decision point (Fig. S2). Our results show that both models completely fail to complete the tasks.

Given that in most realistic environments animals are unlikely to have continuous access to the full environment our results suggest that CA3 recurrence plays an important role in supporting goal-driven behaviours under naturalistic conditions.

Task-relevant neuronal dynamics in agents and animals

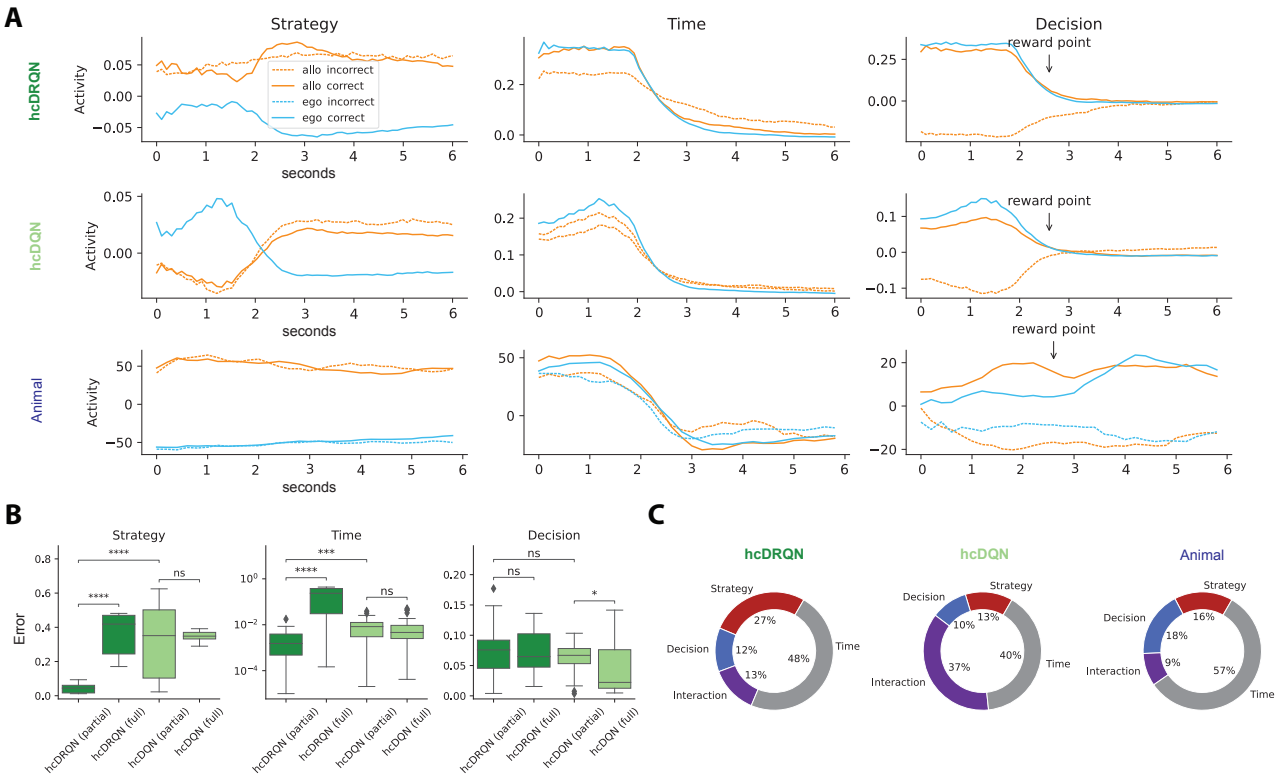


Figure 3. Strategy, temporal, and outcome neural dynamics in RL agents and animals. (**A**) Demixed principal components corresponding to task strategy (allocentric and egocentric for both north and south start locations), time and decision (correct and incorrect). hcDRQN components show separate task strategies whereas hcDQN mixes task strategies. (**B**) Mean squared error between normalised model and animal data demixed components. We also contrast agents trained with full and partial observability (cf. Fig. S3). (**C**) Percentage of explained variance for Decision, Strategy, Interaction and Time components (cf. full components in Fig. S4).

To contrast the neural dynamics predicted by the hippocampal RL agents with experimental observations recorded from awake performing animals we performed dimensionality reduction on CA1 recordings. In particular, we used demixed PCA (dPCA), which enabled us to extract behaviourally-relevant dimensions.

We extracted task-specific neural encodings from the agent throughout learning. The neural activity of the modelled CA1 layer of the agents was stored throughout learning and used to perform dPCA (see Methods for details). This analysis revealed three task-encoding components of interest in the hcDRQN agent (Fig. 3A). First, we find a clear separation at the population level between allo and egocentric tasks for hcDRQN, but not hcDQN (Fig. 3 left). The fact that hcDQN shows a mixed strategy component is consistent with the fact that it cannot learn both allo and egocentric strategies. Next, we find that both hcDRQN and hcDQN exhibit a temporally decaying population dynamics (Fig. 3A middle). Finally, we observe decision- or outcome-specific components predicting that CA1 encodes reward prediction errors well before reaching the decision point.

Next, we tested the predictions generated by our RL agents using tetrode recordings obtained from 612 CA1 neurons. The results of the dPCA show that the data qualitatively validate the results predicted by the hcDRQN agent for the strategy-specific components, but not the hcDQN agent (Fig. 3A bottom). We also observe stronger

neuronal activations in the allocenctric tasks compared to the egocentric tasks, in line with experimental observations (Fig. S5). To better quantify model-data match we used a normalised mean-squared error metric (see Methods). This metric shows that, indeed, hcDRQN better captures experimentally observed strategy neural dynamics (Fig. 3B left). For the time-specific components we observe a decaying component as predicted by the RL agents. Although both the hcDRQN and hcDQN look qualitatively very similar, the error metric shows that hcDRQN provides a better match with the data (Fig. 3B middle). Finally, the decision component also reveals a separation between correct and incorrect trials as predicted by the models, but in contrast to the models this separation remains after the reward point. However, we should point out that reward point in experimental data simply means that a sensor close to the reward was triggered, thus there is likely some delay between triggering the sensor and actually perceiving reward. As before we used the model-data error metric on the decision components (up to the reward point) and found no differences between the hcDQN and hcDRQN (Fig. 3B right).

Models trained under full observability conditions do not appear to provide a good match with experimental observations. To further support this point, we compared model neural dynamics in agents trained with full continuous access to the environment (Fig. S3). Our error metric shows that agents trained with full observability provide a poor match to neural dynamics when compared to agents trained under partial observability (Fig. 3B). These results provide further support for CA3 recurrence as being important to navigate environments under naturalistic conditions.

Finally, we contrast the degree of explained variance across models and data. hcDRQN captures explained variance across behavioural variables in a way that more closely matches awake tetrode recordings (Fig. 3C). Of particular interest is the fact that hcDQN relies more on mixed (or interaction) components (37%) compared to hcDQN (13%) and animal (9%), which is in line with its inability to fully solve all the tasks.

In summary, our neural dynamics analysis suggests that the hippocampus is indeed involved in task strategy, temporal integration and reward-based decision, in line with the predictions made by hcDRQN RL agents.

Hippocampal RL agents with recurrence capture animal behaviour

In order to contrast the behaviour of RL agents with that of animals we studied the trajectories taken during the tasks after learning. We studied the trajectories made by both RL agents and animals (Fig. 4A). To enable a comparison with agent trajectories we discretised animal trajectories into a 9x9 grid. The behavioural trajectories show that hcDRQN better captures animal behaviour in terms of time spent at the starting point, decision and the terminal state (Fig. 4A). On the other hand, the hcDQN agent fails to discriminate between the two allocentric tasks and instead learns only one policy (allocentric south). Next, to quantify the time spent on each state we calculated the ratio between individual states and the final state. This state-to-end ratio shows that hcDRQN better approximates animal behaviour also at a finer level and for allocentric strategies in particular (Fig. 4B,C). Next, to study whether the better fit of hcDRQN to animal behaviour is specific or general we made this analysis across all possible subtasks (Fig. 4D). Our results show that hcDRQN clearly outperforms hcDQN, except for a minor effect on the allocentric north when compared to allocentric south. When analysing RL agents trained with full observability we observed more mixed outcomes, suggesting that also for behavioural data partial observability better aligns with our results above (Fig. S6, S7).

Overall, hcDRQN provides a better match to animal behaviour, further supporting an important role of CA3 recurrence in the hippocampal circuitry.

