Hippocampus facilitates reinforcement learning under partial observability

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. Although the hippocampus has been associated with goal-oriented navigation, its role in real-world behaviour remains unclear. To investigate this, we combined deep reinforcement learning (RL) modelling with behavioural and neural data analysis. First, we trained RL agents in partially observable environments using egocentric and allocentric tasks. We show that agents equipped with recurrent hippocampal circuitry, but not purely feedforward networks, learned the tasks in line with animal behaviour. Next, using dimensionality reduction, our agents predicted reward, strategy, and temporal representations, which we validated experimentally using hippocampal recordings. Moreover, hippocampal RL agents predicted state-specific trajectories, mirroring empirical findings. In contrast, agents trained in fully observable environments failed to capture experimental observations. Finally, we show that hippocampal-like RL agents demonstrated improved generalisation across novel task conditions. In summary, our findings suggest an important role of hippocampal networks in facilitating reinforcement learning in naturalistic environments.

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 the building of 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 continuous 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 capac-

*These authors contributed equally to this work.

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

ity 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). Another line of research has shown that model-based approaches, known as Successor Representation (SR), explain anticipatory features that have been observed experimentally (20, 21). A related approach combined model-free with model-based networks of the hippocampal-striatal system to reproduce a range of behavioural findings from spatial and non-spatial decision making tasks (22).

A complementary viewpoint has explored the involvement of the hippocampus in reward-centric navigational tasks (6, 7, 23–26). For instance, Foster et al. (23) utilized an actor-critic network coupled with goal-independent representations to model water-maze tasks. Subsequently, this framework was expanded into a hierarchical model (25), thus accelerating learning, particularly in previously visited locations. Arleo and Gerstner (27), on the other hand, proposed a model that integrates unsupervised learning for spatial modelling with reward-based learning for goal-oriented behaviour. Nevertheless, such models did not explicitly differentiate the functional impact of CA3 and CA1 networks. Consequently, the contribution of the classical hippocampal layered structure for navigation of goal-driven environments under partially observable conditions, the underlying neural dynamics, and its implications for animal behaviour remain unclear.

Here, we posit that the hippocampal circuitry, particularly CA3 recurrence, is adept at navigating environments under realistic conditions, particularly those characterized by limited visibility and cue uncertainty. To test this hypothesis we combine behavioural and neural data analysis together with deep reinforcement learning (RL) modelling of goal-driven tasks. Both animals and agents were trained to perform ego-allocentric strategies on a T-maze. Our models consist of a hippocampal-like network trained to perform goal-driven tasks using reinforcement learning. 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, which are better captured by a model incorporating CA3 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.

Results

We were inspired by a standard behavioural setup in which animals are trained on plus-mazes (5). Next, we highlight the key elements of this experimental setup, which we model. First, rats were trained in a multi-task setting in which they had to perform a goal-driven navigational task while following two strategies, egocentric and allocentric (Fig. 1a). 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. On a given trial rodents have to infer the task (ego versus allocentric) through trial-and-error. Second, rats were placed on a standard plus-maze defined by a 5cm wall with cues in the surrounding walls. In such setups, and in most navigational task setups rats are unlikely to be continuously accessing the full environment, implying a degree of partial observability (28). Furthermore, we analysed the speed and head direction of rats as they navigated the maze. This analysis suggested that rats did not have constant access to the full state of the environment (Fig. S1). Note that the maze setup was effectively transformed into a T-maze on each trial by blocking the opposite starting arm (Fig. 1a). Finally, animals were trained using in a block-wise fashion with interleaved blocks containing allocentric and egocentric tasks (Fig. 1a). Each block contains sub-blocks corresponding to different starting locations (i.e., north versus south). Despite the relatively complex nature of these tasks with multiple rules, rats can learn all tasks (Fig. 1b).

Building on this task setup, 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 equipped with a hippocampal-like architecture. To mimic the experimental setup described

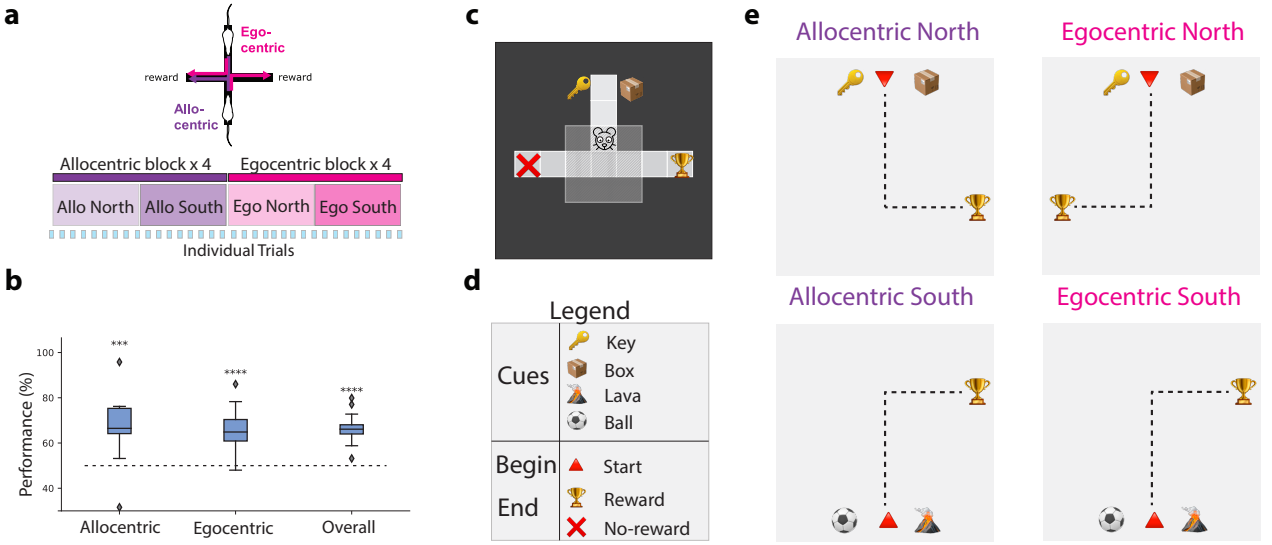


Figure 1. Ego-allocentric task setup in animals and reinforcement learning agents. (**a**) Top, experimental setup in which rats were placed in a plus-shaped maze, which was effectively transformed into a T-maze on each trial by blocking the opposite arm (5). The task consists of reaching the reward 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. There are no statistically significant differences in performance between allocentric and egocentric perspectives. (**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 grey box), which is limited to the current cell alone when the agent enters the terminal arms. Specifically, the agent can only see the squares directly in front of it and to its sides, but not behind it. A total of four cues (cf. (d)) are placed in the environment, two for the 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 the egocentric task, the reward is always on the right side of the starting position. ***: $p < 0.001$, ****: $p < 0.0001$ (two-sided paired t-test).

above, we simulated a 2D minigrid discrete environment (29), 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). Closely following the experimental setup described above, we train the model in a *multi-task* (allocentric north/south and egocentric north/south) using the same *block-wise* training setup, as illustrated in Fig. 1e. As in the experimental setting, RL agents need to figure out the task context through trial-and-error, but this computation in the model is simplified by assuming separate ego and allocentric output action networks (see Methods). We provide a more detailed comparison between our modelling and the experimental setup in Table S1. Motivated by the fact that animals cannot continuously access all sensory information available in their environment (e.g. due to poor visual acuity and head direction), we test different degrees of observability of the environment.

Agents with recurrence learn ego-allocentric goal-driven tasks

Our hippocampal RL models are based on standard deep RL models, specifically, Deep Q-networks (DQN) (30) (see Methods). Our models feature a three-layer hippocampal-like structure: the input layer emulates entorhinal input to the Dentate Gyrus (DG), the first layer represents CA3 and the second CA1. In addition, we also consider an output layer that encodes action-state values, denoted as $Q(a, s)$ (Fig. 2a,b). Environmental cues are provided around the start location of the maze and agents can take one of three actions (move forward, rotate left or rotate right). The entorhinal cortex (EC) is known to supply the hippocampus with a spatial map of the environment (19), which we approximate by providing our model with a 2D top-down spatial map obtained from a minigrid environment (Fig. 1c). The output layer, which encodes state-action Q-values, provides an abstraction of hippocampal-to-striatal networks

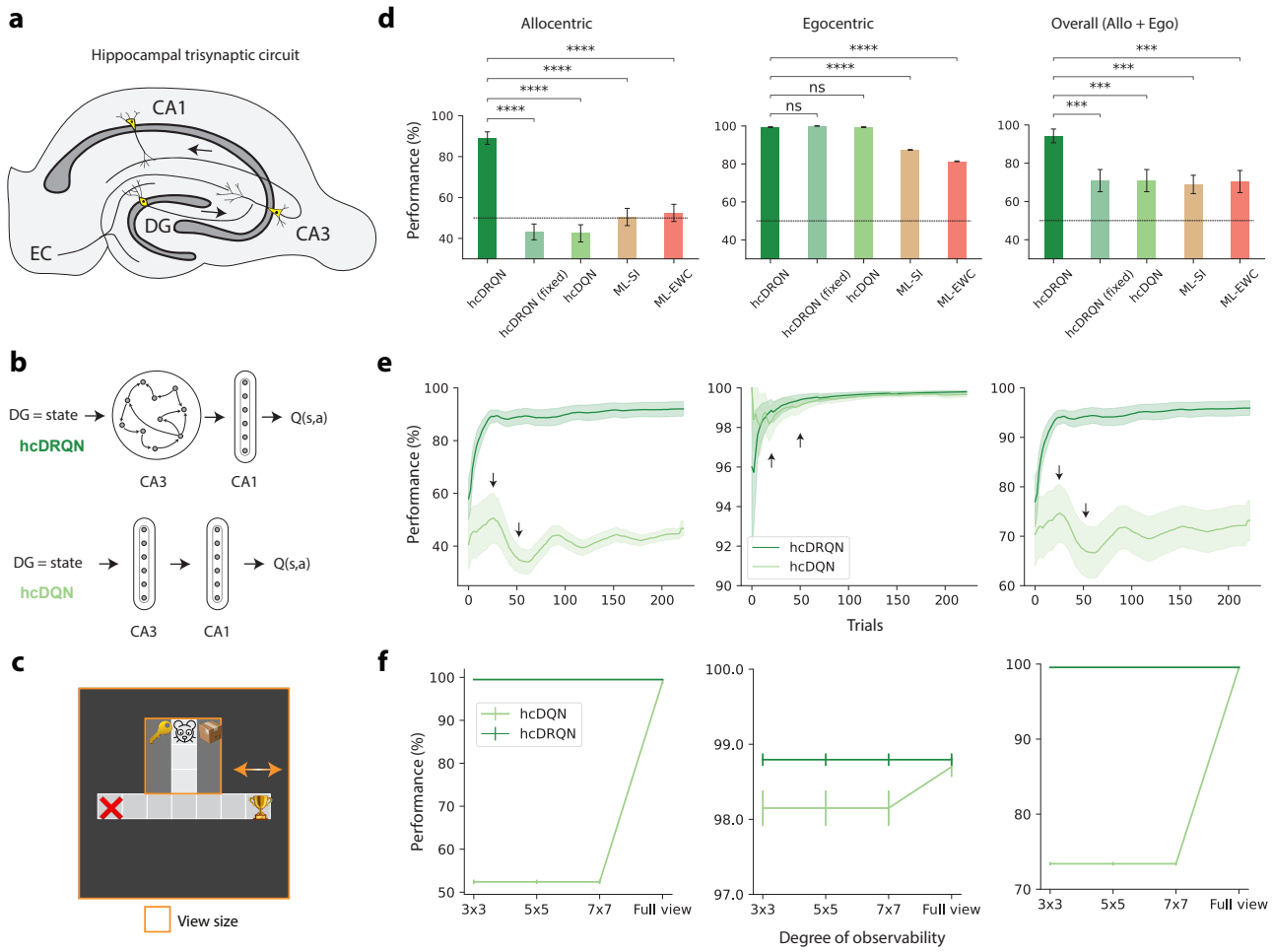


Figure 2. Reinforcement learning agents with CA3 recurrence jointly learn ego and allocentric tasks. (**a**) Classical hippocampal trisynaptic circuitry: entorhinal cortex (EC), dentate gyrus (DG), and hippocampus CA3 and CA1 layers. In addition, an output layer maps hippocampal representations to action Q-values (turn left, turn right, move forward). (**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) and an output action readout layer. (**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.

