

¹ A model of egocentric to allocentric understanding in ² mammalian brains

³ Benigno Uria^{*,1}, Borja Ibarz^{*,1}, Andrea Banino¹, Vinicius Zambaldi¹, Dharshan Kumaran¹,
⁴ Demis Hassabis¹, Caswell Barry², Charles Blundell¹

⁵¹DeepMind

⁶²University College London

⁷*Equal contribution

⁸ In the mammalian brain, allocentric representations support efficient self-location and flex-
⁹ible navigation. A number of distinct populations of these spatial responses have been iden-
¹⁰tified but no unified function has been shown to account for their emergence. Here we de-
¹¹veloped a network, trained with a simple predictive objective, that was capable of map-
¹²ping egocentric information into an allocentric spatial reference frame. The prediction of
¹³visual inputs was sufficient to drive the appearance of spatial representations resembling
¹⁴those observed in rodents: head direction, boundary vector, and place cells, along with the
¹⁵recently discovered egocentric boundary cells, suggesting predictive coding as a principle
¹⁶for their emergence in animals. Strikingly, the network learned a solution for head direc-
¹⁷tion tracking and stabilisation convergent with known biological connectivity. Moreover,
¹⁸like mammalian representations, responses were robust to environmental manipulations,
¹⁹including exposure to novel settings. In contrast to existing reinforcement learning ap-
²⁰proaches, agents equipped with this network were able to flexibly reuse learnt behaviours
²¹—adapting rapidly to unfamiliar environments. Thus, our results indicate that these repre-
²²sentations, derived from a simple egocentric predictive framework, form an efficient basis-
²³set for cognitive mapping.

²⁴ **Introduction** Animals navigate easily and efficiently through the world, learning quickly about
²⁵new places they encounter¹. Yet artificial agents struggle with even simple spatial tasks requiring
²⁶self-localisation and goal directed behaviours. In mammals, the neural circuits supporting spa-
²⁷trial memory have been studied extensively. Empirical work has generated a detailed knowledge
²⁸about populations of spatially modulated neurons – place², grid³, and head direction⁴ cells rep-
²⁹resent body position and direction of facing, while border responsive neurons encode immediate
³⁰environmental topography^{5,6}. Current thinking sees these networks as components of a cognitive
³¹map thought to provide an efficient spatial basis, allowing the structure of novel environments
³²to be learnt rapidly and subsequently supporting flexible navigation, such as short-cuts, detours,
³³and novel routes to remembered goals^{2,7}.

³⁴ Surprisingly, there are significant gaps in our understanding. In particular, it is unclear how
³⁵these populations interact to support spatial behaviour and how they are themselves derived and
³⁶updated by incoming sensory information. Fundamentally, we lack a unifying computational
³⁷account for the emergence of allocentric (world centred) representations from egocentric (self
³⁸centred) sensory experience^{8,9}. This poses a problem for neuroscientists, who seek simplifying
³⁹normative accounts of the brain, and for AI practitioners who aim to build systems with naviga-
⁴⁰tional abilities that match those of animals.

⁴¹ Historically, theoretical approaches to this problem largely presented hand-coded designs fo-

42 cused on a single layer of the biological network^{10,11}. A smaller number of models have presented
43 a more extensive description of hippocampal networks¹², including the process by which repre-
44 sentations might be learnt¹³. Notably a potential architecture for ego–allocentric transformation¹⁴
45 successfully anticipated the existence of egocentric border responsive neurons^{15–17}. Still, in al-
46 most all cases existing models are the product of extensive human oversight.

47 Recently, deep neural networks have provided a normative explanation for the relationship
48 between two allocentric representations – grid patterns emerge when predicting idealised place
49 cell inputs^{7,18–20} – adding to experimental²¹ and developmental²² evidence that points to place-
50 cells as an input to downstream grid-cells. Similar approaches have seen considerable success
51 as models of the initial feed-forward processing conducted by the mammalian visual system^{23,24}.
52 However, although impressive, these models have tended to be limited in scope – often being
53 focused on a single biological circuit, taking either idealised neural activity as an input or being
54 constrained to representations bound to a single reference frame.

55 Here we show that a deep learning model trained to predict visual experience from self-motion
56 is sufficient to explain the hierarchy of egocentric to allocentric representations found in mam-
57 mals, bridging the gap from visual experience to place cells. In particular, we identify head-
58 direction cells, border cells with both ego and allocentric responses, and place cells. Critically,
59 unlike other recent models of spatial representations^{25,26}, we do so without the need for any allo-
60 centric inputs, using only information that is plausibly available to the brain. Thus, our network
61 links this diverse array of cells to a simple predictive framework, it poses hypotheses about the
62 circuit dynamics that produce these responses, and embodies a model linking sensory informa-
63 tion, neural activity, and spatial behaviour. Like their biological counterparts, the firing fields of
64 these units are shaped by the surrounding visual cues and boundaries, but retain their character-
65 istics across different environments. When embodied in artificial agents, this redeployment of
66 existing responses supports rapid adaptation to novel environments, enabling effective navigation
67 to remembered goals with limited experience of the test enclosure – demonstrating the flexible
68 transfer of skills at which animals excel but artificial systems have struggled.

69 **Spatial Memory Pipeline** The hippocampal formation has been characterised as a predictor-
70 comparator, comparing incoming perceptual stimuli with predictions derived from memory^{27–29}.
71 With this in mind we trained a deep neural network to predict the forthcoming visual experi-
72 ence of an agent exploring virtual environments. Specifically, we developed a model composed
73 of a modular recurrent neural network (RNN) with an external memory – the 'Spatial Memory
74 Pipeline' (see supplementary text, Supplementary Fig 1). At each time-step visual inputs enter-
75 ing the pipeline were compressed using a convolutional neural network and compared to previous
76 embeddings stored in the slots of a memory store, the best matching being most strongly reacti-
77 vated. The remainder of the pipeline was trained to predict which visual memory slot would be
78 reactivated next (Fig 1A). Predictions were generated solely on the basis of self-motion informa-
79 tion, provided differentially to RNN modules as angular and linear velocity, in addition to visual
80 corrections from previous steps. There was no direct pathway from detailed visual features to lat-
81 ter parts of the pipeline – visual information being communicated only by the activation of slots
82 in the memory stores (Fig 1B). Thus the Spatial Memory Pipeline forms a predictive code^{30,31},
83 anticipating which future visual slot will be activated, based solely on self-motion.

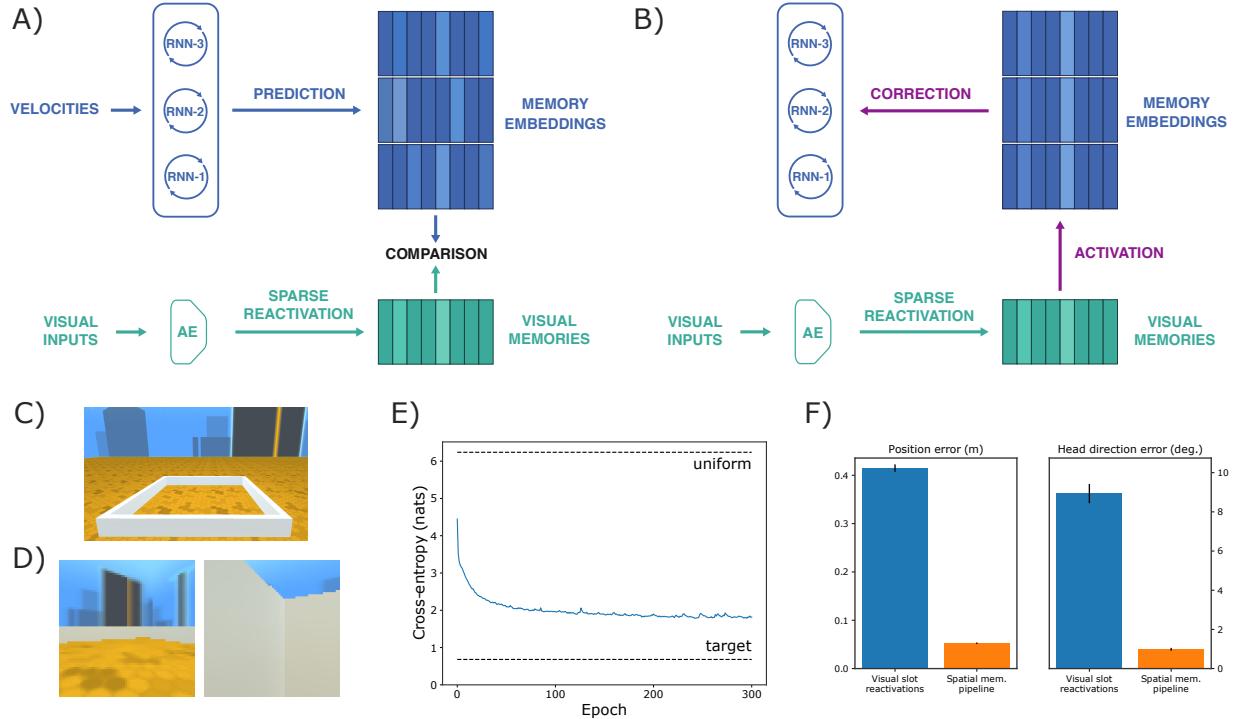


Figure 1: A, B) Computational diagrams for the Spatial Memory Pipeline. **A)** In the prediction step, a set of RNNs anticipate the reactivation of past visual memories by integrating egocentric velocities – AE denotes autoencoder. **B)** In the correction step, the reactivated visual memories correct the RNNs' state to reduce accumulated errors. **C)** External view of white square enclosure with distal cues. **D)** Agent's view from the middle of the enclosure (left) and close to a wall (right). **E)** Prediction loss along training. The model reduces the uncertainty in visual reactivations by 80% with respect to a uniform distribution. **F)** Average decoding error of position and head direction from visual slot activations and spatial memory pipeline RNNs, black bars show one standard error of the mean.

84 **Emergence of allocentric representations** In our first unsupervised-learning experiment, the
85 Spatial Memory Pipeline was trained in a simulated square environment (2.2 by 2.2 meters) re-
86 sembling those used in rodent studies – plain white walls with visible distal cues (Fig 1C, D),
87 using a rat-like motion model³². After training, the network was able to accurately predict the re-
88 activation of visual memories (80% loss reduction with respect to uniform distribution, Fig 1E),
89 effectively integrating self-motion information to anticipate the visual scene. To understand how
90 the network performed this task we inspected the activity profile of units in the modular RNNs.
91 These units exhibited a range of spatially stable allocentric responses strongly resembling those
92 found in the mammalian hippocampal spatial memory system, including head-direction (HD)
93 cells, boundary vector cells (BVCs), and place cells, as well as egocentric boundary vector cells
94 (egoBVCs) (Fig 2, see Methods). Indeed, in the trained network, the agent’s position and direc-
95 tion of facing could be accurately decoded from the RNN activations (Fig 1F, see Supplementary
96 Methods). These representations were complementary to one another – each RNN module devel-
97 oped distinct response patterns, dependent on the form of self-motion input it received (Fig 2A-
98 D). Similar results were observed in environments with different geometries (see supplementary
99 text, Supplementary Fig 2). The results persisted for different random initializations of network
100 parameters (see Supplementary Results). In contrast, a single RNN receiving all the velocity
101 inputs did not develop the whole range of representations (see Supplementary Results).

102 The majority of units in the first RNN module (RNN-1), which received only angular velocities
103 and corrections from the visual memories, exhibited activity that was modulated by the agent’s
104 direction of facing (Fig 2A). These responses were strikingly similar to those of head-direction
105 cells^{4,33} – individual units having a single preferred firing direction invariant across spatial loca-
106 tions (Fig 2B). In total 91% (29/32) of units were classified as HD cells (resultant vector length
107 >0.58, 99th percentile of shuffled data, see Methods) with unimodal responses (average tuning
108 width 76.2°) distributed uniformly in the unit circle (Fig 2H, uniform distribution was selected
109 under Bayes Information Criteria (BIC) over mixtures of Von Mises, see Supplementary Meth-
110 ods).