Agent's behaviour predicts state-dependent action values

Because the recurrent RL agent is able to solve both ego and allocentric tasks we expected this to result in state-action value predictions that are generally more uncertain when compared to the non-recurrent model. To examine this in more detail we analysed the action-values for each state. This highlights the sequence of actions that makes hcDQN take the wrong arm and the correct policy learnt by hcDRQN for both allocentric tasks. Interestingly, on average, hcDRQN has higher Q-values than hcDQN, which reflects the fact that it learns all tasks (Fig. 5B). Next, we studied action selection certainty by calculating the Q-value variance across all possible Q-values for a given state (Fig. 5B). This analysis shows that hcDRQN starts with lower action certainty but that it gradually increases over states until the terminal state. This reflects the effect of appropriate cue integration towards a decision. In contrast, hcDQN becomes less certain after the initial state. Interestingly, the difference in terms of certainty between hcDRQN and hcDQN becomes even stronger when agents are trained under full observability (Fig. S8). The fact that the model that can solve all tasks (hcDRQN) is initially less certain and becomes more certain about its choices is in line with classical

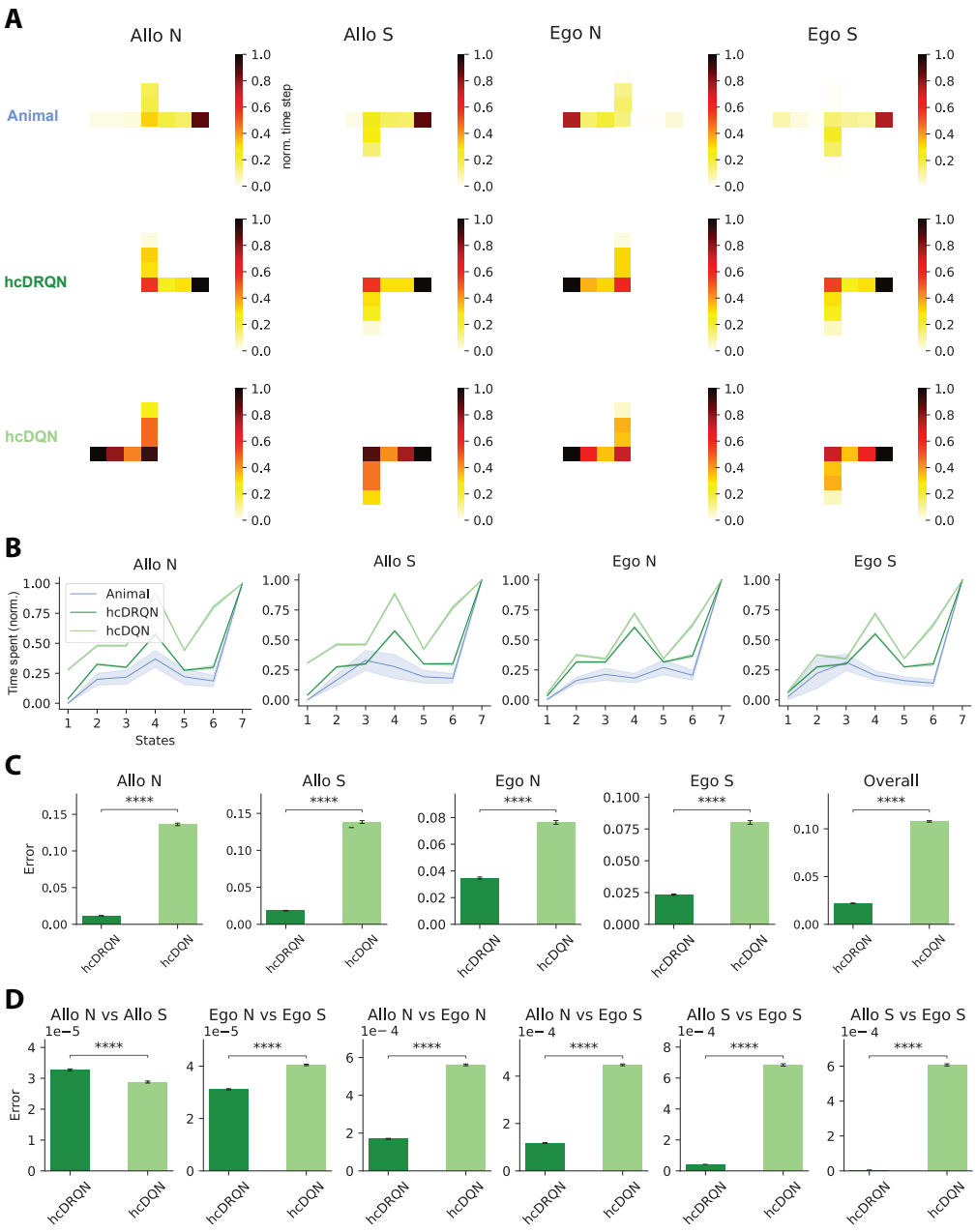


Figure 4. Hippocampal RL agents with recurrence capture animal behaviour. (**A**) Time spent on each maze state across the four strategies for: rats, hcDRQN and hcDQN. Animals spend more time at the decision and rewarded terminal points. hcDRQN better captures animal behaviour and hcDQN fails to solve allocentric north task (cf. Fig. 2). (**B**) Time spent on each state normalised to the time spent on the final (terminal) state in models and animal. (**C**) Error between a given model and animal behavioural data. (**D**) Error between a given model and animal behavioural data across all possible task-pairs. hcDRQN shows overall closer match to animal behaviour when compared to hcDQN. Error bars represent standard error of the mean over 5 different initial conditions.

animal and human behavioural observations [31]. In the decision making literature expert subjects are often less certain then naive subjects, which is related to the Dunning–Kruger effect.

Recurrence enables generalisation to stochastic environments

Until now we have trained RL agents in environments in which cues are always present. However, recurrent neural networks are well placed to deal with stochastic environments by integrating evidence over time [32, 33]. To test the effect of stochasticity on the different agents we created environments in which cues randomly appear and disappear

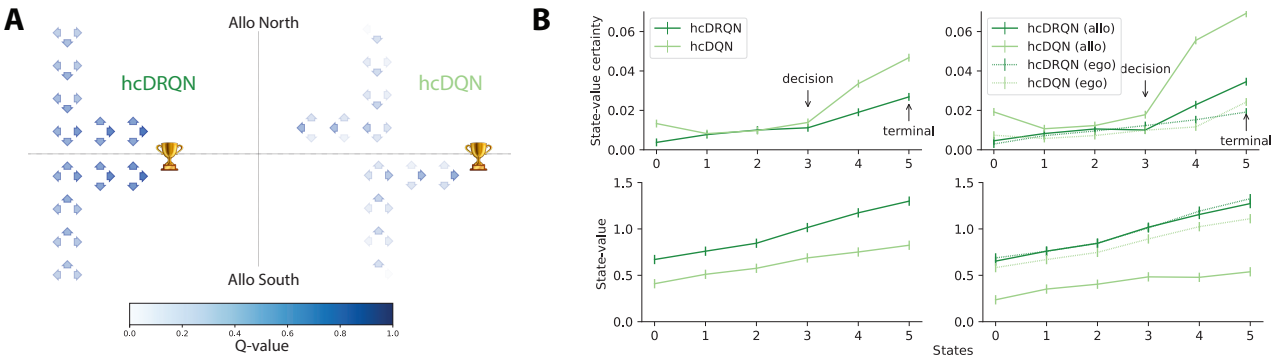


Figure 5. Agent's behaviour predicts state-dependent action values. (**A**) Schematic illustrating state-action policy for hcDRQN and hcDQN for allocentric north subtask. It highlights that hcDRQN solves the task and that it is more uncertain about which action (i.e. different actions have similar values) until the decision point. Each state is represented by 3 coloured arrows corresponding to the state Q-values where darker colour means higher value. (**B**) Top: The state-value certainty given by the variance over Q-values for each state shows that hcDRQN increases its certainty as it gets closer to the decision state and terminal state (see arrows). Bottom: Average state-values over environmental states. Error bars represent standard error of the mean over 5 different initial conditions.