(31). To prevent forgetting of a previously learned task, we use two separate output Q-heads in all models. Consequently, we switch between the allocentric or egocentric head depending on the current task being performed by the agent. However, other strategies are possible, we also demonstrate that the same tasks can be learned using a contextual task-specific signal (task ID; Fig. S2), as is commonly done in computational neuroscience (32).

Motivated by the existence of recurrent connectivity in CA3 (33) and in line with previous work in which brain areas are modelled as gated recurrent neural networks (34, 35) we model CA3 using a Gated Recurrent Unit (GRU) network (36, 37), 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 modern machine learning algorithms

for as a strong control for what is achievable with feedforward networks in our multi-task setup. In particular, we included two of the most popular methods: Elastic Weight Consolidation (ML-EWC; (38)) and Synaptic Intelligence (ML-SI; (39)). Experience replay is often used to train DQN networks (30). We did not use replay in our model because of the lack of evidence of replay from the hippocampus to itself.

First, we compare the hippocampal deep RL models (Fig. 2b) with animals by contrasting their task performance in a partially observable setting (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 network with recurrence, hcDRQN, can successfully learn multiple tasks (Fig. 2d). Moreover, not only does hcDRQN yield the best performance on both tasks but it is also the only model that can learn allocentric tasks while other models perform around chance level. These results show that hcDRQN learns sufficiently generalisable representations across all tasks considered here. This is due to the RNN's ability to integrate information over time, thus being able to link sensory cues with future outcomes. The action-value outputs must then decode this mixed information from the hippocampal network to attribute value to the appropriate actions. In contrast, hcDQN only learns a subset of the tasks. 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 on 3 of the 4 sub-tasks (allo-south, ego-north and ego-south). This is in line with the performance observed experimentally, showing that animals can learn both strategies (Fig. 1b). In addition, our results show that a model without plasticity in the recurrent CA3 layer is not sufficient to learn all tasks (fixed hcDRQN model), which further supports the importance of the RNN for learning these tasks. Studying how the performance evolves over trials within each allocentric and egocentric block, shows that allocentric performance drops for hcDQN after each switch between north vs. south scenarios (Fig. 2e). This is a consequence of hcDRQN, but not hcDQN agents, being able to learn and thus remember previously encountered tasks. Finally, machine continual learning models perform similarly to hcDQN, thus not being able to learn all tasks (Fig. S3c).

CA3 recurrence is needed in partially observable environments

CA3 recurrence enables the hcDRQN to integrate information over time, allowing it to retain initial cues for accurate action selection at decision points. Therefore, the hcDQN model should match hcDRQN performance in fully observable environments. 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. S5). 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.

Agent's task-relevant neuronal dynamics are in align with experimental recordings

Next, we tested whether RL agents trained in partially environments best capture experimental task-specific neural dynamics. To this end, we contrasted the neural dynamics predicted by the model with those recorded from awake performing animals by using dimensionality reduction techniques. In particular, we used demixed PCA (dPCA), which enabled us to extract behaviourally-relevant dimensions.

We extracted task-specific neural activity from agents in an intermediate stage of learning as to have a more comparable performance to that observed in animals. (cf. Fig. 1b and Fig. S4a). The simulated neural activity of the modelled CA1 layer was then used to perform dPCA (see Methods for details). This analysis focused on three task-encoding components of interest in the hcDRQN agent (Fig. 3a): strategy, time and decision. First, we find that the population dynamics for allo and egocentric tasks can be linearly separable when using the hcDRQN strategy demixed PCs, but not for hcDQN (Fig. 3a left). The fact that hcDRQN, but not hcDQN, yields a clear separation of strategies is consistent with our results above showing that it can learn both allo and egocentric strategies (Fig. 2d). Next, we find

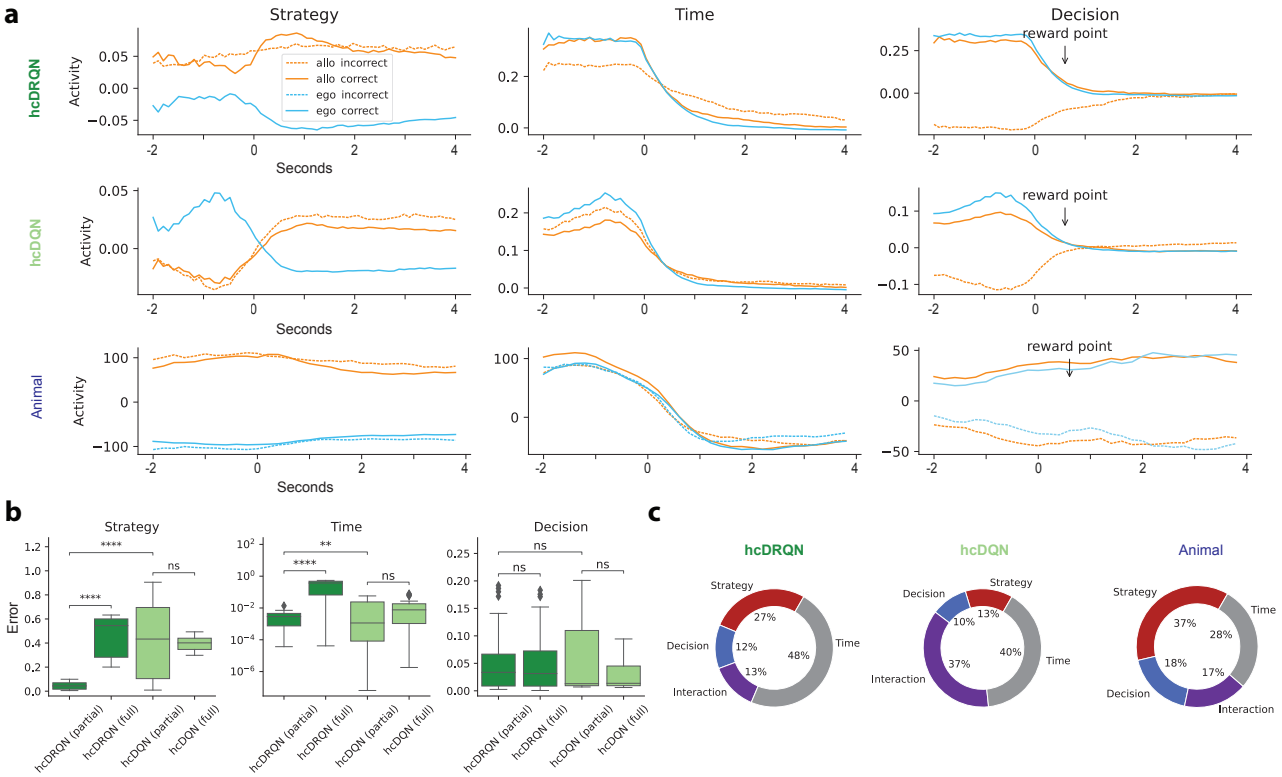


Figure 3. Strategy, temporal, and outcome neural dynamics in RL agents and animals. (**a**) Demixed principal components corresponding to task strategy (allocentric and egocentric for both north and south start locations), time and decision (correct and incorrect). hcDRQN components show separate task strategies whereas hcDQN mixes task strategies. Note that RL agents do not show any egocentric incorrect dynamics because they learn these tasks perfectly (i.e. without incorrect trials). All components are provided in Figs. S8-S10. (**b**) Mean squared error between normalised model and animal data demixed components. We also contrast agents trained with full and partial observability (cf. Fig. S7). (**c**) Percentage of explained variance for all combined Decision, Strategy, Interaction and Time components (cf. with individual components in Fig. S8).

that both hcDRQN and hcDQN exhibit a temporally decaying population dynamics (Fig. 3a middle). These features of the hcDRQN highlight its unique ability to integrate input information over time, with the retained information gradually decaying as time elapses. Finally, we observe decision- or outcome-specific components (correct versus incorrect), which as expected collapse once the reward terminal point is reached. This predicts that the hippocampus should encode reward/no-reward information well before the terminal states. When sensory cues are removed, these task-specific features in the strategy and decision dPCA components are eliminated, aligning with the role of RNN sensory integration in achieving correct task outcomes (Fig. S6).

Next, to contrast the neural representations predicted by the RL agents we used tetrode recordings obtained from 612 CA1 neurons, as described in (5). To conduct this analysis, neural data were aligned to the moment when the reward zone was entered (see Methods). 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 allocentric tasks compared to the egocentric tasks, in line with experimental observations (Fig. S11). 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. This discrepancy between model and animal reward-relevant neural dynamics is likely rooted in the distinction between reinforcement learning agents, which receive instantaneous rewards, and

animals, which require several seconds to consume rewards. 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 the 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).