111 The second module (RNN-2) received angular velocity and speed. Here units exhibited more
112 complex patterns of activity (mean resultant vector = 0.02, 0/128 units classified as HD cells),
113 encoding distances to boundaries at a particular heading (Fig 2C). Thus, they appeared to be
114 similar to egocentric BVCs¹⁵⁻¹⁷ which are theorised to play a central role in transforming between
115 ego and allocentric reference frames^{14,34}. To quantify this impression we adopted an egoBVC
116 metric applied previously¹⁷ (see Methods). In this way 52 of 128 units (41%) were identified
117 as egoBVCs (ego-boundary score >0.07, 99th percentile of bin-shuffled distribution, Fig 2I),
118 with the remaining cells displaying mixed patterns of head-direction and egocentric distance to a
119 wall. Rodent egoBVCs exhibit a uniform distribution of preferred firing directions - albeit with a
120 slight tendency to cluster to the animal’s left and right side - while preferred distances have been
121 reported up to 50cm, with shorter range responses being more numerous¹⁷. Analysis of the model
122 egoBVCs (Supplementary Fig 3A-C) revealed a similar pattern of distances, but a cluster of cells
123 responded to a wall directly in front - plausibly reflecting the monocular input and narrower field
124 of view available to the model (60° vs >180° in rodents) ³⁵.

125 In contrast to the first two RNNs, the third (RNN-3) received no self-motion inputs, thus be-
126 ing dependent upon temporal coherence¹³, and corrections from mispredictions as its sole input
127 and learning signal. Units in this layer were not modulated by heading direction (mean resul-

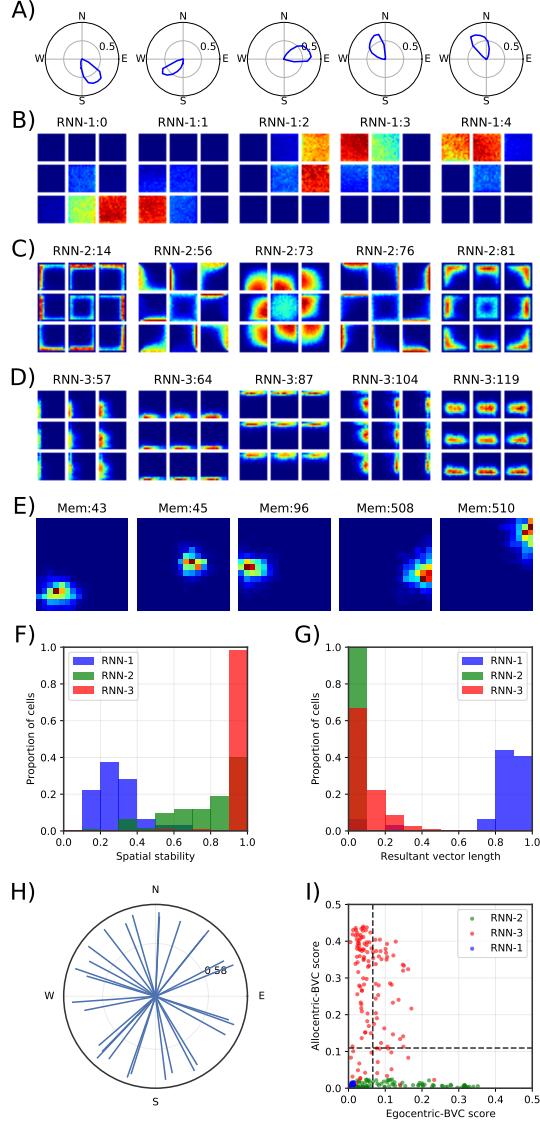


Figure 2: Representations in a spatial memory pipeline trained in the square enclosure shown in Fig 1C. **A)** Polar plot of average activity by heading direction of five units classified as HD cells. **B)** Spatial ratemaps of the units shown in A. Central plot shows average activation for each location across all head directions. The eight plots around each central plot show location-specific activity restricted to times when the heading direction of the agent was in the the corresponding 45° range (e.g. plot located above the central plot shows average activity when the agent was facing in the north direction). **C)** Spatial ratemaps of five units classified as egoBVCs. **D)** Spatial ratemaps of five units classified as BVCs. **E)** Spatial ratemaps of five memory slots reactivated by RNN-3 resembling hippocampal place-cell activity. **F)** Spatial stability of units in each RNN. **G)** Resultant vector length of units in each RNN. **H)** Resultant vector of each unit in RNN-1. **I)** Comparison of egocentric versus allocentric scores for units in each RNN. Dashed lines indicate cell-type classification thresholds.

tant vector = 0.11, 0/128 units classified as HD cells), but were characterised by spatially stable responses (Fig 2F) often extending parallel to particular walls of the environment (Fig 2D). As such, the activity profile of this module was reminiscent of border cells or boundary vector cells (BVCs), which form an allocentric representation of space defined by environmental boundaries^{5,6,36,37}. To assess the units' activity we applied a similar approach to that used for RNN-2 – calculating the resultant vector of the units' activity projected into allocentric boundary space (see Methods). Applying this measure to RNN-3 confirmed that 84% (108/128) of units were classified as BVCs (BVC score >0.11, 99th percentile of shuffled distribution) – 37% (48/128) met the criteria for egoBVCs, but 37 of those 48 had a higher BVC score than egoBVC score (Fig 2I). The same approach applied to RNN-2 identified 0 of 128 units as BVCs. Analysis of the preferred firing directions and distances of the model BVCs revealed a uniform distribution of directions (Supplementary Fig 3E).

In the brain, BVCs are hypothesised to be a core contributor to the spatial activity of place cells³⁶. Consistent with this, BVC-like responses are found in mEC, afferent to hippocampus, as well as within the hippocampus proper^{5,6,37}. In our model, the allocentric representations of RNN-3 were stored in the external memory slots as a second set of targets - being reactivated at each time step by comparison to the current state of the RNN-3. The activity of these memory slots strongly resembled place cells with sparse, spatially localised responses (average spatial field size of 0.21 m^2 SD:0.26 to a total environment area of 4.84 m^2) that were stable across trajectories, and independent of head direction (see Fig 2E and Supplementary Fig 4).

Responses to geometric transformations Simple manipulations of the test environment made without retraining the model, such as rotating distal cues or stretching the enclosure, produced commensurate changes in the receptive fields of spatially modulated cells - resembling the responses of rodent spatial neurons after similar transformations^{38,39} (Supplementary Fig 5, Supplementary Fig 6, and Supplemental Results). Similarly, an extra barrier inserted into a familiar environment becomes a additional target for BVC and egoBVC responses (Supplementary Fig 7, and Supplementary Results), causing some place fields to duplicate and others to be suppressed^{5,6,37}.

Learned attractor dynamics in the head-direction system In the mammalian brain, head-direction tuning is widely believed to originate from a neural ring attractor which constrains activity to lie on a 1D manifold - this provides a mechanism to integrate head turns when appropriately connected with neurons responsive to angular velocity⁴⁰. In our model, when RNN-1 was instantiated separately and provided only with angular velocities (see Methods), it displayed a single activity bump that closely tracked the apparent head direction for several hundred steps, effectively integrating angular velocity over periods that greatly exceeded the duration of training trajectories (Fig 3A). Projecting the observed activity vectors onto their first two principal components revealed that they resided close to a 1-D circular manifold (Fig 3B), a characteristic associated with linear continuous attractor systems. Indeed, with zero angular velocity inputs, random initialisations of the network state quickly converged to a set of point attractors (Fig 3C-D, see Methods).

In contrast, positive or negative angular velocities drove the state to periodic orbits along 1D cyclic attractors (Fig 3E-F). In the mammalian head direction system these dynamics⁴ are hypothesised to be supported by a double ring network with each ring having counter rotated

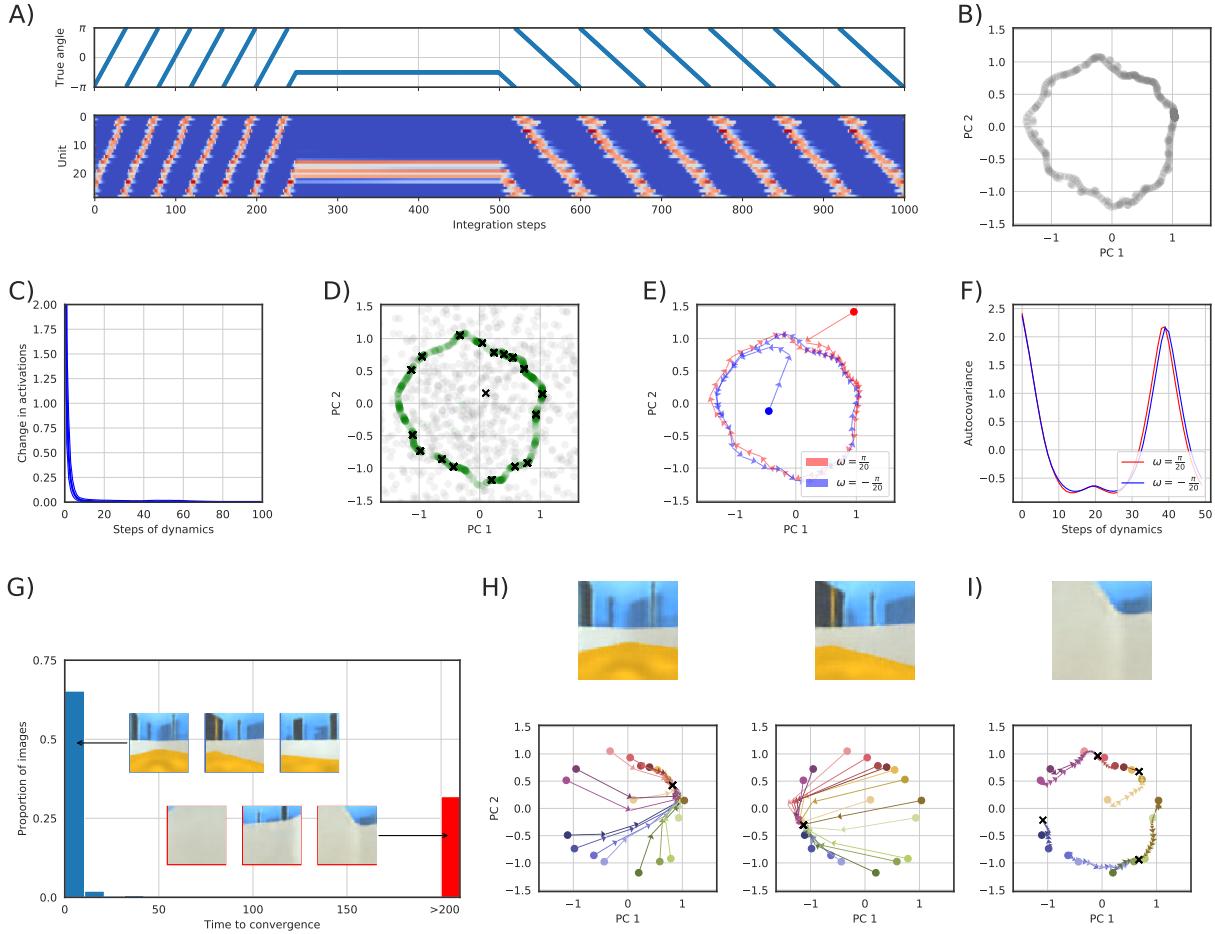


Figure 3: Head-direction cell dynamics. **A)** Activation of each unit over 1000 steps of blind integration. Top: True integrated angle. Bottom: Activation of HD-units in RNN-1 through time (units ordered by the phase of their resultant vectors). For the first 250 steps the angular velocity was set to $\pi/20$ rad/step, the following 250 steps 0 rad/step, the remaining 500 steps $-\pi/40$ rad/step. **B)** Projection of the 1000 activity vectors shown in A onto their first two principal components. **C)** Euclidean distance between successive states in 1000 trajectories with zero angular velocity and random initialisation (see Methods). Thick blue shows the average over one-thousand different initialisations, the thin lines show the 5-th and 95-th percentiles. **D)** Grey shows the projection of the random initial states used in the simulations of panel C onto the space of panel B. Green shows the state after 10 steps of dynamics with zero angular velocity. Black crosses show the states after 1000 steps. All the trajectories converged to a discrete set of attractor states. **E)** Projection on the space of panel B of two 50-step trajectories with constant angular velocity input, one positive and one negative, each starting from a random initialisation. **F)** Autocovariance of states as a function of time lag computed over 1000 randomly initialised 200-step trajectories with positive and negative velocities. **G)** Histogram of steps to convergence to a single attractor state for 512 correction images. Insets show examples with quick convergence (blue) and some that do not converge to a single state (red). **H)** Several example state trajectories, initialised from the attractor states in panel D (different colours), providing visual inputs with unambiguous cues (top) for 200 steps. Black crosses show the final states. **I)** Same as H with ambiguous visual input of a wall.

tuning - a similar solution has been observed in the fly⁴¹. A strikingly similar connectivity can be learned by simple RNN formulations⁴² (Supplementary Fig 8, Supplementary Results). Angular velocity integration alone is not sufficient to maintain a stable representation of head direction – visual cues are required to reset the activity of units and correct for drift. To investigate how our model incorporates visual information in its representation of heading, we simulated the input of visual corrections (512 images from the training environment) and zero angular velocity (see Methods). The majority of images (352/512) resulted in network dynamics with a single attractor point per image, regardless of the initial network state (Fig 3G). These images typically display unambiguous distal cues (Fig 3H). The remaining images (160/512) did not result in a single attractor point, possibly reflecting the geometrical symmetries of the environment. Interestingly, upon visual examination, these images usually lack distal cues and correspond to views of walls and corners (Fig 3I).