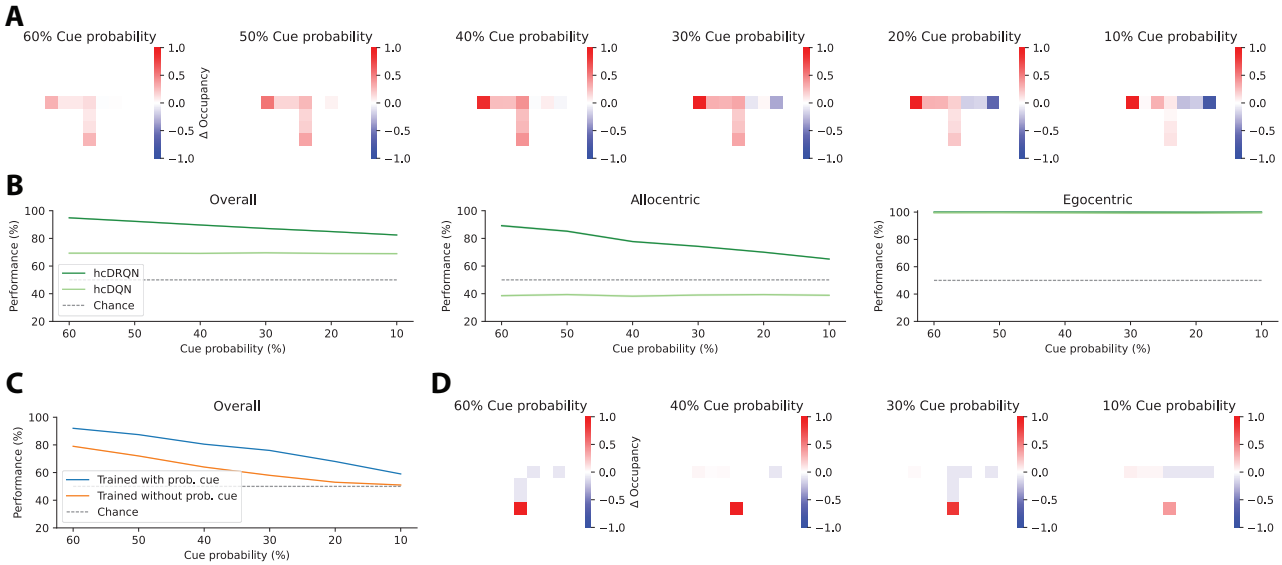


Figure 6. Recurrence enables better generalisability to stochastic environments. (**A**) Change in state-occupancy (with probabilistic cues - without probabilistic cues) for the hcDRQN agent across different degrees of cue removal. (**B**) hcDRQN outperforms hcDQN across different degrees of probabilistic cues. (**C**) Performance of hcDRQN agents trained with probabilistic cues compared to without. (**D**) Change in state-occupancy of hcDRQN agents trained with probabilistic cues.

(Fig. 6). We study two scenarios: (i) incremental random cue removal during inference (i.e. after learning) and (ii) effect random cue removal on learning. During inference, the performance of hcDRQN gradually decreases as the likelihood of cue removal increases. In contrast, for hcDQN, which lacks recurrence, its performance decreases as soon as the cues are removed, regardless of the degree of removal. This highlights the lack of evidence integration of hcDQN, while hcDRQN can handle a high degree of removal 90% while maintaining performance above 80%. When comparing the hcDRQN trajectories to the model without cue removal, the agent switches to the default 'turn left' policy when little or no cues are present. In addition, the model spends more time in the start and middle corridors because it must observe cues before deciding which arm to turn onto. Next, we tested the idea that if the agent was trained in a stochastic environment, this should result in the model being more robust to cue removal. Indeed, when trained under these conditions the model performs consistently better than a model trained without random cue removal (10% improvement). When analysing trajectory behaviour, our model predicts that agents spend more time on the starting location to integrate sensory evidence for longer before committing to a decision.

Taken together, these results suggest that CA3 recurrence also plays an important role in learning to navigating stochastic environments.

hcDRQN generalises to different task conditions

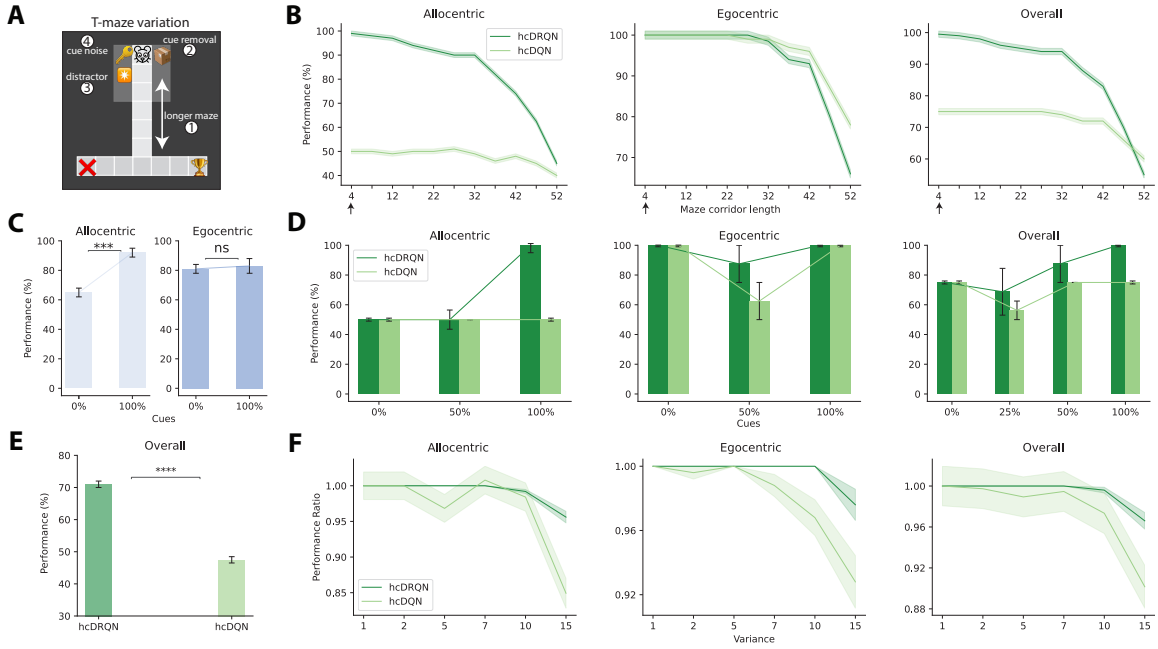


Figure 7. hcDRQN shows better generalisation to maze length, cue removal, distractors and sensory noise. (**A**) T-maze setup for increased length of the middle corridor, cue removal and random noise. (**B**) Performance decrease over gradual increase of the maze length shows that hcDRQN can handle middle maze length being 32 steps. Black arrows on the X-axis represent maze corridor length of 4 steps utilised during training. (**C,D**) Animal and model performance when cues are removed from the environment. Both hcDRQN (light green) and animal (blue) allocentric navigation are highly dependent on cues while egocentric is not affected by cue removal. On the other hand, hcDQN fails to solve both allocentric tasks. (**E**) When adding a distractor cue, hcDQN drops to chance level while hcDRQN can still solve most of the tasks. (**F**) When adding white Gaussian noise to the cues hcDRQN is more stable and robust when compared to hcDQN. Error bars represent standard error of the mean over 5 different initial conditions.

Finally, we tested whether the RL agents considered here can generalise to different task conditions not experienced during training (Fig. 7A). First, we tested different lengths of the initial maze corridor (Fig. 7B). This allowed us to test whether the RL agents memorise the tasks or learn to integrate the cue information and maintain it in memory to trigger the right action at the decision point. Our results show that hcDRQN performance is very robust across a large range of lengths, whereas hcDQN defaults to chance level. This demonstrates that indeed the hcDRQN has successfully learned to integrate cue information, which is then maintained in its recurrent memory for action selection when required.

Next, we tested the models on the same environment it was trained on, but removing a set of cues at a time. Note that this is complete removal of cues, rather than stochastic cues as in the previous section. When retaining all cues the performance obtained by all models is in line with the training performance, with hcDRQN being the best model and the only one doing better than chance in the allocentric tasks (Fig. 7D, all cues). This demonstrates that the models were able to remember all the tasks on which they were trained. To test for generalisation, we gradually reduced the number of cues available in the environment. Our results show that hcDRQN is the only model that can handle half of the cues being removed. Interestingly, cue removal is more detrimental to allocentric navigation than egocentric navigation. This result is in line with experimental observations, in which allocentric but not egocentric task performance is impaired upon cue removal [34](Fig. 7C). In contrast, models trained with full observability cannot generalise, as they rely on the presence of specific cues (Fig. S9).

Finally, we repeated the original task on the T-maze adding a distractor cue and adding (Gaussian) white noise to the cues. When a distractor cue is introduced, hcDQN's performance drops to chance level, whereas hcDRQN

achieves a success rate of approximately 70% (Fig. 7E). To test the robustness of both models to noise we tested a range of noise levels (Fig. 7F). The hcDRQN model can handle a relatively large degree of cue noise without any changes in the overall performance, while hcDQN is more unstable and shows a faster decrease in performance as the noise is increased.