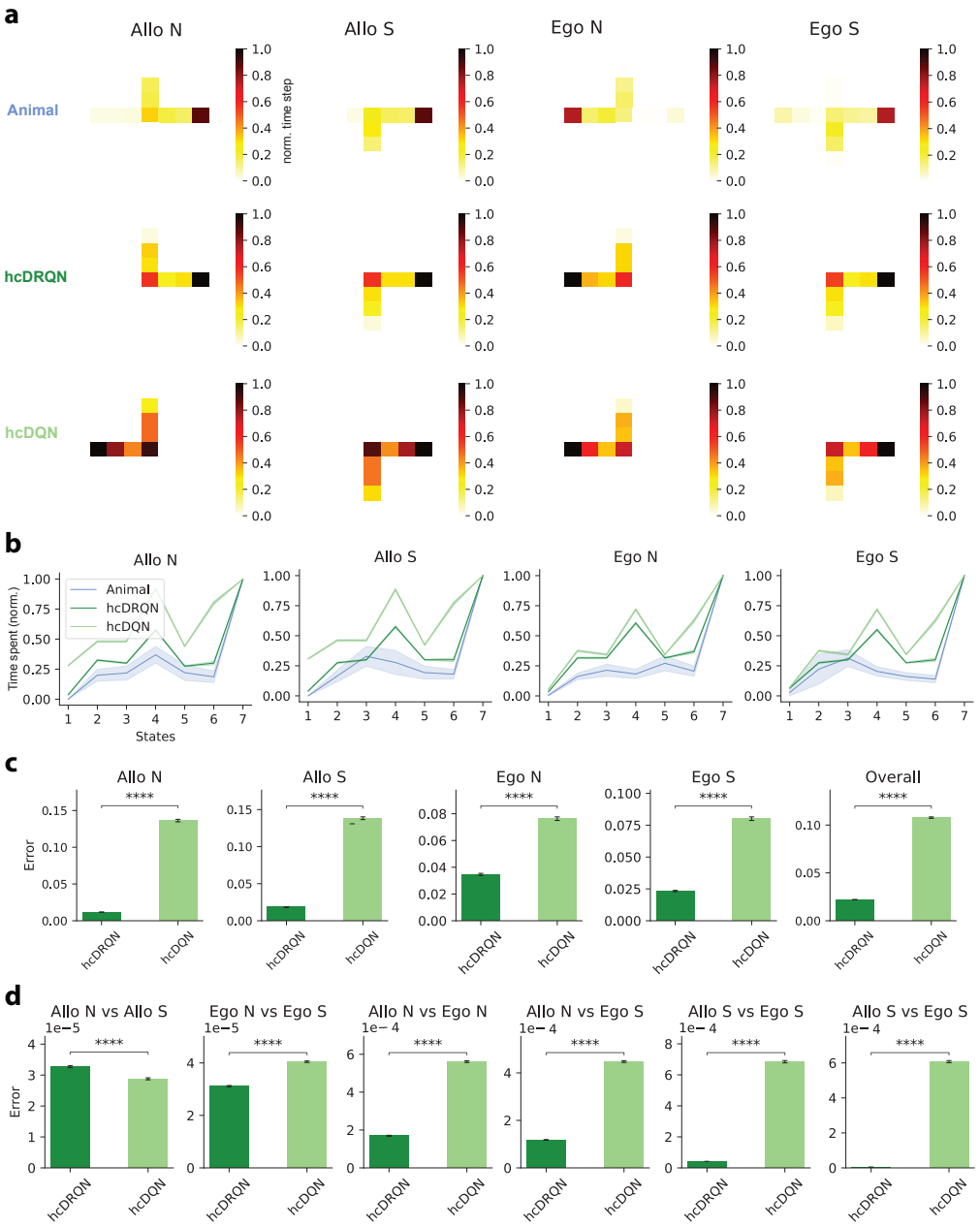


Figure 4. Hippocampal RL agents with recurrence capture animal behaviour. (**a**) Time spent on each maze state across the four strategies for rats, hcDRQN and hcDQN. Animals spend more time at the decision and rewarded terminal points. hcDRQN better captures animal behaviour and hcDQN fails to solve 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. The 2D maze is represented as a series of discrete states along the optimal path from the starting point to the terminal point. Specifically, each of the 7 states corresponds to a unique position along this optimal 1D trajectory within the maze. (**c**) Error of time spent across states between behaviour predicted by the models and animal behaviour. (**d**) Model-animal behavioural errors (as in c) for each 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.

Models trained under full observability conditions do not appear to provide a good match with experimental observations. To further test this point, we compared model neural dynamics in agents trained with full continuous access to the environment (Fig. S7). 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 hcDRQN (13%) and animal (17%), which is in line with its inability to fully solve all the tasks. On the other hand, for both hcDRQN and animal dynamics, but not hcDQN, strategy explains a large amount of the variance (hcDRQN: 27%; animal: 37%).

In summary, our neural dynamics analysis suggests that the hippocampus is indeed involved in task strategy, temporal integration and reward-based decision, in a way that are best matched by hcDRQN RL agents trained in partially observable environments.

Hippocampal RL agents with recurrence better capture animal behaviour

Subsequently, we wondered whether RL agents could also capture animal behaviour during the task. To juxtapose the actions of RL agents against those of animals, we examined the trajectories observed during maze navigation post-learning. In order to facilitate a direct comparison with agent trajectories, we discretized the 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). Interestingly, we find that both models tend to spend more time in the decision state compared to the initial state, which shows that agents take some time before committing to a final decision. This observation is related to vicarious trial-and-error and increased dwell times often found in rodents when approaching decision states (40). Note that agents can stay in a given cell because they can continue rotating left or right, before deciding to move forward. This behaviour is highlighted by cells with darker red (Fig. 4a). On the other hand, as expected, the hcDQN agent fails to discriminate between the two allocentric subtasks and instead learns only one policy (allocentric south; Fig. 4a, bottom). Next, to quantify the time spent on each state we calculated the ratio between time spent in individual states and the final state. This state-to-end ratio shows that hcDRQN also better approximates animal behaviour at this 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 to a subtask 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 a poorer match to behavioural data, further supporting the notion that agents trained under partial observability provide a better description of animal behaviour(Fig. S12, S13).

Overall, hcDRQN trained with partial observability provides a better match to animal behaviour, further supporting the 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 all tasks we expected this agent to yield state-action value predictions with higher and more certain Q-values when compared to the non-recurrent agent. To examine this in more detail we analysed the action-values for each state. First, we highlight the sequence of actions that makes hcDQN take the wrong arm and the correct policy learned by hcDRQN for both allocentric tasks (Fig. 5a). As expected, on average, hcDRQN has higher Q-values than hcDQN, which reflects the fact that it learns all tasks (Fig. 5b, bottom). Next, we studied action selection certainty by calculating the relative Q-value variance across all possible Q-values for a given state (Fig. 5b, top). 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, which can also be observed in models trained with full observability (Fig. S14 and S15). In contrast, hcDQN becomes more certain (i.e. lower variance) after the initial state, becoming again more certain once the decision is made. This result is explained by the relative variance obtained in allocentric tasks (Fig. 5b, right), which is a consequence of the hcDQN's inability to learn allocentric tasks (Fig. 2d).

The observation that the hcDQN becomes overconfident in the allocentric task, which it fails to perform, is reminiscent of the Dunning–Kruger effect. This phenomenon suggests that expert individuals often exhibit less certainty

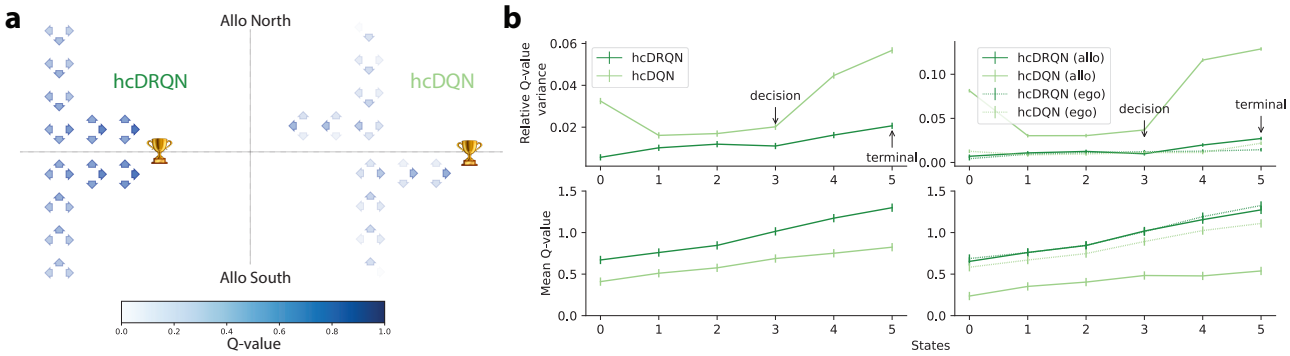


Figure 5. Agent's behaviour predicts state-dependent action values. (**a**) Schematic illustrating state-action policy for hcDRQN and hcDQN for allocentric north subtask. It highlights that hcDRQN solves the task and that it is more uncertain about which action (i.e. different actions have similar values) until the decision point. Each state is represented by 3 coloured arrows corresponding to the state Q-values where darker colour means higher value. Note that the agent is capable of rotating left or right at any position, including the start of the maze. Rotating changes the agent's orientation but not its position, which is why there are non-zero Q-values associated with the left and right actions even at the starting location. (**b**) Top: Relative Q-value variance, given by the variance over Q-values for each state divided by the mean Q-value for each state. It shows that hcDQN decreases its relative variance as it gets closer to the decision state (see arrows) and that it has higher overall relative variance compared to hcDRQN. Bottom: Average Q-values over environmental states. Error bars represent standard error of the mean over 5 different initial conditions.

than those who are less knowledgeable or skilled (41).

Network recurrence enables generalisation to stochastic environments

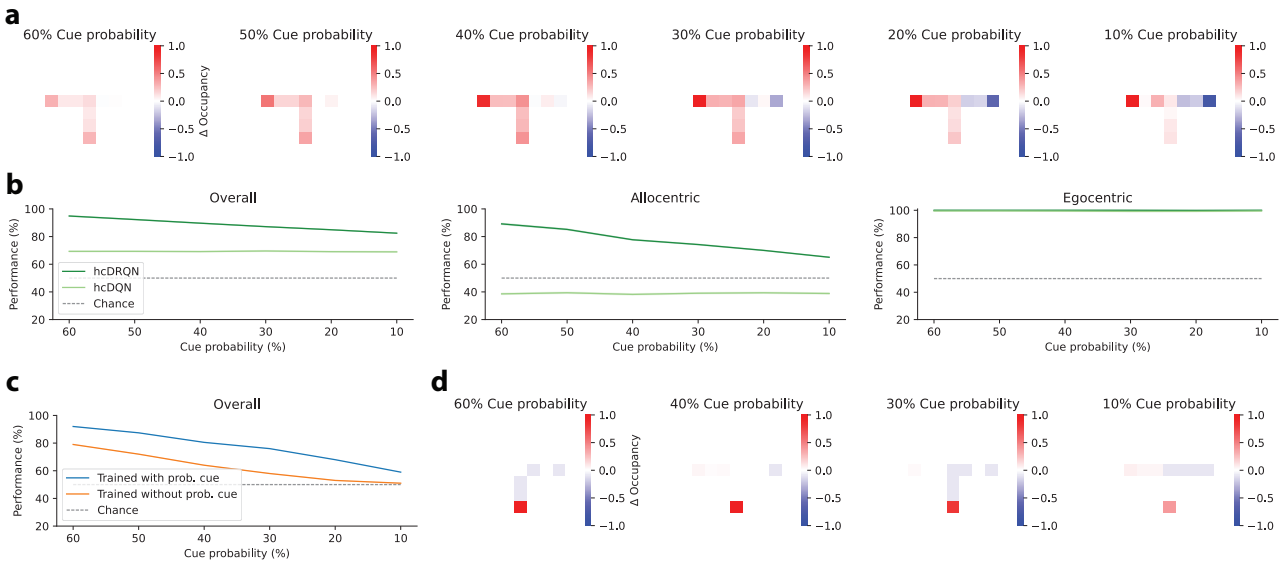


Figure 6. Recurrence enables better generalisability to stochastic environments. (**a**) Change in state-occupancy (with probabilistic cues - without probabilistic cues) for the hcDRQN agent across different degrees of cue removal. (**b**) hcDRQN outperforms hcDQN across different degrees of probabilistic 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.