Spatial characteristics are preserved across environments Next, we sought to establish how stable the different representations in our model were between environments that changed both in terms of their geometry and visual composition. To this end the network was trained concurrently in three environments inspired by empirical studies – a square, a circle, and a trapezoid, all with distinct distal cues, floor and wall textures (Fig 4A). The same RNNs were used throughout but different external memory stores were provided for each environment. Despite the different training environments, we again found similar proportions of head direction, egoBVC, BVC, and place cells in the network (see Supplementary Results). Indeed, although the environments were visually and geometrically distinct, the classification of units did not materially change between them: 84% (27/32) RNN-1 units were classified as HD cells in all three environments, similarly 78% (100/128) of RNN-2 units were classified as egoBVCs, and 61% (78/128) of RNN-3 units as BVCs. Notably, like biological cells⁴, head direction units maintained their relative angular tuning between environments, while egoBVCs and BVCs maintained their preferred distance to walls (Supplementary Fig 9, see Methods).

Representation re-use allows transfer of behaviour In animals, allocentric representations such as head-direction cells and BVCs are rapidly redeployed in novel settings - likely as a result of being constrained to low-dimensional manifolds which decouple the integration of self-motion from high-dimensional perceptual experience^{5,43}. We hypothesised that this self-consistency is an adaptive characteristic allowing spatial behaviour learned in one environment to be quickly transferred to novel environments.

To test this proposal we incorporated the Spatial Memory Pipeline into a deep reinforcement learning agent⁴⁴ (see Methods) trained to find an invisible goal in a 4-room enclosure (Fig 4B), a task inspired by the classic Morris water maze⁴⁵. This task captured two levels of localisation: locally within a room (where in the room?) and globally across rooms (which room?). When the agent reached the goal, it received a reward and was teleported to a random location in the enclosure - to maximise reward it had to reach the goal as many times as possible within each episode. Three visually different enclosures were used simultaneously for training (Fig 4B). As before, separate memory stores were used for each enclosure but RNNs were shared.

Once trained, the agent reached a high degree of proficiency in the task, routinely following the shortest path to the goal (better than human performance). Inspection of the memory pipeline confirmed that the RNNs contained spatial representations consistent across the three visually

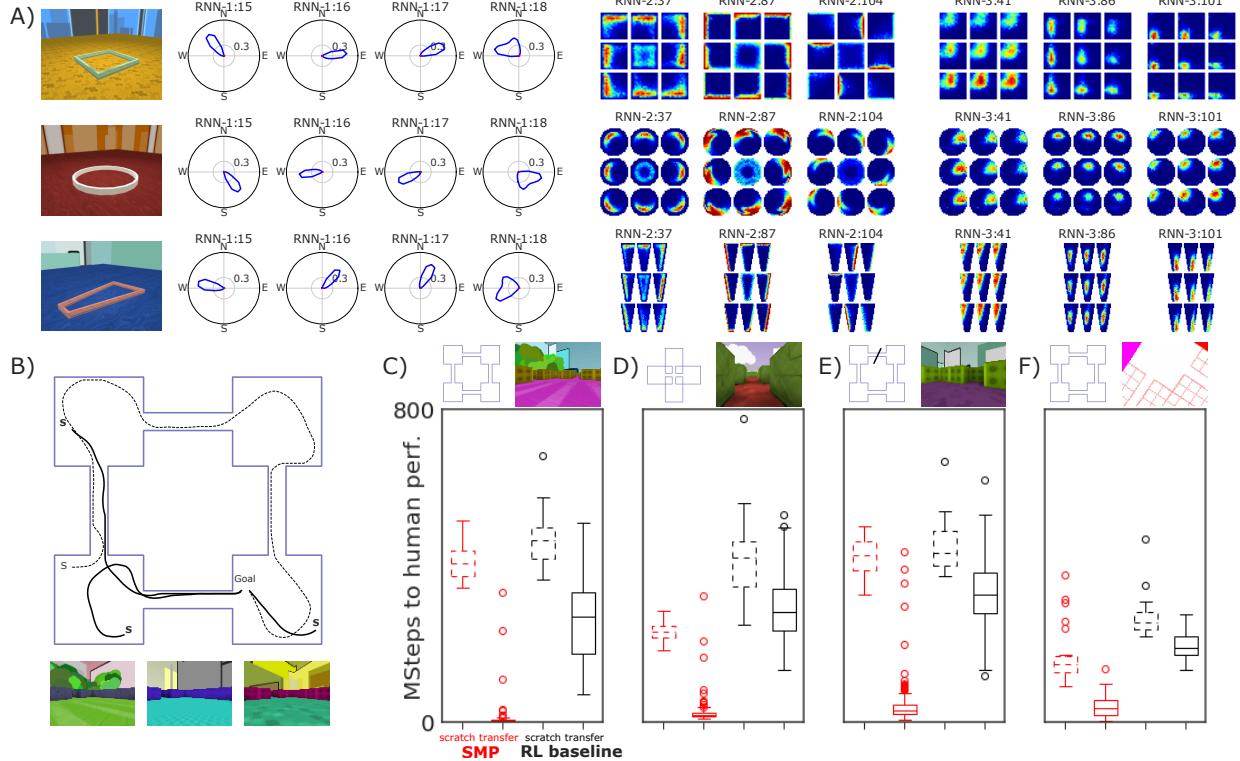


Figure 4: **A)** Stability of spatial representations across three environments, one per row. Left to right: view of the enclosure; allocentric orientation tuning of four units classified as HD cells; spatial ratemap of three units classified as egoBVCs; and spatial ratemap of three units classified as BVCs. Same plotting conventions as Fig 2. **B)** Top, reinforcement learning task in a 4-room enclosure. Spawning points marked with an S. The initial trajectory to the reward is dotted, later trajectories solid. Bottom, visual input from the 3 training enclosures. **C)** Distribution of training times needed to reach human performance in the reinforcement learning task. In red Spatial Memory Pipeline agents, in black baseline agents. Dashed indicates training from scratch, solid indicates transfer to a visually different environment from an agent previously trained in the environments shown in B. On top, schematic of enclosure and example visual input. **D)** Same as C but transfer is to an enclosure with a different layout (cross-shaped). **E)** Same as C but transfer is to a different task, where in each episode a randomly chosen corridor is blocked. **F)** Same as C but transfer is to a conceptual representation of the environment.

214 different environments (Supplementary Fig 12). To test our claim that these allocentric repre-
215 sentations provide efficient bases for transfer, the trained agent was placed in a visually different
216 4-room enclosure (Fig 4C), an enclosure with different room layout and visual aspect (Fig 4D),
217 a variation of the 4-room task where corridors were blocked at random (Fig 4E), and a 4-room
218 enclosure presented conceptually to the agent as a two-dimensional plan view (Fig 4F). In all
219 transfer tasks the agent reached human performance significantly faster than an agent trained
220 from scratch (80x, 12x, 13x and 4x respectively). In contrast, if generic recurrent networks were
221 used instead of the Spatial Memory Pipeline, the agent was still able to learn the initial task but
222 achieved only modest performance gains in transfer (1.7x, 1.5x, 1.3x and 1.3x) (Supplementary
223 Fig 11). Thus, the spatial responses present in the Spatial Memory Pipeline's RNN support rapid
224 generalisation to novel settings, an ability that is commonly observed in mammals but has been
225 an elusive target for artificial agents.

226 **Discussion** The Spatial Memory Pipeline described by our model develops an array of sensory-
227 derived allocentric spatial representations strongly resembling those found in the mammalian
228 brain. These representations are learnt solely using a predictive coding principle from egocentric
229 visual perception and self-motion inputs. Like the brain, the network does not have access to
230 allocentric information, and differs markedly from prior work in which brain-inspired agents
231 were limited by the need for allocentric input during training^{19,20,25,46}. In contrast, the forms of
232 egocentric representation used here are known to contribute to the activity of spatially modulated
233 neurons in the hippocampal region^{47,48} while the predictive framework agrees with empirical
234 and theoretical work linking the hippocampus with anticipation of the future^{7,49,50}. Our core
235 contribution then is the provision of a normative model that explains the appearance of known
236 mammalian spatial representations solely from interaction with the sensory world. This model
237 poses hypotheses about the interactions between these representations, the neural dynamics that
238 generate them, and their role in spatial navigation.

239 The cell-types that we identified were segregated between different sub-networks, being de-
240 fined by the forms of information available to them: angular velocity and visual corrections for
241 head-direction cells; angular velocity, speed, and visual corrections for egoBVCs; and only visual
242 corrections for BVCs. External memories reactivated by BVC inputs strongly resembled place
243 cell activity. This framework implicitly captures several properties of the mammalian spatial sys-
244 tem - distinct functional populations combining to form a sparse, spatially precise representation
245 of self-location similar to that seen in the hippocampus. Like their biological counterparts, spa-
246 tial responses in the RNNs were robust to environmental manipulations, responding predictably
247 to the changes made to local cues while retaining their fundamental firing correlates between en-
248 tirely different enclosures. Our work demonstrates that these representations constitute a flexible
249 and efficient spatial basis, available to be quickly redeployed in a novel setting - sufficient to sup-
250 port rapid transfer learning in a reinforcement learning agent. On the other hand, activity in the
251 memory stores was, by design, entirely distinct between environments, emulating hippocampal
252 remapping, a phenomenon that occurs in response to significant manipulations of environmental
253 cues^{51,52}.

254 In the case of head-direction cells in the first RNN, the activity of units resulted from learnt
255 attractor dynamics – both in the integration of velocities and in the anchoring to visual cues. Low
256 dimensional manifolds of this form are widely accepted to provide the preeminent account of the
257 biological head-direction system, explaining the fixed relative offset seen in biological head-

258 direction cells as well as in our model^{43,53}. Notably a similar head-direction attractor circuit is
259 instantiated topographically in flies^{54,55}.

260 A full understanding of the spatial circuits in the mammalian brain will require resolving
261 the functional dependence between different representations. In our model the relationship be-
262 tween the head-direction and BVC systems is determined by their joint association to visual
263 memories (Supplementary Fig 10) - thus they rotate coherently to follow changes in the angular
264 location of a familiar distal cue⁵⁶ (Supplementary Fig 5). Conversely, in different enclosures
265 with non-overlapping sets of visual memories, biological BVCs have been observed to decou-
266 ple from the head-direction system while retaining their relative tuning⁵. This mechanism of
267 head-direction and BVC dependence contrasts with the hypothesis that postulates BVC activity
268 as the result of direct conjunction of head-direction and egoBVC activations^{14,36}. Hence, new
269 experimental studies on animals jointly recording the head-direction and BVC systems across
270 enclosures that elicit global remapping will be required to ascertain the mechanisms that give
271 rise to BVCs.