Taken together, our generalisation tests demonstrate that hcDRQN generalises better to different and realistic task conditions, in line with animal behaviour.

Discussion

Naturalistic behaviour almost always relies on navigating environments with limited visibility. Here, we have shown that recurrent hippocampal networks play a pivotal role in such environmental setups. Our investigation began by training RL agents to perform ego-allocentric tasks within partially observable environments. Remarkably, agents equipped with recurrent hippocampal circuitry successfully mastered these tasks, mirroring real-world animal behaviour. Additionally, our models predicted reward, strategy, and temporal neuronal representations, which we validated through extensive hippocampal neuronal recordings. Furthermore, hippocampal-like RL agents predicted state-specific trajectories and action uncertainty, closely resembling experimental observations. In stark contrast, agents trained in fully observable environments failed to replicate the experimental data. Most importantly, these hippocampal-like RL agents demonstrated enhanced generalisation capabilities across novel task conditions.

Motivated by the challenging conditions that animals often face in the wild, we have focused on a task setup with partial observability. This is also supported by the lack of visual acuity in rodents [35, 36]. In addition, when our models were trained with full observability, they could not generalise (Fig. S9), which further suggests that partial observability provides a better model of animal behaviour. hcDRQN performs particularly well in partial environments, in line with previous research in artificial neural networks [37]. Partially observable environments represent a more real-world setup, which suggests that hippocampal CA3 region may have evolved to support the ability to navigate partially observable environments.

Classical hippocampal models suggest that CA3 recurrency enables pattern completion [38, 39], while more recent computational models propose that the hippocampus creates a predictive map of the environment through successor representations (SR) [40]. Our research aligns with the SR view and reveals task-specific reward prediction traces (see Fig. 3). Furthermore, our findings underscore the essential role of CA3 in constructing the hippocampal predictive map, consistent with the predictive view of hippocampal function [40]. In another set of studies recurrent neural networks (RNNs) have been trained to support spatial navigation. They have shown that RNNs develop spatial receptive fields similar to experimental findings [14–16]. For instance, Cueva and Wei [15] demonstrated grid-like spatial response patterns, border cells, and band-like cells in trained RNNs. Similarly, Banino et al. [14] revealed grid cell-like representations when training deep recurrent reinforcement learning networks for 3D navigation. Uria et al. [16] trained a similar system, yielding neurons with spatial receptive fields akin to those in Banino et al. [14], Cueva and Wei [15]. While these studies emphasize the importance of recurrent connectivity in hippocampal networks, they do not assess their function in partial observability and its relationship to experimental observations, which is a focus of our work.

Our results show that a model with a recurrent layer (hcDRQN) without experience replay outperforms alternative methods that were specifically designed for continual learning (even when a multi-head setting is considered), consistent with recent machine learning findings [41]. Given that hcDRQN is a systems-level approximation of the hippocampal system, it suggests that the brain relies on a combination of recurrent neural networks to continually adapt to new situations, at least in navigational tasks. It remains to be tested how general these principles are across other areas of the brain. Our models do not use a memory buffer that retains all previous experiences. However, recent work has introduced generative replay models [42, 43] which circumvent the problem of storing all previous experiences. In the future, it would be of interest to explore these variants.

Because our focus is on contrasting models with neuroscientific observations, we have used the same continually interleaving ego- and allocentric tasks as employed experimentally. Interestingly, when tested on egocentric tasks our models are more robust to the removal of cues when compared to allocentric tasks (Fig. 7C), consistent with experimental observations [34]. However, the tasks that we have used here represent only a small subset of all the possible challenges that animals are only faced with. This means that our model-animal comparison is relatively

unfair as animals have to deal with much more than solving these two tasks. Conversely, it is also true that there are many more biological principles that we have not considered in our models. All these elements remain to be explored in future work.

Our model shows that CA3 recurrence is needed to solve all the tasks we tested due to its ability to remember the relevant sensory cues. This in line with experimental results showing that the CA3 region is involved in maintaining working memory representation for delayed-to-match sample tasks [9–11]. Moreover, our work shows that the hippocampus encodes reward prediction error signals. This mirrors the growing evidence suggesting that the hippocampus interacts with the reward-system [44, 45].

Overall, our work suggests that hippocampal networks play a critical role in the ability of animals to continuously adapt to the environment under realistic conditions and with good generalisation properties.

Acknowledgements

We would like to thank the Neural & Machine Learning group and Chris Summerfield for useful feedback. D.P. was funded by the EPSRC Centre for Doctoral Training in Future Autonomous and Robotic Systems (FARSCOPE), S.M. by the Wellcome Trust (PMAG/563) and BBSRC and R.P.C. by the Medical Research Council (MR/X006107/1), BBSRC (BB/X013340/1) and a ERC-UKRI Frontier Research Guarantee Grant (EP/Y027841/1). H.M. was funded by a FWF grant (I 5458; part of the German Research Foundation Research Unit 5159) and a WWTF grant (CS18-039). S.C. was funded by a European Research Council starting grant 716761 and a Swiss National Science Foundation professorship grant 170654. This work made use of the HPC system Blue Pebble at the University of Bristol, UK. We would like to thank Dr Stewart for a donation that supported the purchase of GPU nodes embedded in the Blue Pebble HPC system.

Author contributions

D.P. and S.M. co-developed the computational framework with guidance from R.P.C. D.P., S.M. and M.V.D. performed all RL simulations. S.M., D.P. and R.P.C. analysed the behavioural and neuronal data with contributions from H.M and S.C.. D.P., S.M. and R.P.C. wrote the manuscript, with contributions from H.M. and S.C. R.P.C supervised the project.

References

[1] E. C. Tolman. Cognitive maps in rats and men. *Psychological review*, 55(4):189, 1948.

[2] J. O'keefe and L. Nadel. *The hippocampus as a cognitive map*. Oxford: Clarendon Press, 1978.

[3] E. Tulving. Organization of memory. *Episodic and semantic memory*, 1972.

[4] A. M. Wikenheiser and A. D. Redish. Hippocampal place-cell sequences depict future paths to remembered goals. *Nature*, 497 (7447):74–79, 2013. doi: 10.1038/nature12112.

[5] S. Ciocchi, J. Passecker, H. Malagon-Vina, N. Mikus, and T. Klausberger. Brain computation. selective information routing by ventral hippocampal CA1 projection neurons. *Science*, 348(6234):560–563, May 2015.

[6] M. Sosa and L. M. Giocomo. Navigating for reward. *Nature Reviews Neuroscience*, 22(8):472–487, 2021.

[7] N. Nyberg, É. Duvelle, C. Barry, and H. J. Spiers. Spatial goal coding in the hippocampal formation. *Neuron*, 2022.

[8] M. G. Edelson and T. A. Hare. Goal-dependent hippocampal representations facilitate self-control. *Journal of Neuroscience*, 2023.

[9] I. Lee and R. P. Kesner. Differential contribution of nmda receptors in hippocampal subregions to spatial working memory. *Nature neuroscience*, 5(2):162–168, 2002.

[10] I. Lee and R. P. Kesner. Differential roles of dorsal hippocampal subregions in spatial working memory with short versus intermediate delay. *Behavioral neuroscience*, 117(5):1044, 2003.

[11] P. E. Gilbert and R. P. Kesner. The role of the dorsal ca3 hippocampal subregion in spatial working memory and pattern separation. *Behavioural brain research*, 169(1):142–149, 2006.

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

[13] E. T. Rolls. A quantitative theory of the functions of the hippocampal ca3 network in memory. *Frontiers in cellular neuroscience*, 7:98, 2013.

[14] A. Banino, C. Barry, B. Uria, C. Blundell, T. Lillicrap, P. Mirowski, A. Pritzel, M. J. Chadwick, T. Degris, J. Modayil, et al. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705):429–433, 2018.

[15] C. J. Cueva and X.-X. Wei. Emergence of grid-like representations by training recurrent neural networks to perform spatial localization. *arXiv preprint arXiv:1803.07770*, 2018.

[16] B. Uria, B. Ibarz, A. Banino, V. Zambaldi, D. Kumaran, D. Hassabis, C. Barry, and C. Blundell. The spatial memory pipeline: a model of egocentric to allocentric understanding in mammalian brains. *bioRxiv*, 2020.

[17] J. C. Whittington, T. H. Muller, S. Mark, G. Chen, C. Barry, N. Burgess, and T. E. Behrens. The tolman-eichenbaum machine: unifying space and relational memory through generalization in the hippocampal formation. *Cell*, 183(5):1249–1263, 2020.