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 (42, 43). To test the effect of stochasticity on the different agents, we created environments in which cues randomly appear and disappear (Fig. 6). We study two scenarios: (i) incremental random cue removal during inference (i.e. after learning) and (ii) effect of random cue removal on learning. During inference, the performance of hcDRQN gradually decreases as the likelihood of cue removal increases (Fig. 6a). In contrast, for hcDQN, which lacks recurrence, its

performance decreases as soon as the cues are removed, regardless of the degree of removal (Fig. 6b). 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 (Fig. 6c). Indeed, when trained under these conditions the model performs consistently better than a model trained without random cue removal (10% improvement). Finally, when analysing trajectory behaviour, our model shows that agents trained with probabilistic cues spend more time at the starting location to integrate sensory evidence for longer before committing to a decision (Fig. 6d).

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

hcDRQN agent generalises to different task conditions

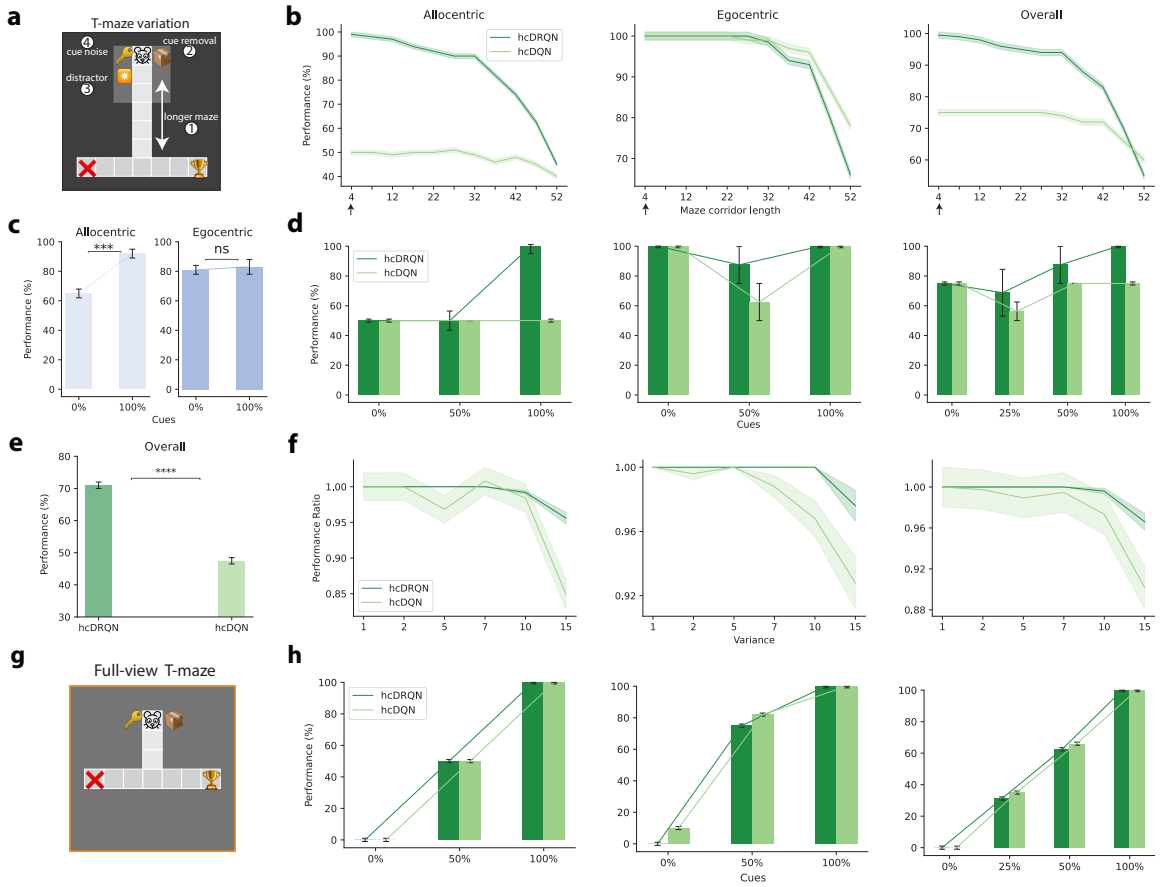


Figure 7. hcDRQN shows better generalisation to maze length, cue removal, distractors and sensory noise. (**a**) T-maze setup for increased length of the middle corridor, cue removal and random noise. (**b**) Performance decrease over gradual increase of the maze length shows that hcDRQN can handle middle 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. (**g**) T-maze setup with full observability. This setup was used to evaluate generalisation capabilities following cue removal (h). (**h**) Compared to models with partial observability, all models with full observability show a drop in performance when cues are removed. Error bars represent the 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 experi-

enced 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 a 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 (Fig. 7d and S16 for the same test with longer maze). 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 (44)(Fig. 7c). In contrast, models trained with full observability cannot generalise, as they rely on the presence of specific cues (Fig. 7g and h). Furthermore, the performance of both allocentric and egocentric tasks under full observability declines proportionally with the removal of cues. In both cases, the models fail to complete the tasks when no cues are present. This is in contrast to the partial observability version, where hcDRQN shows some degree of robustness to cue removal (cf. Fig. 7d).

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 using tetrode hippocampal neuronal recordings. Furthermore, hippocampal-like RL agents predicted state-specific trajectories which resemble experimentally-observed animal behaviour. In stark contrast, agents trained in fully observable environments failed to replicate the experimental data. Importantly, hippocampal-like RL agents with CA3 recurrence 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 (45, 46). In addition, when our models were trained with full observability, they could not generalise (Fig. 7h), 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 (47). Partially observable environments represent a more realistic setup, which together with our results suggests that hippocampal CA3 region may have evolved to support the ability to navigate in these conditions.

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 is in line with experimental results showing that the CA3 region is involved in maintaining working memory representations in delayed-to-match sample tasks (9–11). Moreover, using low dimensionality analysis we have shown that hippocampal neuronal dynamics encodes a difference between correct and incorrect trials.

This could be a reflection of animals taking different trajectories on incorrect and correct trials or reward value transmitted from downstream brain areas as predicted by temporal different reinforcement learning algorithms (48, 49). Furthermore, our work suggests that CA3 recurrence enable hippocampal networks to develop representations that are context-invariant, which can then be used by downstream decoders for action selection and reinforcement learning.

Our study generates experimentally testable predictions on the role of CA3 recurrence in facilitating navigation in environments with limited visibility. In particular, our work predicts that perturbing CA3 recurrence should impair the ability of animals to generalise to new and stochastic task conditions. Furthermore, our findings suggest that plasticity in recurrent CA3 synapses is important in supporting these functions (Fig. 2d).

Because our focus is on contrasting models with neuroscientific observations, we have used the same block-wise training of ego- and allocentric tasks as employed experimentally. In this setup, our work predicts that CA3 recurrence is particularly important to learn allocentric tasks (Fig. 2d). Generalisation tests, show that our models are more robust to the removal of cues in egocentric tasks when compared to allocentric tasks (Fig. 7c), consistent with experimental observations (44) and suggesting that the agents have learned the macrostructure of the task. However, the tasks that we have used here represent only a small subset of all the possible challenges that animals are 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. Relatedly, we have assumed the existence of action-value output networks that perform clean allo-egocentric task-switching. While such computations are believed to be executed by striatal networks (50), the precise mechanisms remain elusive. Finally, we have used artificial algorithms for credit assignment in hippocampal networks. All these elements remain to be explored in future work.

Our results show that in the multi-task block-training setting we tested, a model with a recurrent layer (hcDRQN) without experience replay outperforms alternative methods that were specifically designed for continual learning, consistent with recent machine learning findings (51). 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.

Classical hippocampal models suggest that CA3 recurrency enables pattern completion (52, 53), while more recent computational models propose that the hippocampus creates a predictive map of the environment through successor representations (SR) (21). Our research aligns with this predictive view of the hippocampus, because in our model the hippocampus learns to predict the correct trajectory to take given sensory input, which is then reflected in simulated and experimental neuronal dynamics (Fig. 3a). Furthermore, our findings underscore the essential role of CA3 in constructing the hippocampal predictive map, consistent with the predictive view of hippocampal function (21, 26). In another set of studies recurrent neural networks (RNNs) have been trained to support spatial navigation. They have shown that RNNs develop spatial representations 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 properties akin to those in Banino et al. (14), Cueva and Wei (15). While these studies emphasize the importance of memory (see also (17)) and recurrence in hippocampal networks, they do not assess their function in partial observability and its relationship to experimental observations in goal-driven tasks, which is our focus.

In comparison, other models have explicitly considered reward-based navigational tasks. For example, work by Foster et al. (23) utilises an actor-critic network and goal-independent representations for solving water-maze tasks, which was later extended hierarchically by Tessereau et al. (25). On the other hand, Arleo and Gerstner (27) combined external vision, path-integration and goal-driven reinforcement learning in a model evaluated on a physical robot. Whilst these models do connect various components to distinct brain structures, they have largely overlooked the significance of CA3 recurrence and its implications in real-world scenarios, including partial observability. Our research highlights the critical nature of these features in accurately modelling both behaviour and neuronal dynamics.

A limited number of models in neuroscience have considered environments with partial observability (54, 55). A notable exception are Clone Structured Cognitive Graphs that create latent state-space maps for aliasing (i.e. partial observability) while capturing cognitive map phenomena (55, 56). Our work adds to this body of literature supporting the importance of partial observability in explaining experimental observations.

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.

Materials and 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 (29), mimicking the T-maze environment (Fig. 1a) in line with common experimental setups, which allows 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 contrast 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) (30), 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:

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

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^-). The target network is a frozen version of the primary neural network, which helps to stabilise reinforcement learning. It is updated at regular intervals to avoid inflated estimations of the Q-values. The hyperparameter *target update counter* controls the frequency of updates of the target Q-network. Equation 1 calculates expectation of the squared error between the target Q-value and the predicted Q-value over a batch of state-action experience tuples. We minimise this expected error by adjusting the parameters θ_i of the network using standard gradient descent and Adam optimiser (see parameters in Table S1).