272 In conclusion, we show that an artificial network replicates both the form and function of
273 a biological network central to self-localisation and navigation. A simple training objective is
274 sufficient, in this case, to approximate the development of neural circuits that have been shaped
275 by both selective pressure applied over evolutionary time as well as by direct experience during
276 an animal's life.

277 **Methods**

278 **Materials and methods**

279 **The Spatial Memory Pipeline** The *Spatial Memory Pipeline* consists of two submodules where
280 learning occurs independently. The first level is a visual feature extraction network that reduces
281 the dimensionality of visual inputs. Following visual feature extraction, there is a level of in-
282 tegration through time that encodes estimates of the agent’s location by integrating egocentric
283 velocities. We refer to this integration level as *Memory-Map*. It is composed of a fast-binding
284 slot-based associative memory that plays the role of an idealised hippocampus, and a set of re-
285 current neural networks that receive as inputs egocentric velocity signals and play the role of an
286 artificial neocortex.

287 **Visual-feature extraction** The first level in the *Spatial Memory Pipeline* hierarchy employs a
288 convolutional autoencoder⁵⁷ to reduce the dimensionality of the raw visual input (Supplementary
289 Fig 1A). The autoencoder is composed of an encoder network that transforms the raw visual in-
290 put, $\mathbf{y}^{(raw)}$ into a low-dimensional visual encoding vector, $\mathbf{y}^{(enc)}$, that summarises the structure
291 of the input. To learn such compressed code, $\mathbf{y}^{(enc)}$ is passed through a decoder that produces a
292 reconstruction of the original input, $\hat{\mathbf{y}}^{(raw)}$. All of the encoder and decoder parameters are optim-
293 ised to minimise the reconstruction error $|\mathbf{y}^{(raw)} - \hat{\mathbf{y}}^{(raw)}|^2$. Extended Data Table 4 summarises
294 the autoencoder configuration, see Supplementary Methods for more details.

295 **Memory-maps** A Memory-map module consists of two components: a set of R recurrent neural
296 networks, $\{F_r\}_{r \in 1 \dots R}$, that integrate egocentric velocities, and a slot-based associative memory,
297 \mathcal{M} , that binds upstream inputs, \mathbf{y} , to RNN state values:

$$\mathcal{M} \equiv \left\{ (\mathbf{m}_s^{(y)}, \mathbf{m}_{1,s}^{(x)} \dots \mathbf{m}_{R,s}^{(x)}) \right\}_{s \in 1 \dots S}, \quad (1)$$

298 where the variable s indexes the different memory slots, while the super-index denotes the type
299 of information (y for upstream inputs, and x for the state of RNNs).

300 At each time step, t , the current upstream input, \mathbf{y}_t , reactivates the memory slots with the
301 most similar contents (Supplementary Fig 1B). We formalise this concept using a categorical
302 distribution over memory slots:

$$P_{react}(s | \mathbf{y}_t, \mathcal{M}) \propto e^{\beta \mathbf{y}_t^\top \mathbf{m}_s^{(y)}}, \quad (2)$$

303 where β is a positive scalar that is automatically adjusted to match an entropy target, H_{react}
304 –enforcing sparse activity (see Supplementary Methods).

At every time step, the states corresponding to each of the predictive RNNs, $\{\mathbf{x}_r\}_{r \in 1 \dots R}$ where
 $\mathbf{x}_r \in \mathbb{R}^{N_r}$, are updated using the current egocentric velocities, $\mathbf{v}_{r,t}$:

$$\hat{\mathbf{x}}_{r,t} = F_r(\mathbf{x}_{r,t-1}, \mathbf{v}_{r,t}). \quad (3)$$

305 The type of velocity inputs used for each RNN is reported in Extended Data Table 5. For in-
306 stance, in our unsupervised experiments the first RNN takes as input the sine and cosine of the
307 angular velocity, ω , whereas the second RNN takes the same inputs as the first RNN and also the

308 linear speed, s . Finally, the third RNN receives no velocity inputs and relies only of temporal
309 correlations.

These RNN states also induce a predictive categorical distribution over memory slots (Supplementary Fig 1B):

$$P_{pred}(s \mid \mathcal{M}, \hat{\mathbf{x}}_{1,t} \dots \hat{\mathbf{x}}_{R,t}) \propto \prod_{r=1}^R e^{\pi_r \hat{\mathbf{x}}_{r,t}^\top \mathbf{m}_{r,s}^{(x)}}, \quad (4)$$

310 where π_r are positive scalar parameters that determine the relative importance of each RNN in
311 the predictive distribution and its entropy.

312 Model parameters are optimised to minimise the cross-entropy from this predictive distribution
313 to the distribution of visually driven reactivations:

$$L = \sum_{s=1}^S P_{react}(s \mid \mathbf{y}_t, \mathcal{M}) \log P_{pred}(s \mid \mathcal{M}, \hat{\mathbf{x}}_{1,t} \dots \hat{\mathbf{x}}_{R,t}). \quad (5)$$

314 Thus, the model is trained so P_{pred} anticipates the distribution of memory reactivations P_{react} .

315 Note that at time t , P_{pred} has not yet received any information from the current upstream
316 visual input, \mathbf{y}_t , forcing the RNNs to use previous egocentric velocity inputs to produce good
317 predictions.

However, in order for the RNN representations to be allocentrically grounded, and to correct for the accumulation of integration errors, the RNNs must incorporate positional and directional information from upstream visual inputs as well. This correction step should not be performed at every time step, or the integration of velocities would be unnecessary; in our experiments, it was performed at random timesteps with probability $P_{correction} = 0.1$. The incorporation of visual information is implemented by calculating a correction code for each RNN:

$$\tilde{\mathbf{x}}_{r,t} = \sum_{s=1}^S w_{s,t} \mathbf{m}_{r,s}^{(x)}, \text{ where } w_{s,t} = \frac{e^{\gamma \mathbf{y}_t^\top \mathbf{m}_s^{(y)}}}{\sum_{s'=1}^S e^{\gamma \mathbf{y}_t^\top \mathbf{m}_{s'}^{(y)}}}, \quad (6)$$

318 where γ is a positive scalar parameter that determines the entropy of the distribution of weights
319 (Supplementary Fig 1C). Each $\tilde{\mathbf{x}}_{r,t}$ can be thought of as the result of a weighted reactivation of
320 the RNN memory embeddings by the current visual input, \mathbf{y}_t .

In steps when corrections are provided, these correction codes are input to correction RNN cells, G_r , that combine the predictive state with the correction code (Supplementary Fig 1C):

$$\mathbf{x}_{r,t} = G_r(\hat{\mathbf{x}}_{r,t}, \tilde{\mathbf{x}}_{r,t}). \quad (7)$$

While in steps when no correction is available, the input to the next time step is simply the predicted state (Supplementary Fig 1D):

$$\mathbf{x}_{r,t} = \hat{\mathbf{x}}_{r,t}. \quad (8)$$

321 Model parameters $\Theta = \left\{ \gamma, \left\{ F_r, G_r, \mathbf{m}_{r,\cdot}^{(x)}, \pi_r \right\}_{r \in 1 \dots R} \right\}$ are trained by gradient descent on the
322 prediction loss of equation (5).

323 Note that the contents of the memory store corresponding to the upstream inputs, $\mathbf{m}^{(y)}$, are
324 not part of the optimised parameters, Θ , as they are used to calculate the target distribution.
325 Therefore, in order to fill the slots of \mathcal{M} with memories of upstream inputs, at every timestep with
326 small probability, $P_{storage}$, a slot s is chosen at random and assigned $(\mathbf{m}_s^{(y)}, \mathbf{m}_{1,s}^{(x)} \dots \mathbf{m}_{R,s}^{(x)}) :=$
327 $(\mathbf{y}_t, \mathbf{x}_{1,t} \dots \mathbf{x}_{R,t})$.

328 Due to the sparse activation of memory slots there is low interference between memories,
329 i.e. modifying a $\mathbf{m}_r^{(x)}$ only has a local effect. For this reason these associative memory embed-
330 dings can be optimised at a higher learning rate (see Extended Data Table 2), resulting in fast
331 binding of new memories once the RNN dynamics have been learned.

332 **Architecture used in our experiments** The experimental results in this paper used the archi-
333 tectures described in Extended Data Tables 4-5.

334 **Visual autoencoders** The visual-encoding vectors $\mathbf{y}^{(enc)}$ were obtained by inputting RGB im-
335 ages, $\mathbf{y}^{(raw)}$, of the environment to an encoder network with 4 convolutional layers with bias and
336 ReLU output nonlinearities, chained to a flat output layer (Supplementary Fig 1A). The architec-
337 ture parameters are summarised in Extended Data Table 4. The encoders used for the reinforce-
338 ment learning experiments had greater capacity than the encoders used for the unsupervised-
339 learning experiments, since they were trained to encode a variety of environments, as explained
340 below.

341 In our experiments, the visual autoencoders were not affected by the prediction loss in equa-
342 tion (5). Therefore, the autoencoders were trained beforehand, separately from the rest of the
343 model. During the autoencoder training, images were passed through the convolutional layers
344 and flat output layer to produce the vector of visual encodings, $\mathbf{y}^{(enc)}$. This vector was then
345 passed through a decoder network of de-convolutional layers (with transposed architecture from
346 the encoder but independent parameters) to produce a reconstruction, $\hat{\mathbf{y}}^{(raw)}$, of the input im-
347 age (Supplementary Fig 1A). All parameters in the autoencoder were optimised to minimise the
348 mean square distance between the input and reconstructed images.

349 For all experiments, the autoencoders were trained by minibatch gradient descent using an
350 Adam optimiser with learning rate 10^{-4} .

351 For unsupervised experiments, training minibatches consisted of 50 images taken at random
352 from the same trajectories used to train the spatial memory pipeline. Training was complete
353 after 200,000 minibatches. We trained a separate autoencoder for each environment (square,
354 circular, and trapezoid cages). Note that the manipulated environments in Supplementary Fig 5,
355 Supplementary Fig 7, and Supplementary Fig 6, used the same autoencoder as their original,
356 non-manipulated square environment.

357 For reinforcement learning experiments the training minibatches consisted of 216 images
358 mixed at random from trajectories generated in the training and testing environments using a
359 rat-like motion model (see below). Training was complete after 200,000 minibatches. A separate
360 autoencoder was trained for the 4-room enclosure represented conceptually (Fig 4F), with the
361 same training procedure and architecture.

362 **Rat-like motion model** The rat-like motion model used to generate trajectories for unsupervised
363 experiments (and for training the visual autoencoders for reinforcement learning experiments,
364 see above) was based on published work³², with parameters specified in Extended Data Table 1.
365 At each time step a linear velocity was generated from a Rayleigh distribution and a rotational
366 velocity from a Gaussian distribution. However, if the trajectory was closer than 3 cm to a wall
367 at an angle narrower than 90°, a deterministic rotation was added that turned it to continue to
368 move parallel to the wall, and the linear velocity was reduced by a factor of 4. Additionally, the
369 trajectory switched between periods of movement (linear velocity Rayleigh parameter 0.13) and
370 periods of pure rotation (linear velocity Rayleigh parameter 0.0). The move and stop periods
371 lasted for an exponentially distributed number of steps with mean 50. The trajectories used to
372 train the spatial memory pipeline were sampled every 7 simulation steps. The angular and linear
373 velocity signals fed to the RNNs in unsupervised experiments were computed dividing the total
374 spatial displacement and angular rotation of the agent over each 7-simulation-step interval by the
375 simulation time of those 7 steps. The corresponding visual embedding came from the image of
376 the environment at the given position and orientation, with the camera 20 cm above and parallel
377 to the ground. Each trajectory consisted of 500 samples, corresponding to 3500 simulation steps.