[18] J. O'Keefe and J. Dostrovsky. The hippocampus as a spatial map: preliminary evidence from unit activity in the freely-moving rat. *Brain research*, 1971.

[19] T. Hafting, M. Fyhn, S. Molden, M.-B. Moser, and E. I. Moser. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–806, 2005.

[20] E. B. Knudsen and J. D. Wallis. Hippocampal neurons construct a map of an abstract value space. *Cell*, 184(18):4640–4650, 2021.

[21] M. Chevalier-Boisvert, L. Willems, and S. Pal. Minimalistic gridworld environment for openai gym. https://github.com/maximecb/gym-minigrid, 2018.

[22] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[23] B. D. Devan and N. M. White. Parallel information processing in the dorsal striatum: relation to hippocampal function. *Journal of neuroscience*, 19(7):2789–2798, 1999.

[24] E. Cherubini and R. M. Miles. The ca3 region of the hippocampus: how is it? what is it for? how does it do it? *Frontiers in cellular neuroscience*, 9:19, 2015.

[25] R. P. Costa, I. A. Assael, B. Shillingford, N. de Freitas, and T. Vogels. Cortical microcircuits as gated-recurrent neural networks. In *Advances in neural information processing systems*, pages 272–283, 2017.

[26] J. X. Wang, Z. Kurth-Nelson, D. Kumaran, D. Tirumala, H. Soyer, J. Z. Leibo, D. Hassabis, and M. Botvinick. Prefrontal cortex as a meta-reinforcement learning system. *Nature neuroscience*, 21(6):860–868, 2018.

[27] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[28] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. U. S. A.*, 114(13):3521–3526, March 2017.

[29] F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. March 2017.

[30] K. V. Kuchibhotla, J. V. Gill, G. W. Lindsay, E. S. Papadoyannis, R. E. Field, T. A. H. Sten, K. D. Miller, and R. C. Froemke. Parallel processing by cortical inhibition enables context-dependent behavior. *Nature neuroscience*, 20(1):62–71, 2017.

[31] M. Motta, T. Callaghan, and S. Sylvester. Knowing less but presuming more: Dunning-kruger effects and the endorsement of anti-vaccine policy attitudes. *Social Science & Medicine*, 211:274–281, 2018.

[32] W. Singer. Recurrent dynamics in the cerebral cortex: Integration of sensory evidence with stored knowledge. *Proceedings of the National Academy of Sciences*, 118(33):e2101043118, 2021.

[33] J. Pemberton, P. Chadderton, and R. P. Costa. Cerebellar-driven cortical dynamics enable task acquisition, switching and consolidation. *bioRxiv*, pages 2022–11, 2022.

[34] H. Malagon-Vina, S. Ciocchi, J. Passecker, G. Dorffner, and T. Klausberger. Fluid network dynamics in the prefrontal cortex during multiple strategy switching. *Nat. Commun.*, 9(1):309, January 2018.

[35] A. Hughes. The topography of vision in mammals of contrasting life style: comparative optics and retinal organisation. In *The visual system in vertebrates*, pages 613–756. Springer, 1977.

[36] G. T. Prusky, P. W. West, and R. M. Douglas. Behavioral assessment of visual acuity in mice and rats. *Vision Res.*, 40(16):2201–2209, 2000.

[37] M. Hausknecht and P. Stone. Deep recurrent Q-Learning for partially observable MDPs. July 2015.

[38] S. Leutgeb and J. K. Leutgeb. Pattern separation, pattern completion, and new neuronal codes within a continuous ca3 map. *Learning & memory*, 14(11):745–757, 2007.

[39] E. Rolls. The mechanisms for pattern completion and pattern separation in the hippocampus. *Frontiers in systems neuroscience*, 7:74, 2013.

[40] K. L. Stachenfeld, M. M. Botvinick, and S. J. Gershman. The hippocampus as a predictive map. *Nature neuroscience*, 20(11):1643, 2017.

[41] B. Ehret, C. Henning, M. R. Cervera, A. Meulemans, J. von Oswald, and B. F. Grewe. Continual learning in recurrent neural networks. June 2020.

[42] H. Shin, J. K. Lee, J. Kim, and J. Kim. Continual learning with deep generative replay. May 2017.

[43] I. Stoianov, D. Maisto, and G. Pezzulo. The hippocampal formation as a hierarchical generative model supporting generative replay and continual learning. *bioRxiv*, 2020.

[44] E. T. Rolls and J.-Z. Xiang. Reward-spatial view representations and learning in the primate hippocampus. *Journal of Neuroscience*, 25(26):6167–6174, 2005.

[45] W. R. Du, E. Li, J. Guo, Y.-t. Chen, S. J. Oh, A. Samuel, Y. Li, H. K. Oyibo, and W. Xu. Hippocampus-striatum wiring diagram revealed by directed stepwise polysynaptic tracing. *bioRxiv*, pages 2021–10, 2021.

[46] D. Kobak, W. Brendel, C. Constantinidis, C. E. Feierstein, A. Kepecs, Z. F. Mainen, X.-L. Qi, R. Romo, N. Uchida, and C. K. Machens. Demixed principal component analysis of neural population data. *Elife*, 5, April 2016.

Methods

We begin by outlining the deep reinforcement learning approaches employed in this study, followed by an explanation of the methods utilised for the analysis of neural data.

Reinforcement learning models

We developed a deep reinforcement learning model consistent with the hippocampal architecture. To train the models we designed a custom-built 2D gym-minigrid maze [21], mimicking the T-maze environment (Fig. 1a) in line with common experimental setups, which allow us to compare our models with the behavioural and neural data [5]. In order to capture cues commonly placed on external walls in experimental setups we placed four cues in the environment in both allocentric and egocentric trials, in line with [5]. At the beginning of any given trial the agent was placed at the start of the north or south arms of the maze following the same setup of the animal experiments, and we closed access to the opposite arm thus converting the maze into a T-maze. We considered two terminal states: one rewarded and one unrewarded. After reaching a terminal state (rewarded/unrewarded) we allowed the agent to continue exploring for three extra time steps which allowed us to model the animal behaviour right after reward consumption. During model training we extracted neural activities which we used to contrasted model and animal neural data. Note that direct sensory information about the terminal states was not given as input to the agent.

Deep RL agents:

Reinforcement learning (RL) models an agent that observes the environment and takes an action a. This action transitions the agent into a new state s of the environment which might give back a reward r according to the utility of the action selected. This can be formally defined by a Markov Decision Process as tuple of $\langle \mathcal{S}, \mathcal{A}, P, \mathcal{R}, \gamma \rangle$ where $\mathcal{S}$ is the set of all the states, $\mathcal{A}$ is the action set, $\mathcal{P}$ the transition matrix $P(s'|s,a)$ from current state s to the next state s' when taking action a. The objective is to maximise the expected total rewards, called return G_t defined as $G_t = \sum_{k=t}^{T} \gamma^{k-t} R_{k+1}$ where t is the current time step, R_{t+1} is the reward obtained at time $t+1$, γ is a discount factor such that $0 \leq \gamma < 1$ and T is the time at which the episode terminates.

For the hippocampal RL models we build on the standard deep reinforcement learning models. In particular, we use Deep Q-networks (DQN) [22], in which states s are provided as input to an artificial neural network that are then mapped onto value-action pairs $Q(s,a)$. The network is trained using state-outcome transition tuples, (s, a, s', r), where s is the current state, a is the action, r denotes reward outcome and s' the next state. The error function used to train the hippocampal network follows a Q-update function as E_i at step i:

$$E_i(\theta_i = E_{s,a,s',r\sim D}(r + \gamma \max_{a'} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i))^2 \quad (1)$$

where θ denotes the network weights, γ is the discount factor and D is the dataset of past trajectories. As done by standard DQNs we use the concept of *target network* (θ^-) which helps to stabilise learning (see Methods).

To create a model that more closely captures the hippocampal circuitry, we consider a three-layered structure with the input layer modelling entorhinal input to Dentate Gyrus (DG), the first layer represents CA3, a second layer represents CA1, and the output encodes the value of a given action-state pair, $Q(a,s)$ (Fig. 2b). The DG input originates from the entorhinal cortex (EC) which is believed to provide the hippocampus with a spatial map of the environment [19]. In our model, the EC spatial map is approximated by the 2D top-down spatial map provided by the minigrid environment (Fig. 1c). The output layer encodes state-action Q values, which abstracts out hippocampal-to-striatum functional connectivity [23].