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 layer 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 (31). A Q-value output action head was not sufficient to solve all tasks that we consider, even in the presence of replay mechanisms (Fig. S3a). Therefore, to ensure that the network could solve the two tasks (ego and allocentric) we use task-specific heads (i.e. two heads) at the output (see Methods), for all the models we consider. In biological networks this could be implemented through task-specific contextual signals (57). Indeed, it is also possible to train agents with only one head, provided that a task-specific contextual signal is provided as input to the model (Fig. S2). Note that it is only at the output level that we assume task-separation (i.e. one output head for allocentric and another for egocentric tasks). This means that the representations developed by the hippocampus have to be generalisable across the different tasks. Given that we can capture key aspects of experimental data with this setup, this implies that task-specific Q-action values are encoded outside the hippocampus.

All our models have a three-layered (hippocampal-like) structure in which the input layer is of shape (RxC) where R is the number of rows in the input grid, C is 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 ($\gamma = 0.9$), no replay memory (i.e. replay memory buffer size $= 1$, but see Fig. S3 for different variants) and a batch size of 1 during training. Adam is used as the optimiser with a learning rate α of 0.001. We use a CA3 layer size of 50 for all models considered. In the case of hcDRQN these units are Gated Recurrent Units. A Gated Recurrent Unit (GRU) (36, 37) network is a specially designed recurrent neural network that can capture long temporal dependencies in data. GRUs achieve this through a gating mechanism, which regulates the flow of information. This mechanism includes an update gate, controlling the extent to which a GRU unit keeps the previous state, and a reset gate, deciding how much past information to forget. The learning rate and ϵ ($0.3 - 0.05$) for ϵ-greedy, which controls the trade-off between exploration (choosing new actions) and exploitation (using known actions) have been selected using a grid-search. All the hyperparameters are given in Table S2.

Partial observability

In our grid-based environment, models operate under conditions of limited environmental 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). To model egocentric trials the reward location changed between north and south, but for allocentric trials it was kept constant. 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 egocentric trials. The same north/south combination was maintained throughout the duration of the egocentric trials. In total, the training consisted of 10,000 individual trials (200 blocks each with 25 trials for both ego and allocentric tasks).

All model parameters (weights and biases) were updated using BPTT within the trial and once the model reached the terminal state (i.e. there is no BPTT across trials and batch-size=1). Further, BPTT is applied at every two steps backwards in time until the first step is reached, as is commonly done in deep RL (58). The maximum number of steps that BPTT can go back in time is 128 steps, which is also the max number of steps during inference. hcDQN is trained in a similar fashion but we do not backpropagate the gradients in time towards the first step. During model testing, such as during the generalisation tests, we turn off learning/plasticity.

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

Continual Learning, also known as Incremental or Life-long Learning, refers to the ability of a model to sequentially master multiple tasks without losing previously acquired knowledge, even when data from older tasks are no longer accessible during the training of new ones. We evaluated the performance of two state-of-the-art regularisation-based continual learning algorithms: Elastic Weight Consolidation (EWC) (38) and Synaptic Intelligence (SI) (39). These artificial algorithms were tested against our recurrent and non-recurrent DQN architectures in identical task settings. EWC adds a penalty term to the loss function, derived from a Fisher Information Matrix (also called weight importance), which guides the learning such that information pertaining to previous tasks remains preserved. On the other hand, SI determines the importance of each neural network weight throughout the learning process and uses this information to restrict weight updates, preserving prior task knowledge. Both importance measures, which are tunable configurations (hyperparameters), were selected through hyperparameter optimisation (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 components. 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 a 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 dependent 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 supercomputer 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 **** ($p < 0.0001$).

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 bin size of 0.2s (5). Neural activity was aligned to the crossing of a reward-sensor. We considered neural activity 2s before and 4s after crossing the reward sensor for dPCA analysis. The reward sensor was located 10 cm from the end of the rewarded area. Animals were trained on a T-maze task in which rats had to follow both allocentric and egocentric navigational rules to reach reward points. We performed demixed Principal Component Analysis (dPCA) (59) on the neuronal firing rates using 3 behavioural variables – trial decision, strategy and time (dPCA $\lambda = 2.919e^{-05}$ was found using grid-search as done by (59)). We used the dPCA code made available by the authors of (59) in https://github.com/machenslab/dPCA. For models, we aligned data to the timestep in which a terminal state was reached and converted timesteps to seconds by making one timestep correspond to 200ms. To contrast with animal data, we continued recording the neural activity for 4s after reaching the terminal. The animal experiments used in our work were approved by the appropriate ethics committee as in Ciocchi et al. (5).

To quantitatively assess the differences in neural dynamics between observed data and model, we calculated the mean squared error (MSE) for the principal components derived from dPCA of both the animal and model CA1 neural activities. The MSE represents the average of the squared differences between the observed dPCs (Y_i) and the dPCs predicted by the model ($\hat{Y}_i$), as given by $\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$. Y and $\hat{Y}_i$ were normalised by their max and min values before computing this error metric.

Behavioural data

The behavioural data (i.e. animal task performance) consists of a total of 47067 trials recorded over multiple days (3 to 7) from a total of 5 animals (5). However, as some animals only had a maximum of 800 continuous trials we used a maximum of 800 continual trials per animal. The maze was cleaned with an odour-neutral solution every 10 trials to avoid odour-guided navigation as described in (44).

Acknowledgements

We would like to thank the Neural & Machine Learning group, Kim Stachenfeld and Chris Summerfield for useful feedback.

Funding

DP was funded by the EPSRC Centre for Doctoral Training in Future Autonomous and Robotic Systems (FARSCOPE), SM by the Wellcome Trust (PMAG/563) and BBSRC and RPC by the Medical Research Council (MR/X006107/1), BBSRC (BB/X013340/1) and a ERC-UKRI Frontier Research Guarantee Grant (EP/Y027841/1). HM was funded by a FWF grant (I 5458; part of the German Research Foundation Research Unit 5159) and a WWTF grant (CS18-039). SC 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

The author contributions are as follows:

Conceptualization: DP, SM, and RPC Methodology: DP, SM, HMV, SC and RPC Investigation: DP, SM, MVD, HMV, SC and RPC Visualization: DP, SM, HMV and RPC Supervision: RPC Writing—original draft: DP, SM, and RPC Writing—review and editing: DP, SM, HMV, SC and RPC

Competing Interests

The authors declare no competing interests.

Data and Materials Availability

All data needed to evaluate the conclusions in the paper are present in the paper and/or the Supplementary Materials. The animal data and code used in this study is available in Dryad (http://doi.org/10.5061/dryad.qv9s4mwnw). We used the PyTorch library for all reinforcement learning models. The code used in our simulations is also available at http://github.com/neuralml/hcRL.

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*. Academic Press, 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*, 110(3):394–422, 2022.

[8] M. G. Edelson and T. A. Hare. Goal-dependent hippocampal representations facilitate self-control. *Journal of Neuroscience*, 43 (46):7822–7830, 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. In *International Conference on Learning Representations*, 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*, pages 2020–11, 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*, 34(1):171–175, 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] P. Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural computation*, 5(4): 613–624, 1993.

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

[22] J. P. Geerts, F. Chersi, K. L. Stachenfeld, and N. Burgess. A general model of hippocampal and dorsal striatal learning and decision making. *Proceedings of the National Academy of Sciences*, 117(49):31427–31437, 2020.

[23] D. J. Foster, R. G. Morris, and P. Dayan. A model of hippocampally dependent navigation, using the temporal difference learning rule. *Hippocampus*, 10(1):1–16, 2000.

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

[25] C. Tessereau, R. O'Dea, S. Coombes, and T. Bast. Reinforcement learning approaches to hippocampus-dependent flexible spatial navigation. *Brain and Neuroscience Advances*, 5:2398212820975634, 2021.

[26] C. Fang and K. L. Stachenfeld. Predictive auxiliary objectives in deep rl mimic learning in the brain. *arXiv preprint arXiv:2310.06089*, 2023.

[27] A. Arleo and W. Gerstner. Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity. *Biological cybernetics*, 83(3):287–299, 2000.

[28] T. Yoon, J. Okada, M. W. Jung, and J. J. Kim. Prefrontal cortex and hippocampus subserve different components of working memory in rats. *Learning & memory*, 15(3):97–105, 2008.

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

[30] 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.

[31] 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.

[32] G. R. Yang, M. R. Joglekar, H. F. Song, W. T. Newsome, and X.-J. Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature neuroscience*, 22(2):297–306, 2019.

[33] 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.

[34] 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.

[35] 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.

[36] K. Cho, B. van Merriënboer, Ç. Gulçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

[37] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[38] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[39] F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017.

[40] A. D. Redish. Vicarious trial and error. *Nature Reviews Neuroscience*, 17(3):147–159, 2016.

[41] J. Kruger and D. Dunning. Unskilled and unaware of it: how difficulties in recognizing one's own incompetence lead to inflated self-assessments. *Journal of personality and social psychology*, 77(6):1121, 1999.

[42] 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.

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

[44] 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.

[45] 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.

[46] 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.

[47] M. Hausknecht and P. Stone. Deep recurrent q-learning for partially observable mdps. In *2015 AAAI Fall Symposium Series*, 2015.

[48] 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.

[49] W. Du, E. Li, J. Guo, R. Arano, Y. Kim, Y.-T. Chen, A. Thompson, S. J. Oh, A. Samuel, Y. Li, et al. Directed stepwise tracing of polysynaptic neuronal circuits with replication-deficient pseudorabies virus. *Cell Reports Methods*, 3(6), 2023.

[50] M. Seo, E. Lee, and B. B. Averbeck. Action selection and action value in frontal-striatal circuits. *Neuron*, 74(5):947–960, 2012.

[51] B. Ehret, C. Henning, M. R. Cervera, A. Meulemans, J. von Oswald, and B. F. Grewe. Continual learning in recurrent neural networks. In *International Conference on Learning Representations (ICLR 2021)*, 2021.

[52] 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.

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

[54] E. Vértes and M. Sahani. A neurally plausible model learns successor representations in partially observable environments. *Advances in Neural Information Processing Systems*, 32, 2019.

[55] D. George, R. V. Rikhye, N. Gothoskar, J. S. Guntupalli, A. Dedieu, and M. Lázaro-Gredilla. Clone-structured graph representations enable flexible learning and vicarious evaluation of cognitive maps. *Nature communications*, 12(1):2392, 2021.

[56] M. Lazaro-Gredilla, I. Deshpande, S. Swaminathan, M. Dave, and D. George. Fast exploration and learning of latent graphs with aliased observations. *arXiv preprint arXiv:2303.07397*, 2023.

[57] 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.

[58] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *International Conference on Learning Representations*, 2021.

[59] 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.