378 **Environment dimensions for unsupervised experiments** For the single-environment unsupervised-
379 learning experiments a square enclosure 2.2 m in side was used. The multi-environment exper-
380 iments, additionally, included a circular enclosure of radius 1.5 m, and an isosceles trapezoidal
381 enclosure with altitude 4.4 m and parallel sides 2.2 m and 0.75 m long. Single-environment
382 experiments were also carried out in the circular enclosure of 1.5 m radius (see Supplementary
383 Results). The height of the walls in all environments was 0.25 m.

384 **Sigmoid-LSTM and Sigmoid-Vanilla** As our aim was to compare the activation of artificial
385 RNNs to firing rate data from biological neurons, we limited the activations of all RNNs to
386 positive values. In order to do this, we modified the commonly used *LSTM* and *Vanilla-RNN* cells
387 by substituting their *tanh* output non-linearity for a *sigmoid*. We called these cells *Sigmoid-LSTM*
388 and *Sigmoid-Vanilla-RNN* respectively. As it learns faster, we used *Sigmoid-LSTM* in all our
389 experiments; with the exception of the supplementary HD double ring analysis (see Methods),
390 where we used a *Sigmoid-Vanilla-RNN* for ease of analysis. In a Vanilla-RNN the mapping from
391 the current state to the next is extremely simple. Namely, the current vector of cell activations,
392 \mathbf{h}_t , depends on the previous activations, \mathbf{h}_{t-1} , and inputs, \mathbf{x}_t , through equation:

$$\mathbf{h}_t = \sigma(\mathbf{W}\mathbf{h}_{t-1} + \mathbf{V}\mathbf{x}_t + \mathbf{b}), \quad (9)$$

393 where σ is the element-wise sigmoid function, and we refer to \mathbf{W} as the weight matrix of
394 dynamics and \mathbf{V} as the weight matrix of inputs.

395 **Sparse reactivation** The inverse-temperature parameter in the reactivation of memory slots, β
396 in equation (2), is constantly regulated so the entropy of P_{react} matches the hyperparameter H_{react}
397 (0.5 nats in unsupervised-learning experiments, 1.0 nats in reinforcement learning). In order to do
398 this, β was parameterised as $\beta = e^{\beta_{logit}}$ and after every trajectory, β_{logit} was increased/decreased
399 by 0.001 when the average entropy of P_{react} was lower/higher than H_{react} .

400 **Training details** The training parameters for the unsupervised experiments are summarised in
401 Extended Data Table 2. The training was implemented with the TensorFlow platform. We used
402 back-propagation through time (BPTT) with an unroll length of 50 trajectory steps; since trajec-
403 tories consisted of 500 steps, each trajectory produced 10 BPTT unrolls. Batch size was 32, each
404 batch consisting of unrolls from different trajectories.

405 The single-environment experiments used the Adam optimisation algorithm. Different learning
406 rates were used for the memory slot contents and the rest of the network parameters: since
407 the target distribution over slots was very sparse, we could apply a larger learning rate for the
408 memory slot contents.

409 For reasons of ease of implementation, the multi-environment experiments used the RMSProp
410 optimisation algorithm. Each of the three environments trained in a separate process, updating
411 the shared parameters synchronously with Distributed TensorFlow. The shared parameters were
412 the weights and biases of the RNNs (both the F_r prediction RNNs and the G_r correction RNNs).
413 The memory contents \mathcal{M} and the entropy-scaling variables γ and π_r were separate for each
414 environment.

415 Similarly to previous work¹⁹, dropout was used in the RNN outputs when predicting the mem-
416 ory reactivations.

417 **Assessment of attractor dynamics in the head-direction system** For the evaluation of inte-
418 gration dynamics in Fig 3A the state of RNN-1 was initialised to $m_{1,1}^{(x)}$, and angular integration
419 (equation 3) iterated for a thousand steps using the angular velocity $\frac{\pi}{20}$ rad/step for 250 steps,
420 0 rad/step for the following 250 steps, and $-\frac{\pi}{40}$ rad/step for the final 500 steps. Visual input
421 was not given to the model. Fig 3A shows the evolution of the RNN state, $x_{1,t}$, ordering its units
422 by the phase of its resultant vector.

423 The principal-component space displayed in Fig 3B was calculated by taking the two principal
424 eigenvectors of the covariance matrix of unit activity in Fig 3A. Repeated activity vectors were
425 discarded from the calculation of the mean and covariance matrix.

426 For the assessment of fixed attractor points in Fig 3C-D, the state of each unit in RNN-1 was
427 independently initialised by drawing a sample from a Gaussian with four times the standard
428 deviation of its activations in Fig 3A. A thousand different random initialisations were used and
429 each run for 1000 steps of angular integration (equation 3) inputting 0 rad/step angular velocity.

430 For the assessment of periodic orbits in Fig 3E-F, the same one thousand initialisations were
431 used, but each was run for 200 steps of angular integration (equation 3) using either a clockwise
432 $-\frac{\pi}{20}$ rad/step or counter-clockwise $\frac{\pi}{20}$ rad/step angular velocity. The 50 first steps of two arbi-
433 trary trajectories with opposing velocities were shown in Fig 3E. Each of the two autocovariance
434 curves in Fig 3F was calculated using all 1000 trajectories.

435 For the assessment of point-attractors under the influence of visual inputs in Fig 3G-I, we took
436 512 random images from the training dataset, and run the model for 200 steps of prediction and
437 correction. Each of these 200 steps consisted of a step of visual correction, equations 7 and 8,
438 followed by zero angular-velocity integration, equation 3. For every image we run 18 trajectories
439 initialised from each the fixed points found in Fig 3D. We considered that a single fixed-point

440 had been reached at a particular time step if the maximum distance among the 18 trajectories was
441 less than 0.1 in the PCA space. The bottom plots in Fig 3H-I show the trajectory of each of these
442 18 initialisations for three examples of visual inputs.

443 **Preservation of spatial characteristics across environments** We trained the *Spatial Memory*
444 *Pipeline* simultaneously in three environments, sharing the RNN parameters across the environments
445 but keeping a separate memory store for each. The separation of memories simulates
446 hippocampal remapping, since there is no correspondence between the slots for the different
447 environments. The RNN units, on the other hand, behaved coherently across the environments,
448 giving further credence to their role as analogues of HD, egoBVC and BVC cells. Supplementary
449 Fig 9 summarises the stability of these representations across the three environments (1, square;
450 2, circle; 3, trapezoid). The relative preferred angles between HD units in RNN-1 were preserved
451 with great accuracy between environments (Extended Data Fig 9A), i.e., the head direction system
452 rotated as a whole without change in functionality. In contrast, the relative directional tuning
453 of BVC units in RNN-3, although not entirely random (Kuiper test rejected the uniform distribution
454 for the angle differences, p -value < 0.01 for all environment pairs), displayed high variability
455 across units (Extended Data Fig 9C). This means that BVC units in our model, under remapping,
456 are not locked to the head direction system, unlike the case without remapping (see rotation manipulations,
457 Supplementary Fig 5). The preferred angles of EgoBVC units in RNN-2 stayed the same in different environments (inter-environment angular differences clustered tightly around
458 zero, Extended Data Fig 9B), as expected for a system of egocentric coordinates. Distance tuning
459 of egoBVCs was very stable across environments (Extended Data Fig 9D), while BVC distance
460 tuning, although significantly preserved (mean across units of the absolute difference between
461 environments significantly smaller than the mean across shuffled pairs, p -value < 10^{-4}), again
462 showed more variability (Extended Data Fig 9E).

464 **Environment for reinforcement learning experiments** We assessed the performance of reinforcement learning agents in the DeepMind Laboratory platform⁵⁸. The layout of the 4-room
465 enclosures (Fig 4B) consisted of four 5 x 5-tile rooms (1.25 x 1.25 m assuming an agent speed
466 of 15 cm/s) linked by four 5 x 1 corridors. The enclosure was surrounded by a sky-box at infinity - so as to provide directional but not distance information - textured with buildings, clouds
467 and trees, which provided distal cues. There were no special markings on the walls or floors of
468 the enclosure, making all rooms and corridors identical except for their relative orientation with
469 respect to the distal cues. At the start of each episode the agent (described below) was placed in
470 a random location and was required to explore in order to find an unmarked goal, paralleling the
471 task of rodents in the classic Morris water maze. When the goal was reached, the agent received a
472 reward of 10 points and was teleported to a random location in the enclosure. The goal remained
473 fixed for the length of the episode, so a well-trained agent would first explore the enclosure in
474 order to discover the goal location and then, after each teleportation, orient itself to return to
475 the same goal as quickly as possible. The length of the episodes was 3000 agent steps (12000
476 environment frames, see below).

479 The agent received observations in the form of camera input (96 x 72 pixels), rotation velocity,
480 and egocentric translation velocity (components parallel and perpendicular to the agent's
481 orientation).

482 The agent could start at any position in the enclosure. The action space was discrete (six

483 actions) but afforded fine-grained motor control (that is, the agent could rotate in small increments,
484 accelerate forwards or backwards, or effect rotational acceleration while moving forwards). Agent actions were repeated for 4 consecutive environment frames⁴⁴; one actor step
485 consisted therefore of 4 environment steps.

487 The visual appearance of the environment was determined by the floor, wall and distal cues.
488 We used 8 different sets of textures, three of them for training and the other five to evaluate
489 transfer. Transfer was evaluated on the same layout as training (Fig 4C), on a cross-shaped
490 layout with 5 x 5-tile rooms linked by two 5 x 1 corridors (Fig 4D), in the same training layout
491 where in each episode one corridor at random was blocked by a wall (Fig 4E), and, finally, on the
492 same training layout provided to the agent not from a first-person camera view but as a top-down
493 schematic representation centered on the agent position and rotated according to the agent's head
494 direction (Fig 4F). In this schematic representation, each of the 4 boundaries of the layout were
495 filled with a different colour in lieu of distal cues. The image size was 96 x 72 pixels, as in the
496 camera-input experiments.

497 **Reinforcement learning agent architecture and training** We used importance-weighted actor-
498 learner agent (IMPALA)⁴⁴ to learn the task. IMPALA is an actor-critic setup to learn a policy
499 distribution π over the available actions and a baseline function V^π . Our IMPALA setup con-
500 sisted of 96 actors, repeatedly generating trajectories of experience, and one learner that used the
501 trajectories sent by actors to learn the policy. At the beginning of each episode actors updated
502 their local policy-network, value-network and *Spatial Memory Pipeline* parameters to the latest
503 learner parameters and, at the end of each episode, sent the trajectory of observations, rewards,
504 states, and policy distributions to the learner. The learner continuously updated its policy, value
505 and *Spatial Memory Pipeline* parameters via back-propagation through time (BPTT) on batches
506 of 64 100-step chunks of trajectories collected from different actors. Both the actor-critic pa-
507 rameters and the *Spatial Memory Pipeline* parameters were optimised simultaneously, but with
508 respect to separate losses. The *Spatial Memory Pipeline* loss is just as described previously in
509 the unsupervised experiments, and determines the gradients for the *Spatial Memory Pipeline* pa-
510 rameters. The actor-critic loss is composed of three terms: a policy gradient loss to maximise
511 expected advantage, a baseline value loss to predict the episode returns, and an entropy bonus to
512 prevent premature convergence. The possible policy lag between the actors and the learner was
513 corrected in the policy gradient and baseline losses with V-trace⁴⁴. We used Adam⁵⁹, a stochastic
514 gradient descent optimiser, for both the *Spatial Memory Pipeline* and the actor-critic loss. A
515 schematic of the agent network is displayed in Supplementary Fig 11A.

516 The inputs to the actor-critic network were the previous time-step reward and one-hot encoded
517 action, the concatenation of the current outputs of all the *Spatial Memory Pipeline*'s RNNs
518 (current allocentric code), and the the concatenation of the outputs of all the *Spatial Memory*
519 *Pipeline*'s RNNs observed last time the goal was reached (goal allocentric code, which was set
520 to zero as long as the goal had not been reached in the episode). These inputs were fed to a
521 256-unit LSTM whose output in turn went through a policy MLP (one 256-unit hidden layer
522 with ReLU activation) to produce the policy, and a value MLP (one 256-unit hidden layer with
523 ReLU activation) to predict the value. Note that the actor-critic network did not receive direct
524 visual input to learn the task - only representations from the *Spatial Memory Pipeline*.