All our models have a four-layered structure in which the input layer is of shape (RxC) where R is the number of rows in the input grid, C the number of columns. The output layer has Nx1 shape ($N = 3$) denoting the Q-values for the 3 actions that the agent can take in the environment (left, right, forward). We use a standard discount factor ($\gamma = 0.9$), a memory buffer of size 1 and a batch size of 1 during training. Adam is used as the optimiser with a learning rate α of 0.001. We choose a CA3 layer size of 50 for hcDRQN as well as for all the other models considered. The learning rate and epsilon for epsilon greedy have been selected using a grid-search. All the hyper parameters are given in Table S1.

Name	Value
Discount factor, γ	0.9
Adam learning rate, α	0.001
Epsilon-greedy, ϵ, max-min	0.3 - 0.05
Epsilon-decay	0.9
Batch size	1
MLP layers	4
Input size	9/81
Hidden size	50
hcDRQN hidden size	50
Output size	3
Memory size	1
Target update counter	25

Supplementary Table S1. Hyperparameters used to run the experiments given in the paper.

Partial observability

In our grid-based environment, models operate under conditions of limited enviromental observability, mirroring the real-world challenges faced by navigating animals. Moreover, in reality, animals rarely possess complete access to all pertinent sensory data during navigation. For instance, they may initially focus on cue information but then shift their attention to executing motor commands to reach their destination. In experimental neuroscience, while cues are typically positioned along the outer walls of a room [5], animals do not continuously fixate on these cues. Additionally, maze setups often involve the incorporation of walls of varying heights, further restricting the visual input available to the animals.

To substantiate the importance of CA3 in navigation and its ability to better align with experimental findings in partially observable environments, we compare our models against those trained with full visibility. This comparison underscores the significance of CA3 and its capacity to more accurately capture experimental outcomes when navigating in environments with limited sensory input.

Training details

The training phase consisted of a block of allocentric and egocentric trials. Specifically, each block contains 25 trials, and there were a total of 4 blocks (allocentric north/south, egocentric north/south). The agents were first exposed to blocks of allocentric trials in the north direction, which were then alternated with blocks of allocentric trials in the south direction. This alternating pattern was repeated four times before switching to the egocentric trials. The same north/south combination was maintained throughout the entire duration of the egocentric trials. In total, the training consistented of 10,000 individual trials (200 blocks each with 25 trials for both ego and allocentric tasks).

A two-head setup is utilised, where the final layer outputs two Q-values: Q-value-allo and Q-value-ego. The state input to the models are 2D matrices where cues, walls and no-walls were encoded as scalar values. For the partial view the observation size was 9 (3x3) while for the full view was 81 (9x9).

Generalisation tests

We performed four types of generalisation tests (Fig. 7):

1. *Longer maze*: In this test, the length of the starting corridor was increased while keeping the length of the two terminal arms constant.
2. *Cue removal*: Different combinations of cue removal were performed, ranging from removing all cues to removing none. The cues were removed at the beginning of the trial, meaning that the agent had no access to the cue at any point during the task. This is in contrast to the experiments on probabilistic cue removal (Fig. 6), where the agent still had access to these cues with a given probability.
3. *Distractor*: Another cue (represented by a scalar value) was added just next to (below) the existing cues.
4. *Random noise*: We added normally distributed noise, $\mathcal{N}(\mu, \sigma^2)$ with $\mu = 0$ and σ^2 in the range of (1,15).

Continual learning algorithms

To compare with modern artificial algorithms capable of multi-task learning we tested Elastic Weight Consolidation (EWC) and Synaptic Intelligence (SI). Hyperparameters were selected through a hyperparameter search (the EWC weight importance was set to 800 and the SI weight importance to 30).

Experiments with fully observable environments

We repeated the main results with a fully observable environment with both hcDRQN and hcDQN models. Although both models can learn all tasks in terms of performance, their dPCA analysis does not show a clear separation of all the components. In hcDRQN, decision and strategy follow the same trends with activity dropping to zero right after the reward point. This is the opposite of the partial view hcDRQN and animal activity where strategy components keep the separation even after the reward point. Moreover, hcDRQN fails to capture the time component. The hcDQN model completely fails to separate the strategy north compoents. Overall, given that full view model fails to capture dPCA components we argue that the fully observable environment does not provide a good match of the hippocampal data. We run further tests to analyse the animal trajectories and the generalisation capabilities of these full view RL models. Although most of the trajectory maps show close match between the animal and hcDRQN, there are situations in which the hcDQN seems to be a better match to animal data. The generalisation tests highlight the limits of the fully observable models, as cue are gradually removed performance drops drastically, emphasising the dependency of these models on the cues. Animal performance and partial view RL models show evidence that egocentric task do not rely on cues, however the fully observable models remain highly depended on the cues. Overall our results suggest that hcDRQN trained with partial observability provides an overall best match with animal behaviour and neuronal encodings.

Computing details

All experiments were conducted on the BluePebble super computer at Bristol; mostly on GPUs (GeForce RTX 2080 Ti) and some on CPUs (Intel(R) Xeon(R) Silver 4112 CPU @ 2.60GHz). We did not record the total computing time for the experimental results presented in this paper, but this can be estimated as follows. To train each model (one seed with all the task-specific trials) takes approx 1 hour and 30 min. For each of the models we run 5 random seeds, resulting in approx 6 hours per model. When recording the activations, the total time is around 8 hours. Testing a single model for one seed takes approx 5 min. Overall total time it takes to run our models is 32 hours (8 x 4) for training with 5 seeds and 2 hours for the testing results with 5 seeds.

Statistical analysis

Due to the inherent variability of the starting conditions on the learning path of these models, we trained our models across 5 different randomly selected seeds. To assess the significance of all relevant figures, we conducted a two-sided paired t-test on the relative alterations across the various seeds. Significance levels are denoted as follows: * ($p < 0.05$), ** ($p < 0.01$), *** ($p < 0.001$), and **** ($p < 0.0001$).

Data and code availability

We used the PyTorch library for all reinforcement learning models. The code and respective simulated data used for our experiments is available at https://github.com/neuralml/hcRL.

Neural and behavioural experimental data

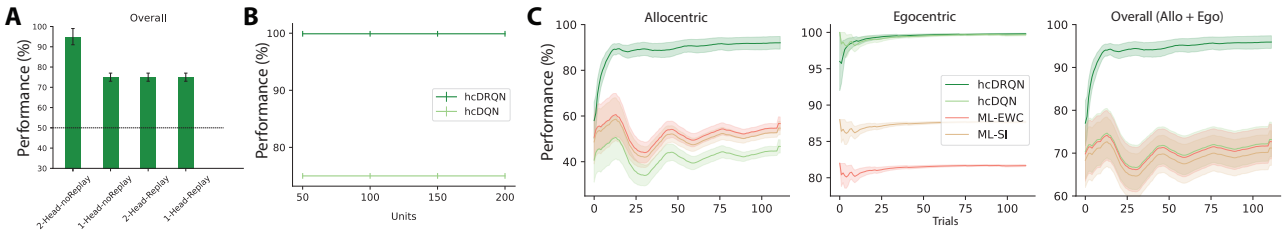
Neural data analysis using demixed PCA

We used the neural activities of 612 hippocampal CA1 neurons from five behaving rats were recorded, which were obtained in the dorsal and ventral CA1 using multiple tetrodes (Fig. 3a; see full details in Ciocchi et al. [5]). Spike sorting was used to assign spikes to different neurons (full details in [5]), which were then converted to firing rates of individual neurons using a sliding window. Animals were trained on a T-maze task in which rats had to follow both allocentric and egocentric navigational rules to reach reward points (Fig. 3a). We performed demixed Principal Component Analysis (dPCA) [46] on the neuronal firing rates using with 3 behavioural variables – trial decision, strategy and time (dPCA $\lambda = 2.919e^{-05}$ was found using grid-search as done by [46]). We used the dPCA code made available by the authors of [46] in https://github.com/machenslab/dPCA.

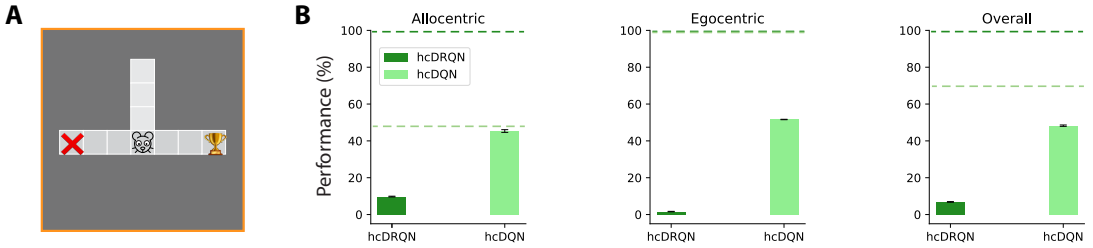
507 ## Behavioural data

508 The behavioural data (i.e. animal task performance) consists of a total of 47067 trials recorded over multiple days (3
509 to 7) from a total of 5 animals [5]. However, as some animals only had a maximum of 800 continuous trials we used
510 a maximum of 800 continual trials per animal.