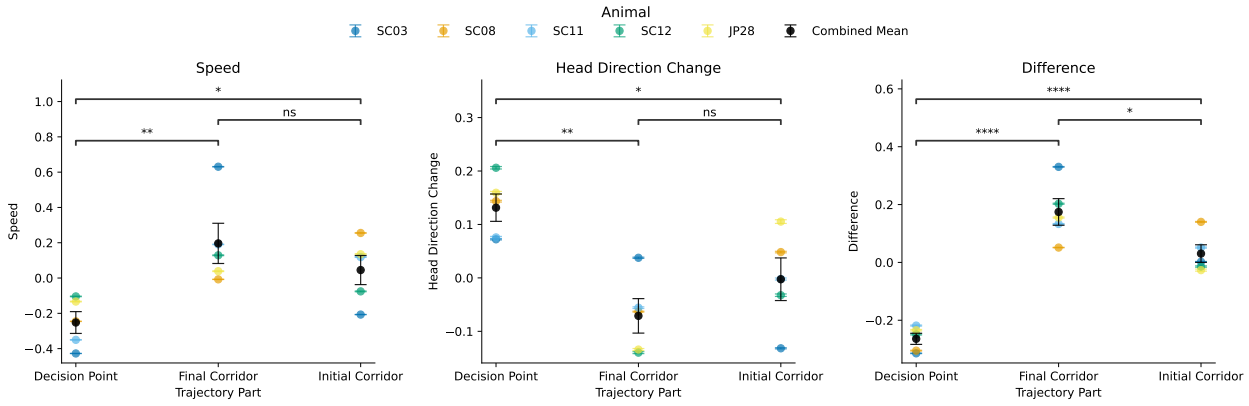
Supplementary Materials

	Experimental task	RL agent
Credit assignment	Reward-based learning[1]	Q-learning + BPTT (within trial)
Main tasks	Allo, Ego	Allo, Ego
Sub-tasks	North, South	North, South
Task inference	Trial-and-error	HC + output layers[2]
Action selection	Striatum[1]	Separate ego and allo action outputs
Policy	Unclear[1]	ϵ-greedy
Sensory input	External cues and maze	Simplified cues and 2D maze
Training regime	Block-wise	Block-wise
Trials per block	25-50	25
Total number of trials	800	15000
Trained params.	Synapses[1]	Weights and biases
Env. observability	Inherently limited[1]	Partial (3x3) or full (9x9) view
Sensory repres.	Latent from complex input	Latent from (simplified) 2D world view
Architecture	Full brain	Focus on hippocampus
Neural activity	CA1 tetrode recordings (sub-sampling)	Full access to CA1 activity

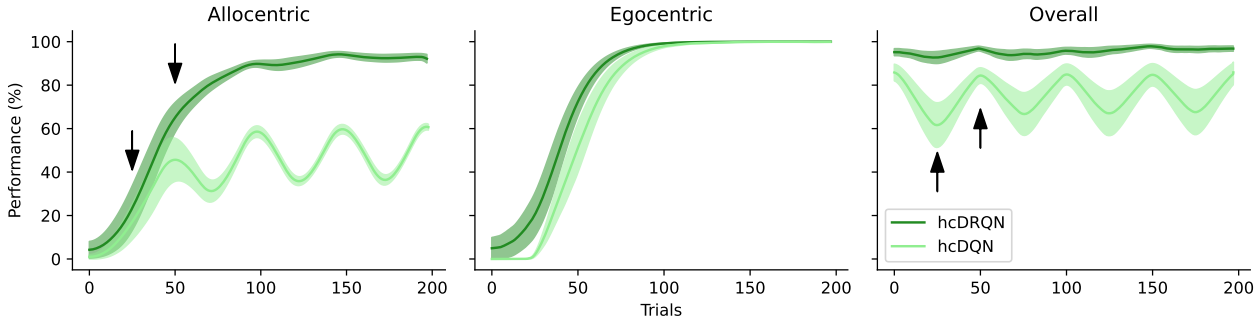
Supplementary Table S1. Mapping between experimental setting and our RL modelling framework. [1] These elements remain open to interpretation, and we offer our own perspective on how the brain may achieve these components. [2] Although our RL agents still require trial-and-error to infer the task, this inference process is made easier by assuming separate ego and allocentric output action heads.

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

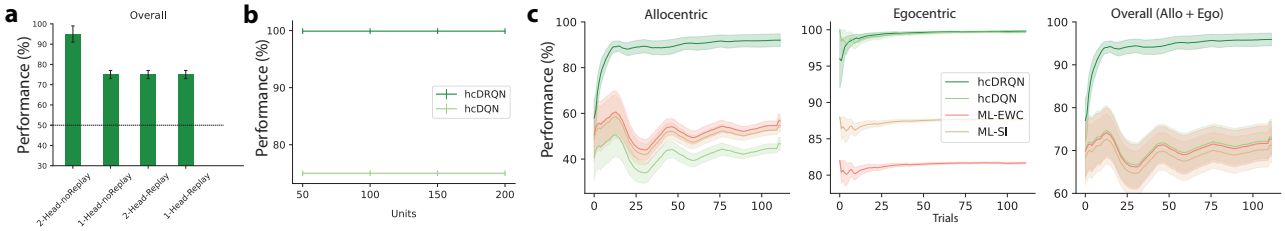
Supplementary Table S2. Hyperparameters used to run the experiments given in the paper.



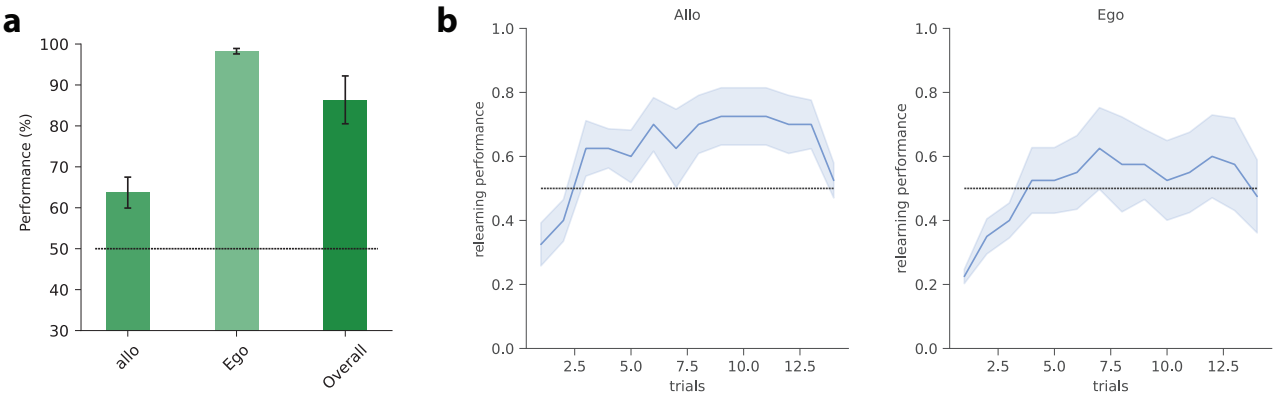
Supplementary Figure S1. Speed and head direction change in animals. This figure compares the animals' speed and head direction change in three zones of the maze: the initial corridor, the decision point, and the final corridor. The results show that the animals move at a lower speed when at the decision point compared to the corridor zones (left), indicating a slowdown as they make navigational choices. Conversely, the head direction change is significantly higher at the decision point than in the corridors (right), suggesting that the animals engage in more head movements, which is likely a reflection of turning towards the final arm. In the corridor zones, the animals tend not to look around extensively, supporting the idea that they do not constantly scan the environment during forward navigation, thus suggesting a degree of partial observability.



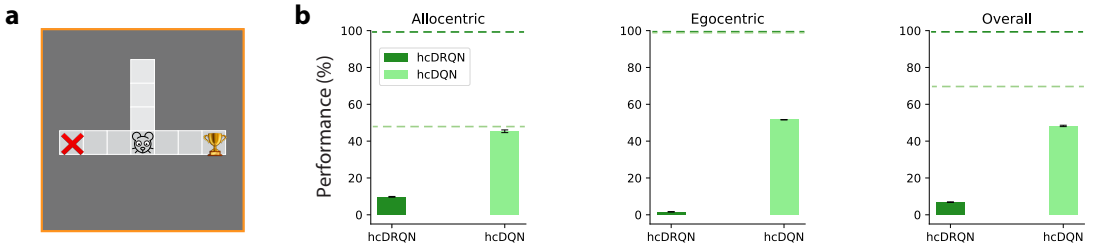
Supplementary Figure S2. Performance of hcDRQN and hcDQN using a single head output and task ID. Although both models receive task ID information, only hcDRQN successfully solves both allocentric and egocentric tasks. The arrow indicates the switching point between north and south starting locations.



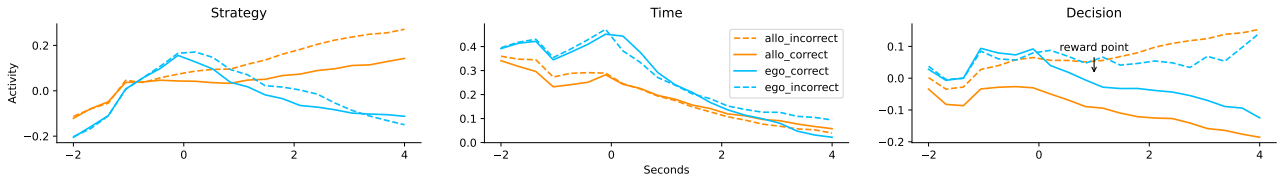
Supplementary Figure S3. Model performance with/without multi-head, replay buffer, different numbers of CA3 neurons and model learning curves. (**a**) Only hcDRQN model trained with two heads and without experience replay is able to solve all the tasks while all the variants with one/two head and with/without experience replay reach only 75% performance. Alternatively its possible to train agents with only one head but providing them with task-specific information Fig. S2. (**b**) Changing the number of CA3 neurons has no effect on the final performance. (**c**) Learning curves for hcDQN, hcDRQN, ML-EWC, ML-SI.



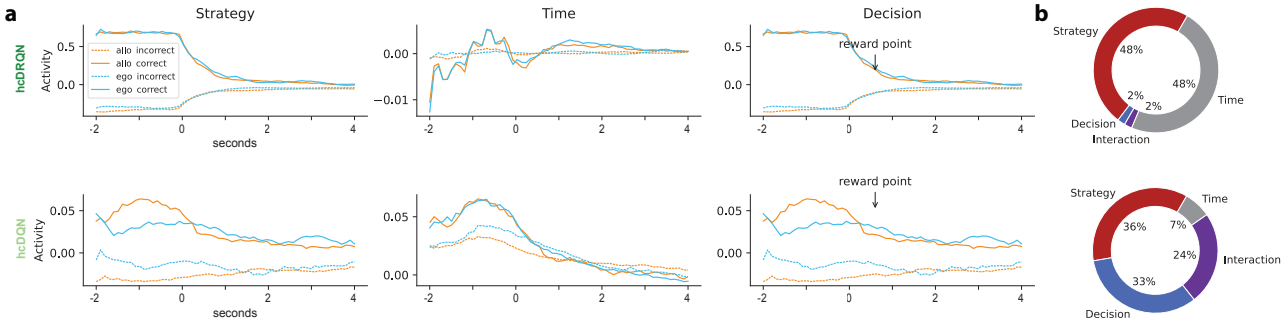
Supplementary Figure S4. hcDRQN model performance mid learning and animal learning curves. (**a**) Allocentric, egocentric and overall performance for hcDRQN model taken during middle phase of training. (**b**) Learning curve for animal data averaged across allo/egocentric blocks. They resemble the ones from the hcDRQN model.