525 The inputs to the *Spatial Memory Pipeline*, similarly to the unsupervised learning experiments,

526 consisted of egocentric velocities and vision. Because the DeepMind Laboratory agent has in-
527ertia, it could shift sideways while moving, even though the action set did not allow strafing.
528 Therefore we provided as inputs not only the translation velocity in the heading direction of the
529 agent, but also the component perpendicular to it. These, together with the sine and cosine of the
530 rotational velocity of the agent, formed the 4-component velocity input vector (as opposed to the
531 3-component vector used in unsupervised experiments, that lacked a sideways translation com-
532 ponent). The visual input was the embedding vector (length 128) of a convolutional autoencoder
533 trained offline to reconstruct images of all the training and transfer environments as explained
534 previously, except the top-down schematic view, for which a separate autoencoder was trained.
535 Compared to the unsupervised experiments, the *Spatial Memory Pipeline* for the reinforcement
536 learning experiments used a bigger RNN-2 size and more memory slots. The hyperparameters
537 of the architecture are described in Extended Data Tables 4-5, while the hyperparameters for
538 training are described in Extended Data Table 3.

539 In the training phase, each actor was assigned one of the 3 visually different training envi-
540 ronments. The *Spatial Memory Pipeline* had correspondingly three sets of memory stores, and
541 experiences coming from each environment were channelled automatically to the corresponding
542 bank. The RNN parameters, in contrast, were shared among all the environments. The agent
543 trained for a combined 2 billion actor steps, then it was tested on the transfer environments. 25
544 different seeds were used for training, and transfer was evaluated once from each of them.

545 We evaluated the trained agents in each of the transfer environments separately. In each trans-
546 fer experiment all the IMPALA actors experienced only the transfer environment, and memory
547 stores were initialised blank. The memory store contents were overwritten at a quicker pace
548 (probability 0.01 per step) at the beginning of the transfer experiments, until full, then continued
549 to be overwritten at the normal rate (probability 0.0001 per step). The rest of the agent parameters
550 started from their values in the trained agent.

551 **Reinforcement learning baselines** To support the hypothesis that it is the *Spatial Memory*
552 *Pipeline* representations that allow the agent to transfer the behaviour to a novel environment
553 with little training, we repeated the transfer experiments replacing the *Spatial Memory Pipeline*
554 with networks of comparable capacity: 1) a generic recurrent network (LSTM) (Supplemen-
555 tary Fig 11B) that integrated visual and velocity inputs, trained with the agent policy loss. 2)
556 A two-layer network (Supplementary Fig 11C) where the first layer consisted of three LSTMs,
557 each integrating the visual and velocity inputs from one of the three training environments, and
558 the second layer consisted of a single LSTM integrating the batched outputs of the first-layer
559 LSTMs; this network paralleled the *Spatial Memory Pipeline* architecture, where each training
560 environment had its own separate memory stores. And 3) a *Spatial Memory Pipeline* identical
561 to the main experiment pipeline, except the visual correction was performed at every time step
562 ($P_{\text{correction}} = 1.0$); this *Spatial Memory Pipeline* did not need to integrate velocities over multiple
563 agent steps in order to predict the slot activations, and consequently did not develop any useful
564 spatial representations. For replacements 2) and 3), transfer was evaluated only on the same
565 layout - but different textures - as training (Supplementary Fig 11D). All baselines were trained
566 to the same performance as the *Spatial Memory Pipeline* agent before transfer. Replacement 1)
567 had somewhat unstable learning in the top-down schematic representation of the environment
568 (Fig 4F). A smaller learning rate was used in this case, see Extended Data Table 3, since it
569 removed the instability while preserving the average learning curves.

570 The distributions shown in Fig 4C-F compare the learning time (in millions of actor steps)
571 to reach human performance in each of the 4 transfer settings either training from scratch or
572 transferring from a trained agent, either with the *Spatial Memory Pipeline*-endowed agent or
573 with baseline network 1). Distributions in Fig 4C-E consist of 25 points for the *scratch* box plots
574 (5 environments x 5 seeds) and 125 points for the *transfer* box plots (5 transfer environments x 25
575 original seeds), distributions in Fig 4F consist of 25 points (1 environments x 25 seeds both for
576 scratch and transfer). For each seed, the time to human performance is measured as the steps
577 required for the average episode reward of the actors to surpass the average human performance
578 and remain above it more than 80% of the following training steps. Human performance was
579 evaluated as the average of 5 episodes played by one of the authors after another 5 episodes of
580 getting acquainted with the task. Computing bootstrapped distributions of the median learning
581 time to reach human performance in each condition yield the following means and standard
582 deviations of the ratios between scratch and transfer medians for the transfer settings in Fig 4C,
583 D, E, F, respectively: for the *Spatial Memory Pipeline*, 85.6 ± 8.9 , 11.4 ± 0.6 , 13.7 ± 1.0 and
584 4.4 ± 0.9 . For baseline network 1), 1.7 ± 0.1 , 1.5 ± 0.1 , 1.3 ± 0.1 and 1.4 ± 0.1 .

585 Supplementary Fig 11D-G shows the average learning curves with the standard error of the
586 mean returns across seeds and environments, comparing the *Spatial Memory Pipeline*-endowed
587 agent with baseline network 1). Additionally, Supplementary Fig 11D shows learning curves for
588 baselines 2) and 3) as described above.

589 The three replacement networks, while able to learn the Morris water-maze task in the training
590 environments, failed to transfer their learned behaviour to visually novel environments (Supple-
591 mentary Fig 11D). After 50 million steps of transfer training in a novel environment, practically
592 all the *Spatial Memory Pipeline* agents had reached human performance, while none of the agents
593 with generic recurrent networks had, needing a number of training steps for transfer comparable
594 to those needed to learn in the first place (>250 million steps). The *Spatial Memory Pipeline* that
595 was purposefully trained with continuous vision ($P_{correction} = 1.0$) to disrupt the development
596 of spatial representations took much longer, demonstrating that it is not an architectural bias, but
597 the spatial representations learned that assist in the transfer of behaviour.

598 **Data and code availability** The data and code used to generate figures 2 and 3 will be available
599 under the open source Apache 2.0 license at the following URL: https://github.com/deepmind/deepmind-research/tree/master/spatial_memory_pipeline.

601 There is Supplemental Information that contains additional results, discussion and methods.

602 **Acknowledgements** We thank Matt Botvinick, Vivek Jayaraman, Tim Behrens, Daan Wierstra, Sergei
603 Lebedev, Christopher Summerfield, Kimberly Stachenfeld, and Charlie Beattie for their valuable advice.

604 **Competing Interests** The authors declare that they have no competing financial interests.

605 **Correspondence** Correspondence and requests for materials should be addressed to Benigno Urias and
606 Charles Blundell (email: buria@google.com, cblundell@google.com).

607 **Author Contributions**

608 Conceived project: B.U, A.B, B.I, C.Ba, C.Bl, D.K, D.H.
609 Contributed ideas to experiments: B.U, C.Ba, B.I, C.Bl, A.B, V.Z, D.K
610 Performed experiments and analysis: B.U, B.I, V.Z, A.B
611 Development of testing platform and environments: B.I, V.Z, B.U, A.B
612 Wrote paper: C.Ba, C.Bl, B.U, B.I, A.B, D.K, D.H
613 Managed project: C.Bl, C.Ba, D.H

614

615

- 616 1. Tolman, E. C. Cognitive maps in rats and men. *Psychological review* **55**, 189 (1948).
- 617 2. O'Keefe, J. & Dostrovsky, J. The hippocampus as a spatial map: Preliminary evidence from
618 unit activity in the freely-moving rat. *Brain research* (1971).
- 619 3. Hafting, T., Fyhn, M., Molden, S., Moser, M.-B. & Moser, E. I. Microstructure of a spatial
620 map in the entorhinal cortex. *Nature* **436**, 801 (2005).
- 621 4. Taube, J. S., Müller, R. U. & Ranck, J. B. Head-direction cells recorded from the postsubiculum
622 in freely moving rats. i. description and quantitative analysis. *Journal of Neuroscience*
623 **10**, 420–435 (1990).
- 624 5. Solstad, T., Boccara, C. N., Kropff, E., Moser, M.-B. & Moser, E. I. Representation of
625 geometric borders in the entorhinal cortex. *Science* **322**, 1865–1868 (2008).
- 626 6. Lever, C., Burton, S., Jeewajee, A., O'Keefe, J. & Burgess, N. Boundary vector cells in the
627 subiculum of the hippocampal formation. *Journal of Neuroscience* **29**, 9771–9777 (2009).
- 628 7. Stachenfeld, K. L., Botvinick, M. M. & Gershman, S. J. The hippocampus as a predictive
629 map. *Nature neuroscience* **20**, 1643 (2017).
- 630 8. Schiller, D. *et al.* Memory and space: towards an understanding of the cognitive map.
631 *Journal of Neuroscience* **35**, 13904–13911 (2015).
- 632 9. Moser, M.-B., Rowland, D. C. & Moser, E. I. Place cells, grid cells, and memory. *Cold
633 Spring Harbor perspectives in biology* **7**, a021808 (2015).
- 634 10. Zipser, D. A computational model of hippocampal place fields. *Behavioral neuroscience*
635 **99**, 1006 (1985).

- 636 11. Sharp, P. E. Computer simulation of hippocampal place cells. *Psychobiology* **19**, 103–115
637 (1991).
- 638 12. Fuhs, M. C. & Touretzky, D. S. Synaptic learning models of map separation in the hip-
639 pocampus. *Neurocomputing* **32**, 379–384 (2000).
- 640 13. Franzius, M., Sprikeler, H. & Wiskott, L. Slowness and sparseness lead to place, head-
641 direction, and spatial-view cells. *PLoS computational biology* **3**, e166 (2007).
- 642 14. Byrne, P., Becker, S. & Burgess, N. Remembering the past and imagining the future: a
643 neural model of spatial memory and imagery. *Psychological review* **114**, 340 (2007).
- 644 15. Wang, C. *et al.* Egocentric coding of external items in the lateral entorhinal cortex. *Science*
645 **362**, 945–949 (2018).
- 646 16. Hinman, J. R., Chapman, G. W. & Hasselmo, M. E. Neuronal representation of environmen-
647 tal boundaries in egocentric coordinates. *Nature Communications* **10**, 2772 (2019).
- 648 17. Alexander, A. S. *et al.* Egocentric boundary vector tuning of the retrosplenial cortex. *Science*
649 *Advances* **6**, eaaz2322 (2020).
- 650 18. Dordek, Y., Soudry, D., Meir, R. & Derdikman, D. Extracting grid cell characteristics from
651 place cell inputs using non-negative principal component analysis. *Elife* **5**, e10094 (2016).
- 652 19. Banino, A. *et al.* Vector-based navigation using grid-like representations in artificial agents.
653 *Nature* **557**, 429 (2018).
- 654 20. Cueva, C. J. & Wei, X.-X. Emergence of grid-like representations by training recurrent
655 neural networks to perform spatial localization. *arXiv preprint arXiv:1803.07770* (2018).
- 656 21. Brandon, M. P., Koenig, J., Leutgeb, J. K. & Leutgeb, S. New and distinct hippocampal place
657 codes are generated in a new environment during septal inactivation. *Neuron* **82**, 789–796
658 (2014).
- 659 22. Tan, H. M., Wills, T. J. & Cacucci, F. The development of spatial and memory circuits in
660 the rat. *Wiley Interdisciplinary Reviews: Cognitive Science* **8**, e1424 (2017).
- 661 23. Yamins, D. L. & DiCarlo, J. J. Using goal-driven deep learning models to understand sensory
662 cortex. *Nature neuroscience* **19**, 356–365 (2016).
- 663 24. Zhuang, C. *et al.* Unsupervised neural network models of the ventral visual stream. *Pro-
664 ceedings of the National Academy of Sciences* **118** (2021).
- 665 25. Whittington, J. C. *et al.* The tolman-eichenbaum machine: Unifying space and relational
666 memory through generalization in the hippocampal formation. *Cell* **183**, 1249–1263 (2020).
- 667 26. George, D. *et al.* Clone-structured graph representations enable flexible learning and vicari-
668 ous evaluation of cognitive maps. *Nature communications* **12**, 1–17 (2021).
- 669 27. Eichenbaum, H. Hippocampus: cognitive processes and neural representations that underlie
670 declarative memory. *Neuron* **44**, 109–120 (2004).