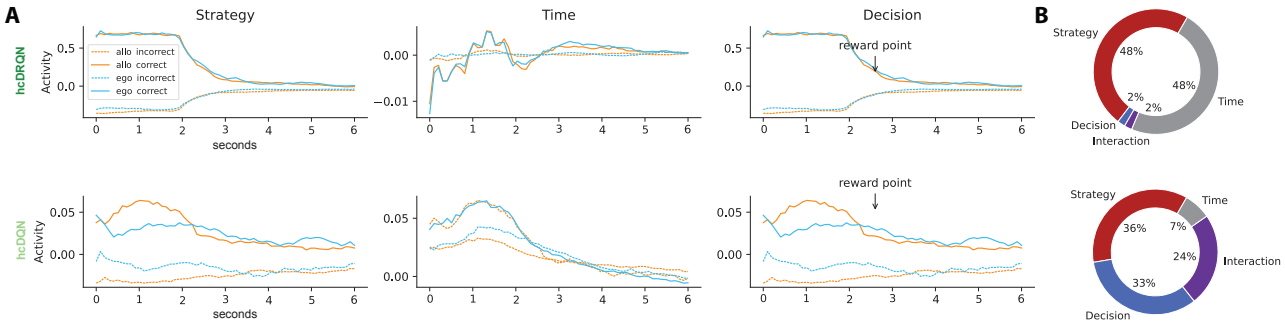
Supplementary information



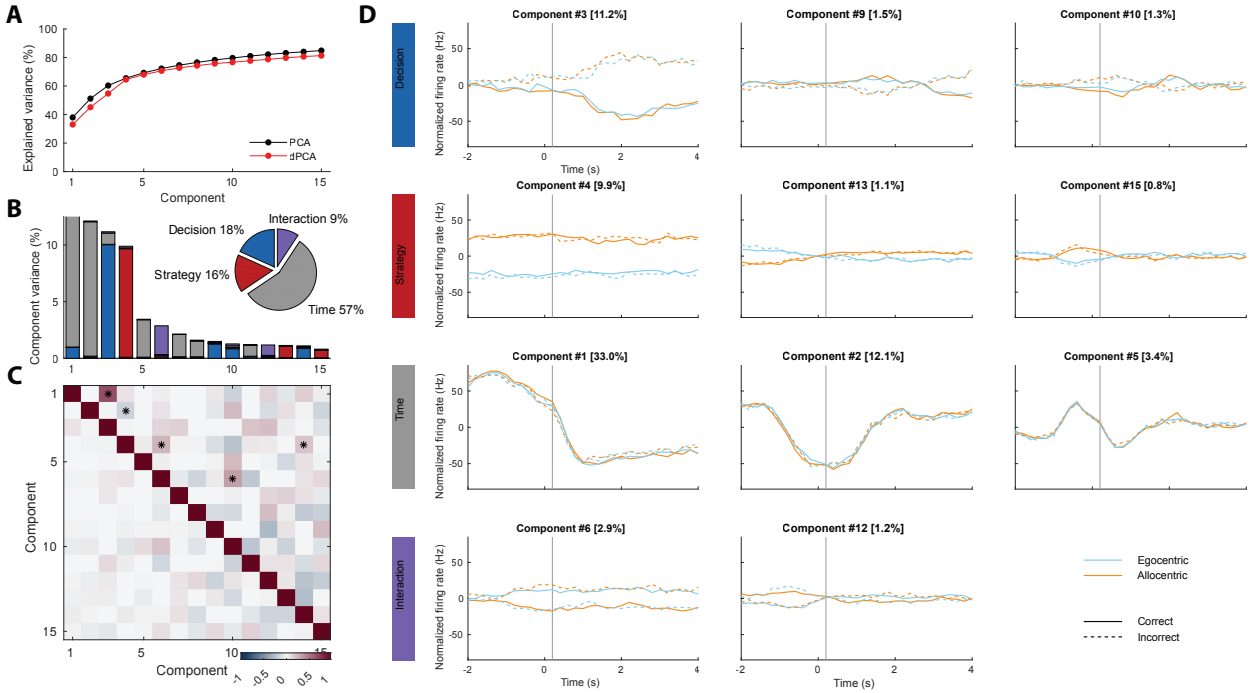
Supplementary Figure S1. Model performance with/without multi-head, replay buffer, different numbers of CA3 neurons and model learning curves. (**A**) Only hcDRQN model trained with two heads and without experience replay is able to solve all the tasks while all the variants with one/two head and with/without experience replay reach only 75% performance. (**B**) Changing the number of CA3 neurons has no effect on the final performance. (**C**) Learning curves for hcDQN, hcDRQN, ML-EWC, ML-SI.



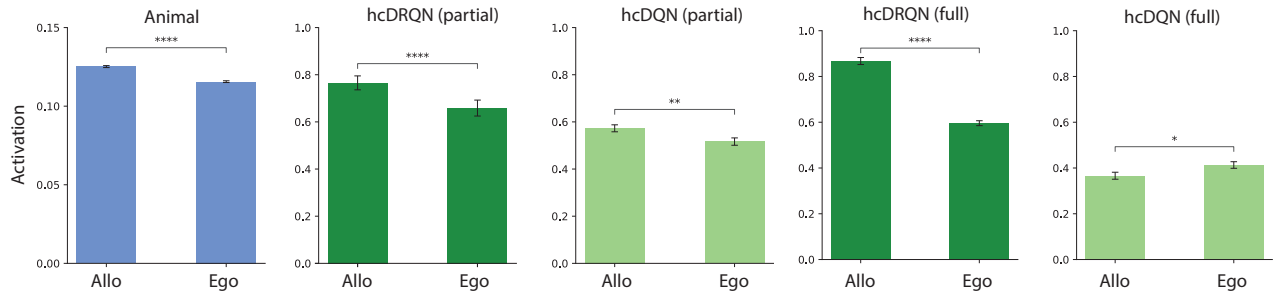
Supplementary Figure S2. CA3 recurrence is needed in partially observable environments. (**A**) Minigrid environment showing full view size with cue removal with agent at the decision point. (**B**) Task performance with cue removal after the agent reaches the decision point for both hcDQN and hcDRQN trained with full observability. Dotted line represents performance of partial view models.