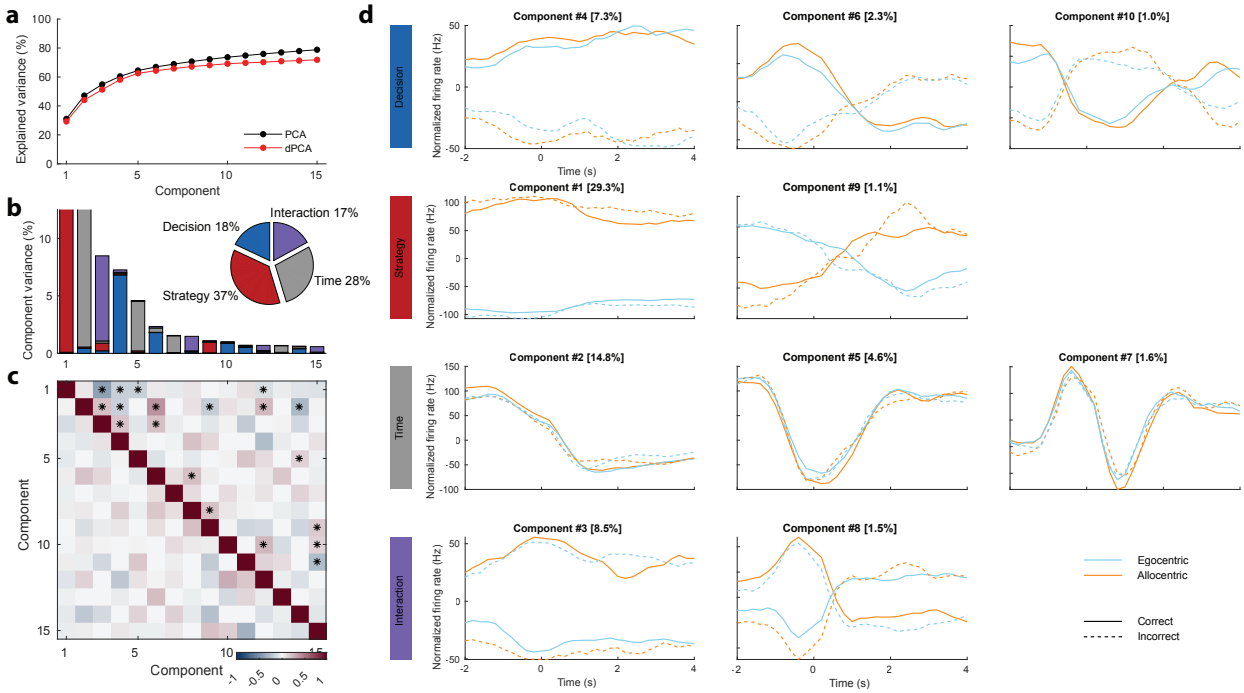
Supplementary Figure S5. Fully observable models require the cues all the time. (**a**) Minigrid environment showing full view size with cue removal with agent at the decision point. (**b**) Task performance with cue removal after the agent reaches the decision point for both hcDQN and hcDRQN trained with full observability. Dotted line represents performance of partial view models.



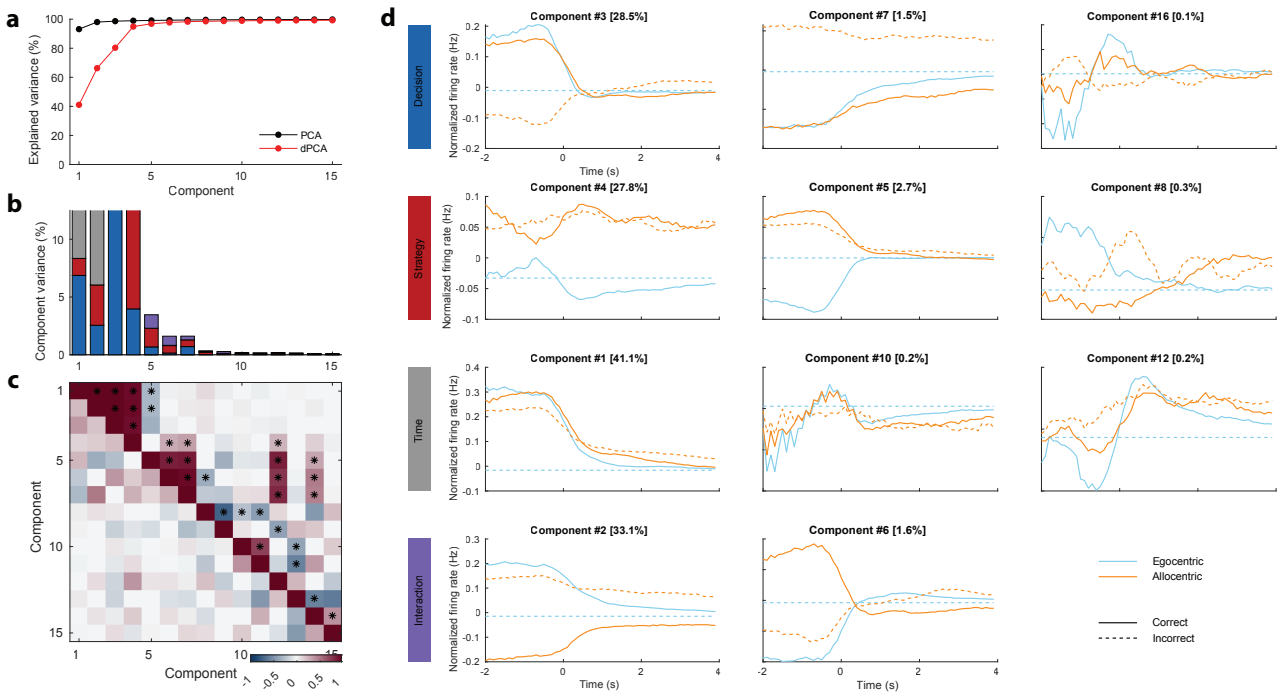
Supplementary Figure S6. Demixed PCA (dPCA) of the hcDRQN model in a maze environment without cues. Compared to the version with cues (Fig. 3A, first row), the separation of the components for strategy and decision is less clear, suggesting that cues play a critical role in enhancing the model's ability to differentiate between these task-related variables.



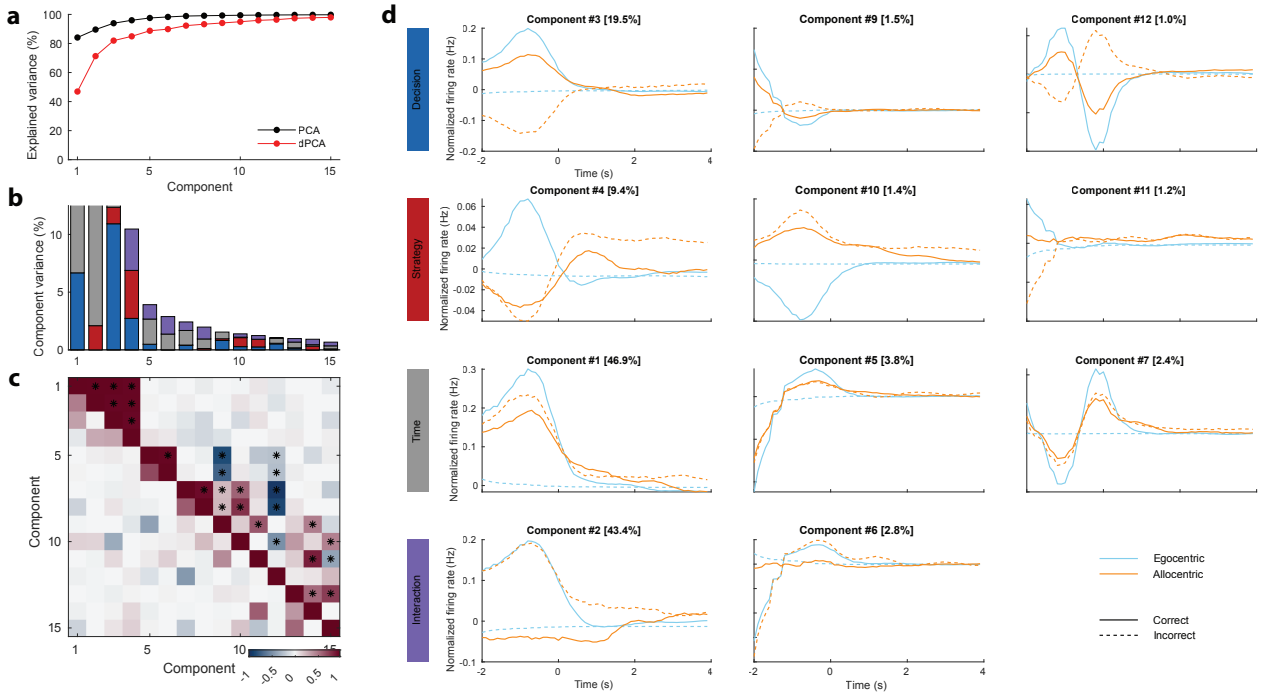
Supplementary Figure S7. Outcome, strategy and temporal neural dynamics in RL agents with full view. (**a**) Demixed PCA components corresponding to Decision - Correct vs Incorrect, Strategies - Allocentric and Egocentric. Qualitatively, full view hcDRQN shows similar trends for decision and strategies, but fails to capture the time component. Full view hcDQN presents mixing activity for strategy components. (**b**) Percentage of explained variance for different components.



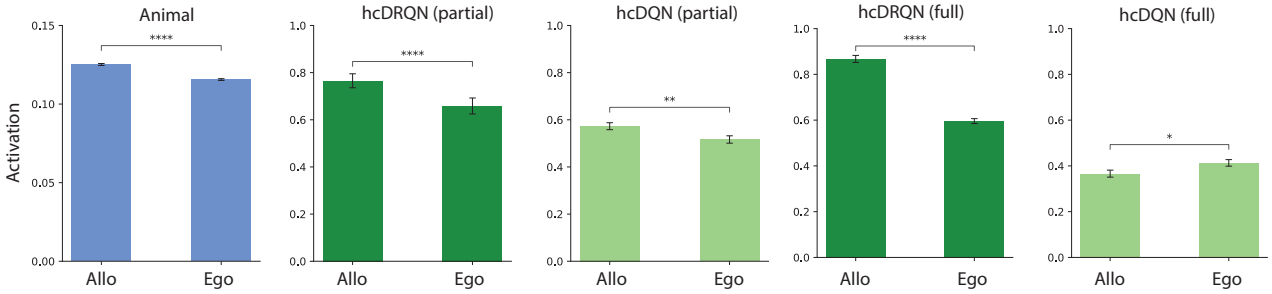
Supplementary Figure S8. Demixed PCA with all the components for animal data. (**a**) Cumulative explained variance of PCA (black) against dPCA (red). (**b**) Variances explained by each demixed principal component. In the pie chart, the total data variance is divided per task-specific variable. (**c**) 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 two strategy components; third row: first three time components; last row: first two decision/strategy interaction components. Figure produced using code made available by (59) (follows a similar structure to the figures available in the original dPCA paper).