- 671 28. Kumaran, D. & Maguire, E. A. An unexpected sequence of events: mismatch detection in
672 the human hippocampus. *PLoS biology* **4**, e424 (2006).
- 673 29. Lisman, J. & Redish, A. D. Prediction, sequences and the hippocampus. *Philosophical*
674 *Transactions of the Royal Society B: Biological Sciences* **364**, 1193–1201 (2009).
- 675 30. Rao, R. P. & Ballard, D. H. Predictive coding in the visual cortex: a functional interpretation
676 of some extra-classical receptive-field effects. *Nature neuroscience* **2**, 79 (1999).
- 677 31. Friston, K. & Kiebel, S. Predictive coding under the free-energy principle. *Philosophical*
678 *Transactions of the Royal Society B: Biological Sciences* **364**, 1211–1221 (2009).
- 679 32. Raudies, F. & Hasselmo, M. E. Modeling boundary vector cell firing given optic flow as a
680 cue. *PLoS computational biology* **8**, e1002553 (2012).
- 681 33. Ranck, J. B. Head direction cells in the deep cell layer of dorsolateral pre-subiculum in
682 freely moving rats. *Electrical activity of the archicortex* (1985).
- 683 34. Bicanski, A. & Burgess, N. A neural-level model of spatial memory and imagery. *ELife* **7**,
684 e33752 (2018).
- 685 35. Wallace, D. J. *et al.* Rats maintain an overhead binocular field at the expense of constant
686 fusion. *Nature* **498**, 65–69 (2013).
- 687 36. Hartley, T., Burgess, N., Lever, C., Cacucci, F. & O’Keefe, J. Modeling place fields in terms
688 of the cortical inputs to the hippocampus. *Hippocampus* **10**, 369–379 (2000).
- 689 37. Barry, C. *et al.* The boundary vector cell model of place cell firing and spatial memory.
690 *Reviews in the Neurosciences* **17**, 71–98 (2006).
- 691 38. O’Keefe, J. & Burgess, N. Geometric determinants of the place fields of hippocampal neu-
692 rons. *Nature* **381**, 425 (1996).
- 693 39. Barry, C., Hayman, R., Burgess, N. & Jeffery, K. J. Experience-dependent rescaling of
694 entorhinal grids. *Nature neuroscience* **10**, 682 (2007).
- 695 40. Zhang, K. Representation of spatial orientation by the intrinsic dynamics of the head-
696 direction cell ensemble: a theory. *Journal of Neuroscience* **16**, 2112–2126 (1996).
- 697 41. Seelig, J. D. & Jayaraman, V. Neural dynamics for landmark orientation and angular path
698 integration. *Nature* **521**, 186–191 (2015).
- 699 42. Cueva, C. J., Wang, P. Y., Chin, M. & Wei, X.-X. Emergence of functional and structural
700 properties of the head direction system by optimization of recurrent neural networks. *arXiv*
701 *preprint arXiv:1912.10189* (2019).
- 702 43. Redish, A. D., Elga, A. N. & Touretzky, D. S. A coupled attractor model of the rodent head
703 direction system. *Network: Computation in Neural Systems* **7**, 671–685 (1996).
- 704 44. Espeholt, L. *et al.* Impala: Scalable distributed deep-rl with importance weighted actor-
705 learner architectures (2018). 1802.01561.

- 706 45. Morris, R. G. Spatial localization does not require the presence of local cues. *Learning and*
707 *motivation* **12**, 239–260 (1981).
- 708 46. Bicanski, A. & Burgess, N. A neural-level model of spatial memory and imagery. *ELife* **7**,
709 e33752 (2018).
- 710 47. Muller, R. U. & Kubie, J. L. The effects of changes in the environment on the spatial firing
711 of hippocampal complex-spike cells. *Journal of Neuroscience* **7**, 1951–1968 (1987).
- 712 48. Gothard, K. M., Skaggs, W. E., Moore, K. M. & McNaughton, B. L. Binding of hippocampal
713 cal neural activity to multiple reference frames in a landmark-based navigation task. *Journal*
714 *of Neuroscience* **16**, 823–835 (1996).
- 715 49. Blum, K. I. & Abbott, L. A model of spatial map formation in the hippocampus of the rat.
716 *Neural computation* **8**, 85–93 (1996).
- 717 50. Hassabis, D. & Maguire, E. A. The construction system of the brain. *Philosophical Trans-*
718 *actions of the Royal Society B: Biological Sciences* **364**, 1263–1271 (2009).
- 719 51. Hayman, R. M., Chakraborty, S., Anderson, M. I. & Jeffery, K. J. Context-specific acqui-
720 sition of location discrimination by hippocampal place cells. *European Journal of Neuro-*
721 *science* **18**, 2825–2834 (2003).
- 722 52. Leutgeb, S. *et al.* Independent codes for spatial and episodic memory in hippocampal neu-
723 ronal ensembles. *Science* **309**, 619–623 (2005).
- 724 53. Sharp, P. E., Blair, H. T. & Cho, J. The anatomical and computational basis of the rat head-
725 direction cell signal. *Trends in neurosciences* **24**, 289–294 (2001).
- 726 54. Kim, S. S., Rouault, H., Druckmann, S. & Jayaraman, V. Ring attractor dynamics in the
727 drosophila central brain. *Science* **356**, 849–853 (2017).
- 728 55. Fisher, Y. E., Lu, J., D’Alessandro, I. & Wilson, R. I. Sensorimotor experience remaps visual
729 input to a heading-direction network. *Nature* **576**, 121–125 (2019).
- 730 56. Poulter, S., Lee, S. A., Dachtler, J., Wills, T. J. & Lever, C. Vector trace cells in the subiculum
731 of the hippocampal formation. *bioRxiv* 805242 (2019).
- 732 57. Kramer, M. A. Nonlinear principal component analysis using autoassociative neural net-
733 works. *AIChE journal* **37**, 233–243 (1991).
- 734 58. Beattie, C. *et al.* Deepmind lab (2016). 1612.03801.
- 735 59. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint*
736 *arXiv:1412.6980* (2014).

737 **Supplemental Information for A model of egocentric to allocentric understanding in
738 mammalian brains**

739 Benigno Uria^{*,1}, Borja Ibarz^{*,1}, Andrea Banino¹, Vinicius Zambaldi¹, Dharshan Kumaran¹,
740 Demis Hassabis¹, Caswell Barry², Charles Blundell¹

741 ¹DeepMind

742 ²University College London

743 *Equal contribution

744 This section contains:

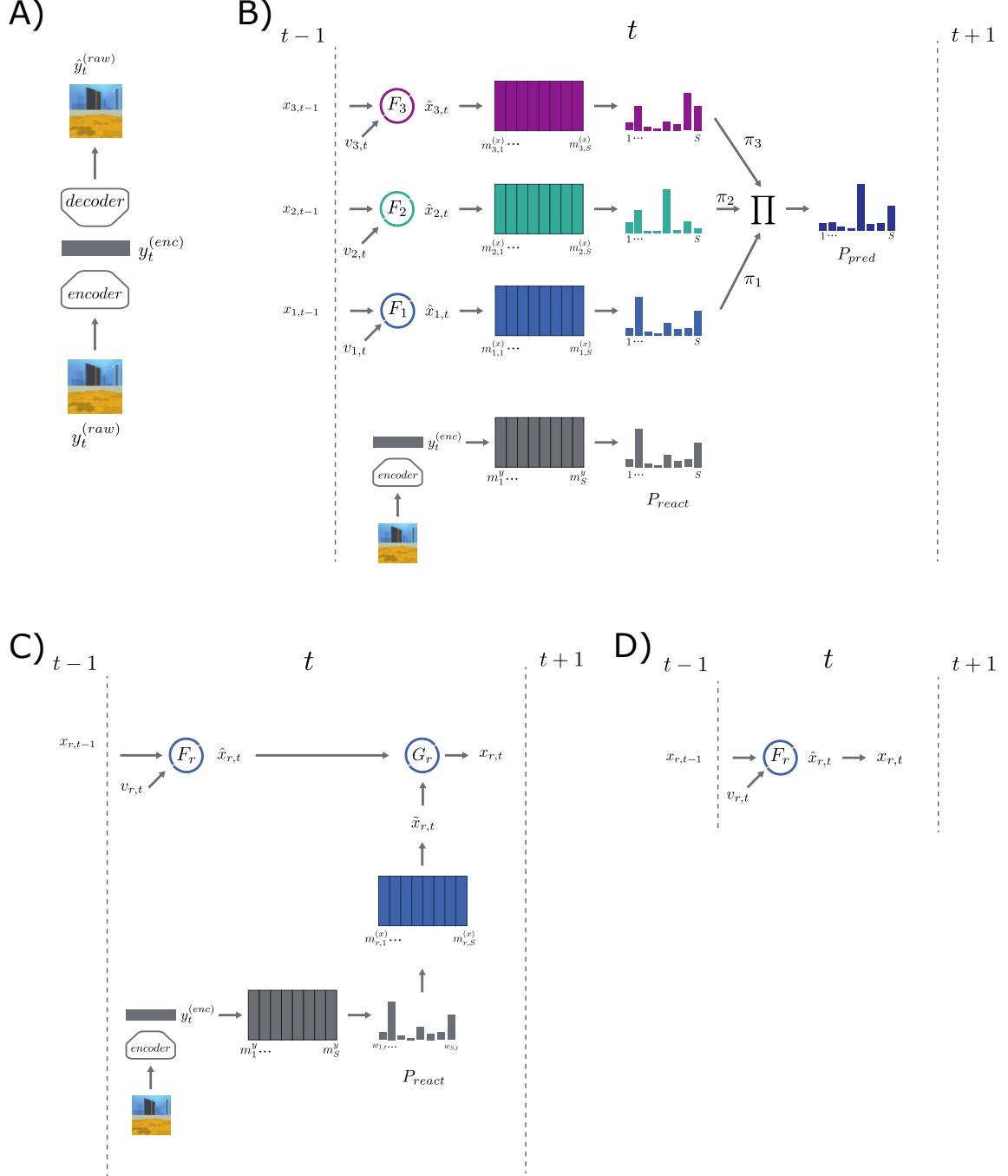
745 • Supplementary Figures

746 • Supplementary Results

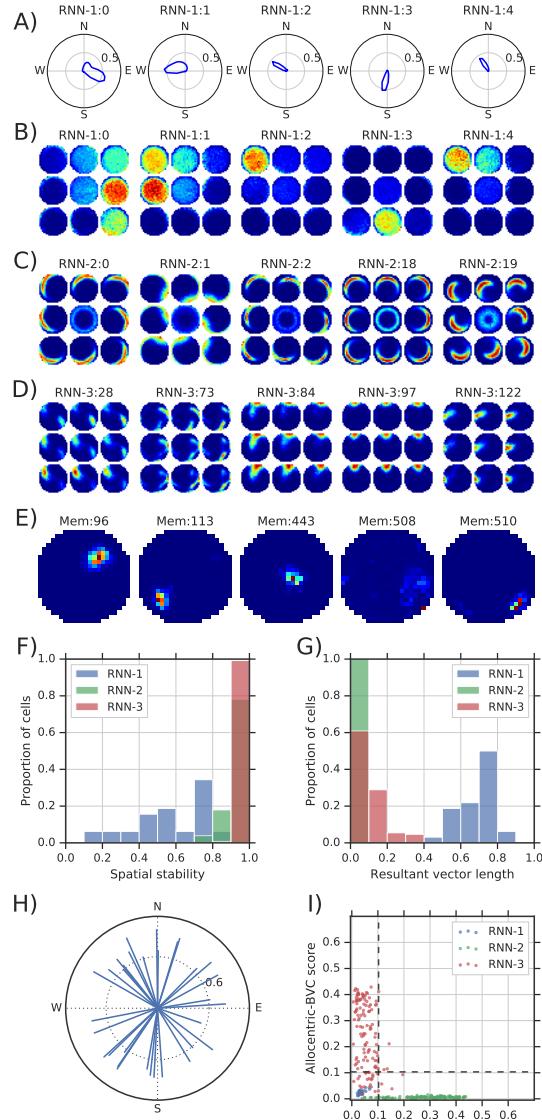
- 747 – Emergent representations in a circular environment
748 – Effects of distal cue rotation on the model’s representations
749 – Effects of enclosure stretch on the model’s representations
750 – Effects of barrier insertion on the model’s representations
751 – Preservation of spatial characteristics across environments
752 – RNN-1 ring attractor analysis