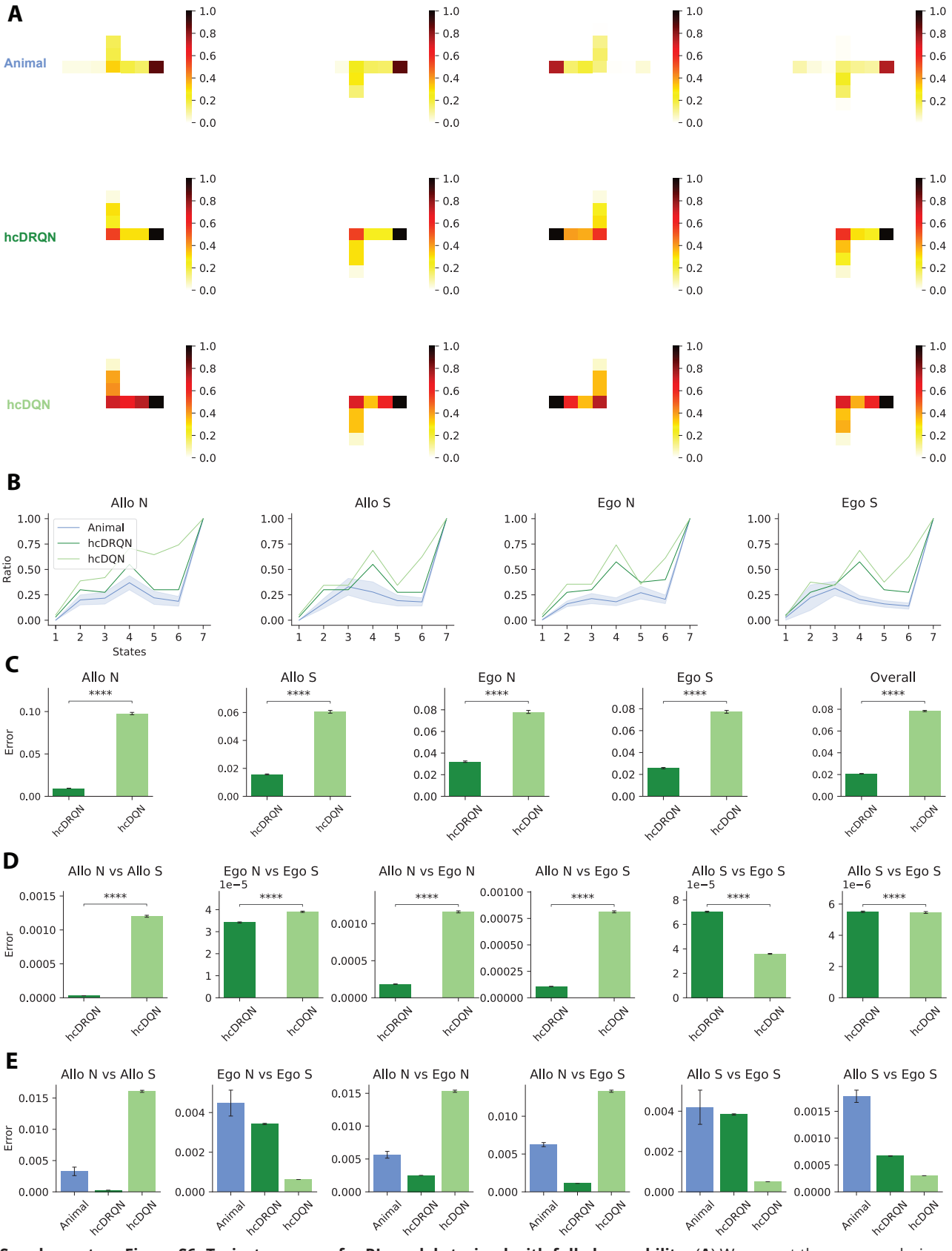
Supplementary Figure S3. Outcome, strategy and temporal neural dynamics in RL agents with full view. (**A**) Demixed PCA components corresponding to Decision - Correct vs Incorrect, Strategies - Allocentric and Egocentric. Qualitatively, full view hcDRQN shows similar trends for decision and strategies, but fails to capture the time component. Full view hcDQN presents mixing activity for strategy components. (**B**) Percentage of explained variance for different components.



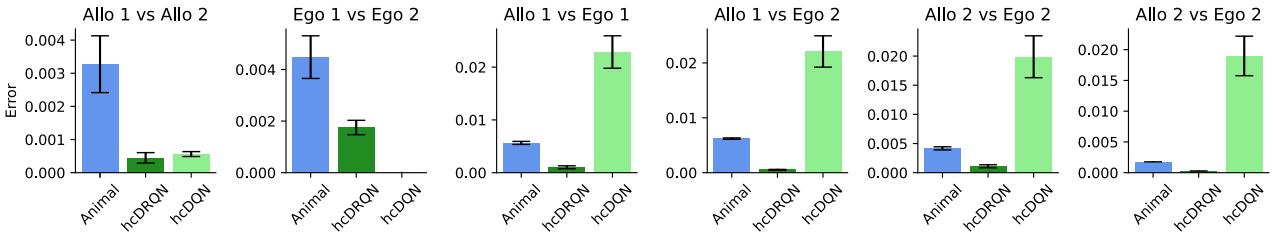
Supplementary Figure S4. Full demixed PCA component analysis. (**A**) Cumulative explained variance of PCA (black) against dPCA (red). (**B**) Variances explained by each demixed principal component. In the pie chart, the total data variance is divided per task-specific variable. (**C**) Dot products between all pairs of demixed principal components is shown in the upper-right triangle. Stars denote the pairs that are significantly non-orthogonal. Correlation among all demixed principal component pairs is displayed in the lower-left triangle. (**D**) Top row: first three decision components; second row: first three strategy components; third row: first three time components; last row: first two decision/strategy interaction components. Figure produced using code made available by [46] (follows a similar structure to the figures available in the original dPCA paper).



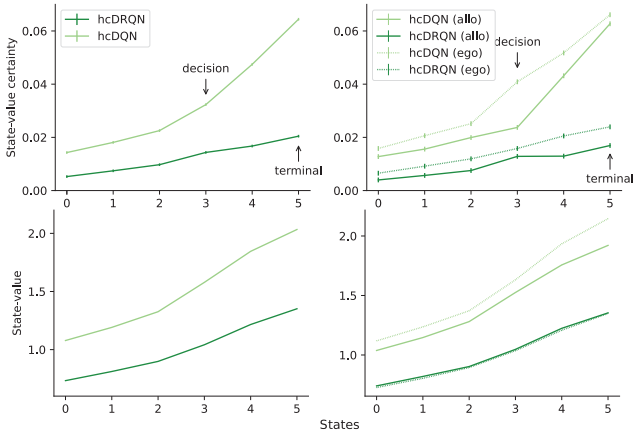
Supplementary Figure S5. Task-specific neural activity. Comparing neural activity between allocentric vs egocentric tasks shows that in both animal and partial view models, allocentric activity is higher. In the full view models, hcDRQN follows the previous pattern while hcDQN shows higher activity in egocentric tasks.



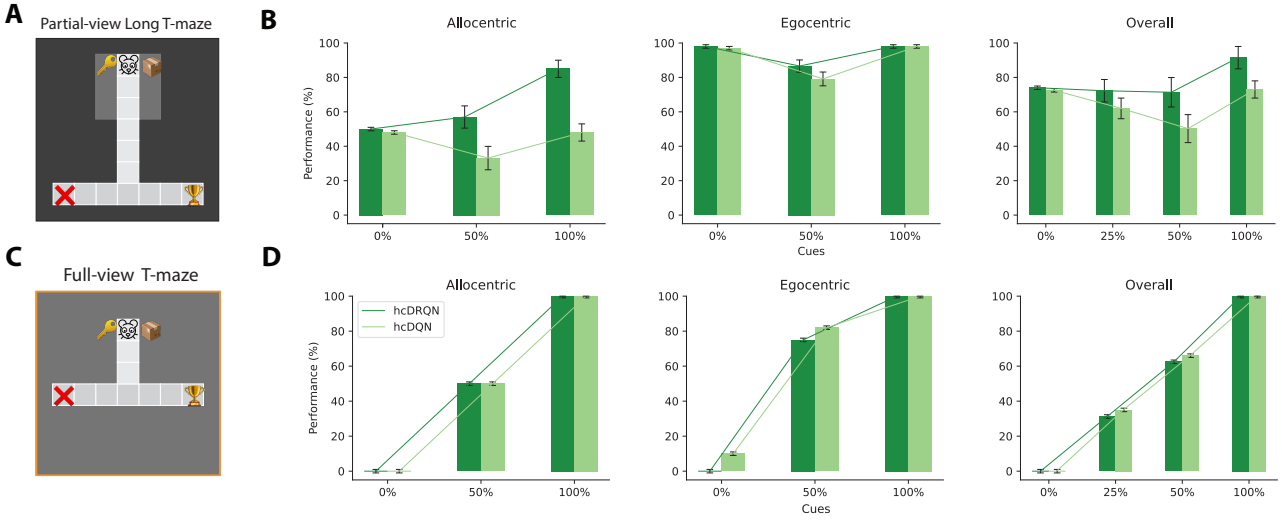
Supplementary Figure S6. Trajectory maps for RL models trained with full observability. (**A**) We repeat the same analysis as done in Fig. 4 with full view models. (**B**) Time spent on each state normalised to the time spent on the final (terminal) state in models and animal. (**C**) Error between agents and animals for each strategy shows that hcDRQN better captures animal behaviour. (**D**) Error between models and animal across all possible task-pairs shows mixed behaviour in terms of which model provides a closer match to animal behaviour. (**E**) Error between all possible task-pairs shows that full view hcDRQN errors are lower for Allo North task ratios while in full view hcDQN are lower for Allo South task.



Supplementary Figure S7. Error between animal and model trajectory-occupancy maps (cf. Fig. 4). Error between all possible task-pairs shows that hcDRQN (trained with partial view) more closely matches animal data when compared to hcDQN (trained with partial view).



Supplementary Figure S8. State-dependent action values for full view model. Top: The state-value certainty given by the variance over Q-values for each state. Bottom: Average state-values.



Supplementary Figure S9. Generalisation test with partial and full view. (**A,B**) Effect of cue removal on models trained with partial view and a long-maze. (**C,D**) Effect of cue removal on models trained with full view and short-maze.