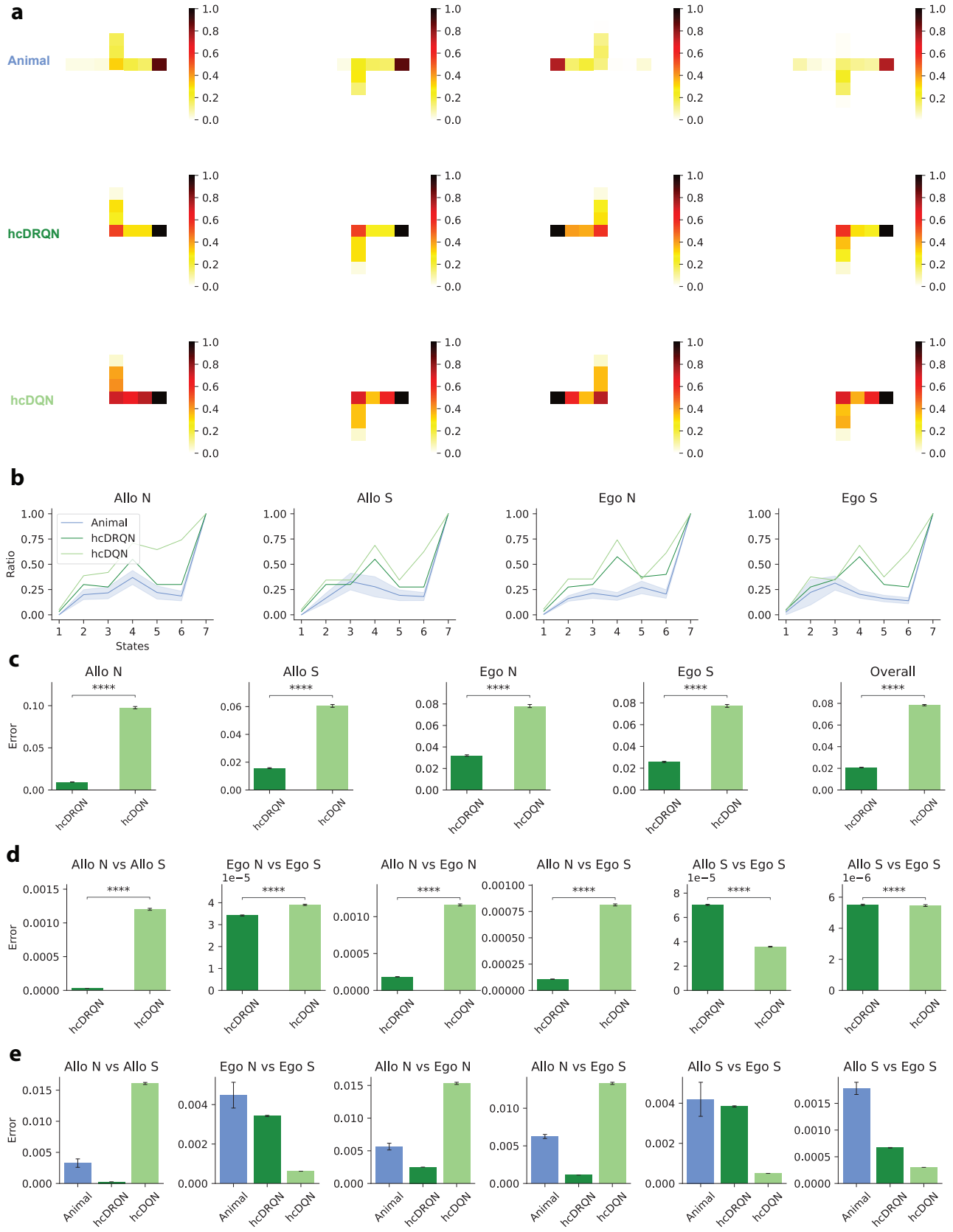
Supplementary Figure S9. Example of typical demixed PCA with all components for hcDRQN. (**a,b,c,d**) As in the previous figure. This is shown for one seed only due to computational reasons, but the results are consistent across seeds.



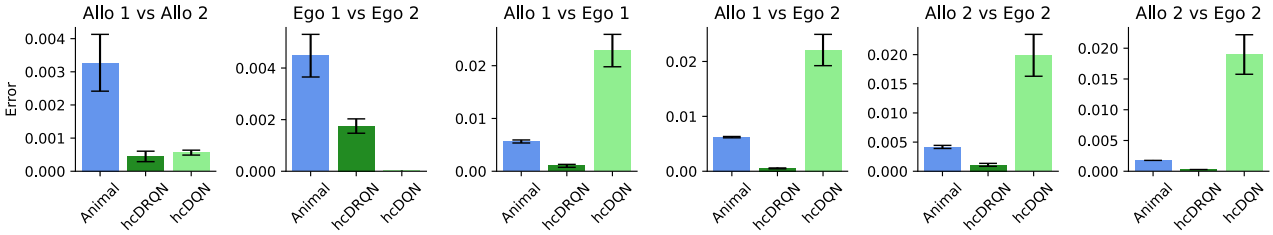
Supplementary Figure S10. Example of typical demixed PCA with all components for hcDQN. (**a,b,c,d**) As in the previous figure. This is shown for one seed only due to computational reasons, but the results are consistent across seeds.



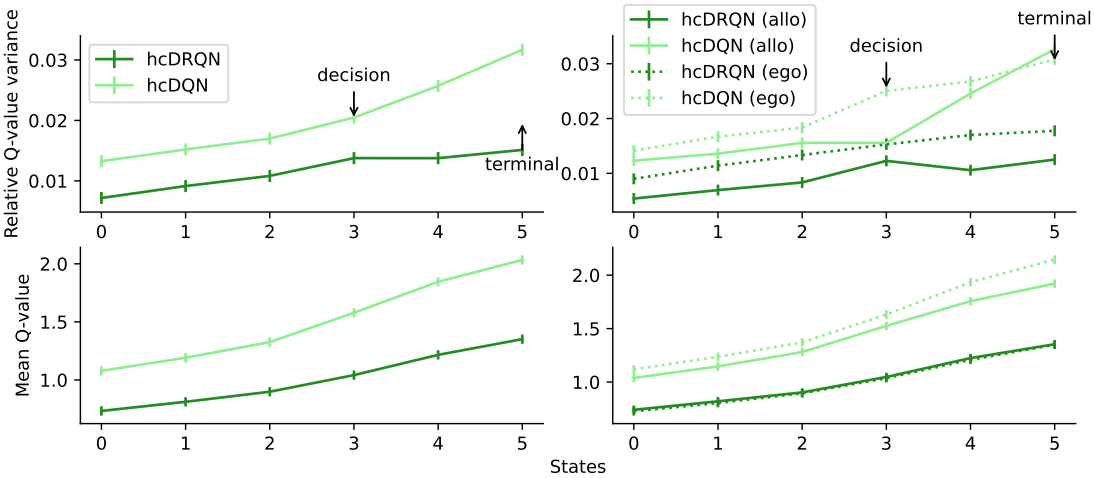
Supplementary Figure S11. Task-specific neural activity. Comparing neural activity between allocentric vs egocentric tasks shows that in both animal and partial view models, allocentric activity is higher. In the full view models, hcDRQN follows the previous pattern while hcDQN shows higher activity in egocentric tasks.



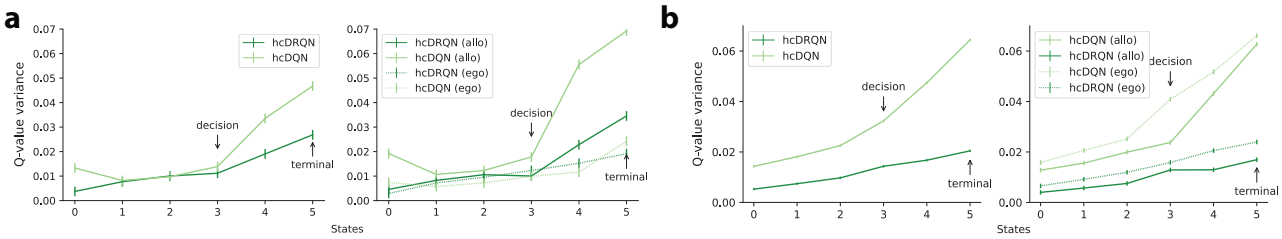
Supplementary Figure S12. Trajectory maps for RL models trained with full observability. (**a**) We repeat the same analysis as done in Fig. 4 with full view models. (**b**) Time spent on each state normalised to the time spent on the final (terminal) state in models and animals. (**c**) Error between agents and animals for each strategy shows that hcDRQN better captures animal behaviour. (**d**) Error between models and animals 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.



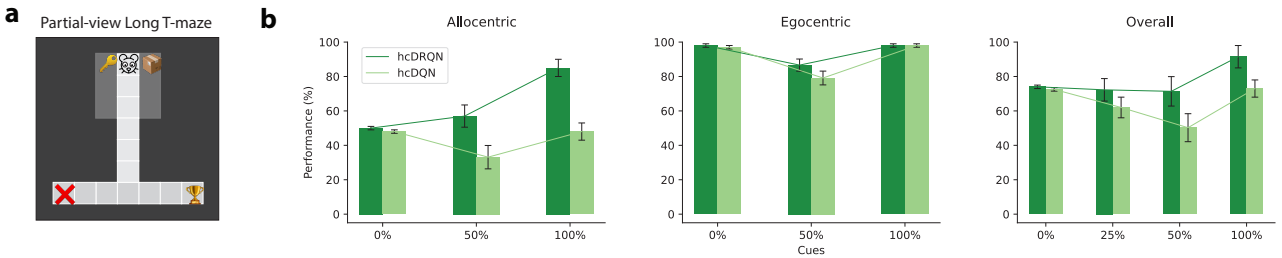
Supplementary Figure S13. Error between animal and model trajectory-occupancy maps (cf. Fig. 4). Error was calculated between all possible task-pairs for both animal and models (1 refers to North and 2 refers to South starting position).



Supplementary Figure S14. State-dependent action values for full view model. Top: Relative Q-value variance, given by the variance over Q-values for each state divided by the mean Q-value for each state, shows that hcDQN has higher overall relative variance compare to hcDRQN. Bottom: Average Q-values.



Supplementary Figure S15. Q-value variance for partial and full view models. (**a**) Partial view: the Q-value variance given by the variance over Q-values for each state. (**b**) Full view's Q-value variance.



Supplementary Figure S16. Cue removal with long T-maze and partial view. (**a**) Partial view in a maze with a long corridor. (**b**) Allocentric and egocentric task performance.