753 • Supplementary Methods

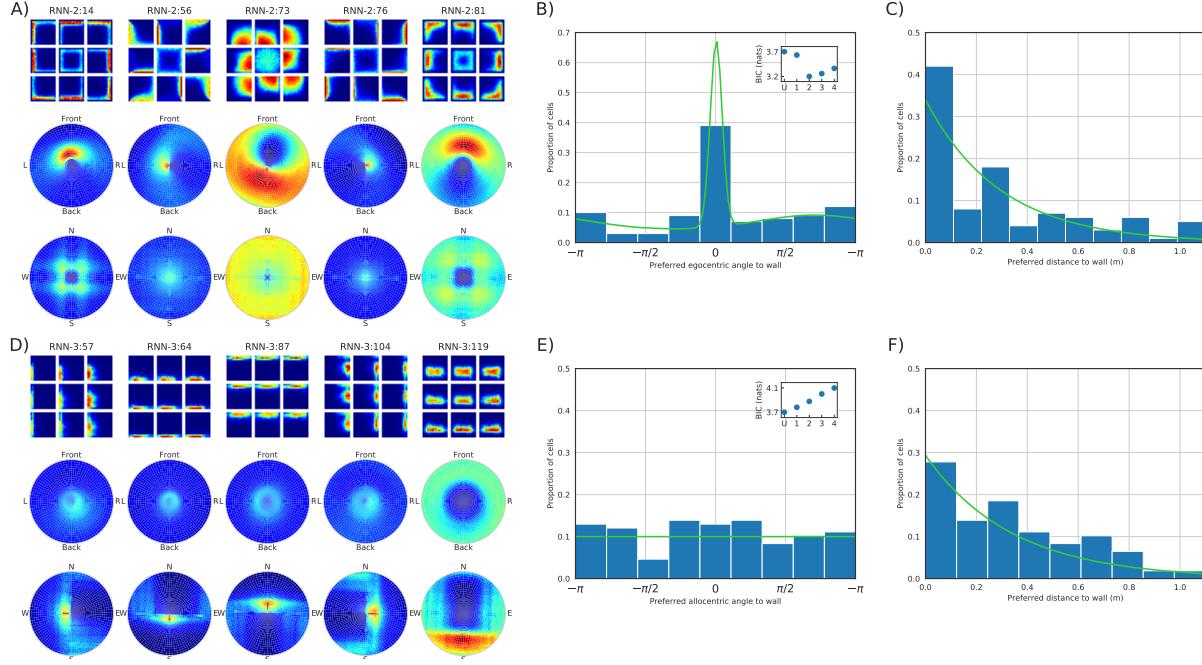
- 754 – Decoding of position and heading angle
755 – Quantitative categorisation of spatial representations
756 – Spatial ratemaps
757 – Resultant vector of binned data
758 – Head-direction cells
759 – Egocentric boundary score
760 – Allocentric boundary score
761 – Spatial stability score
762 – Place cell field characterisation
763 – Model selection criterion
764 – Reinforcement learning baselines



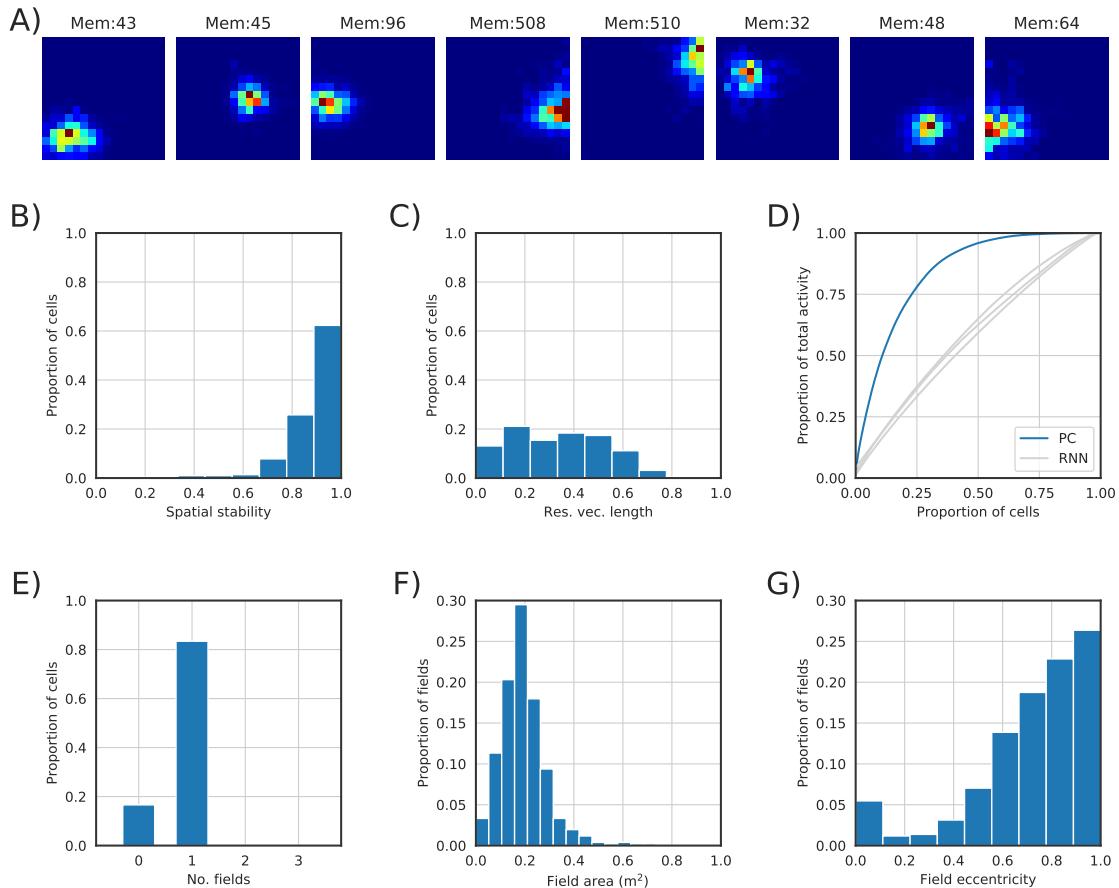
Supplementary Figure 1: Diagram of the *Spatial Memory Pipeline*. **A)** Computational diagram showing the encoder and decoder networks of the convolutional autencoder used during its training. **B)** Computational diagram showing how the loss is calculated at every time step for an example model with three predictive RNNs. P_{react} is the target distribution and P_{pred} the predicted distribution over memories. **C)** Computational diagram of the model dynamics for a time step with visual correction (10% of steps). A correction code \tilde{x} is calculated as a visually-dependent weighted average of visual memory embeddings $\mathbf{m}_{r,:}^{(x)}$. **D)** Computational diagram of the model dynamics for a time step without visual correction (90% of steps).



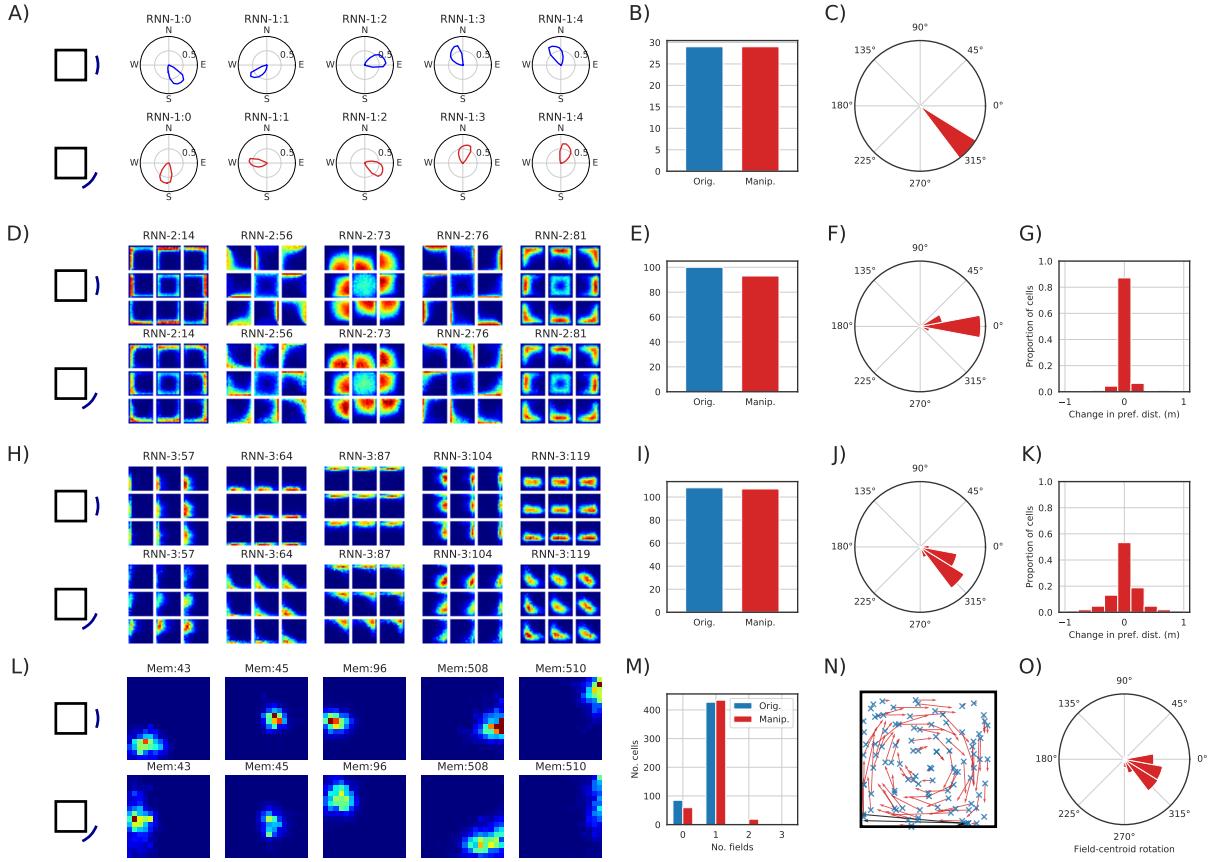
Supplementary Figure 2: Spatial representations in the *Spatial Memory Pipeline* trained in a circular open field environment with white walls. **A)** Polar plot of activity by heading direction of five cells classified as HD cells. **B)** Spatial ratemaps of five cells shown in A. Central plot shows average activation for each location across all head directions. The eight plots around each central plot show location-specific activity restricted to times when the heading direction of the agent was in the corresponding 45° range (e.g. plot located above the central plot shows average activity when the agent is facing in the north direction). **C)** Spatial ratemaps of five cells classified as egocentric-boundary cells (egoBVCs). **D)** Spatial ratemaps of five cells classified as boundary-vector cells (BVCs). **E)** Spatial ratemaps of five memory slots reactivated by RNN-3. **F)** Spatial stability of cells in each RNN (see Supplementary Methods). **G)** Resultant vector length of cells in each RNN. **H)** Resultant vector of each cell in RNN-1. **I)** Comparison of cells' responses to egocentric versus allocentric boundaries in each RNN.



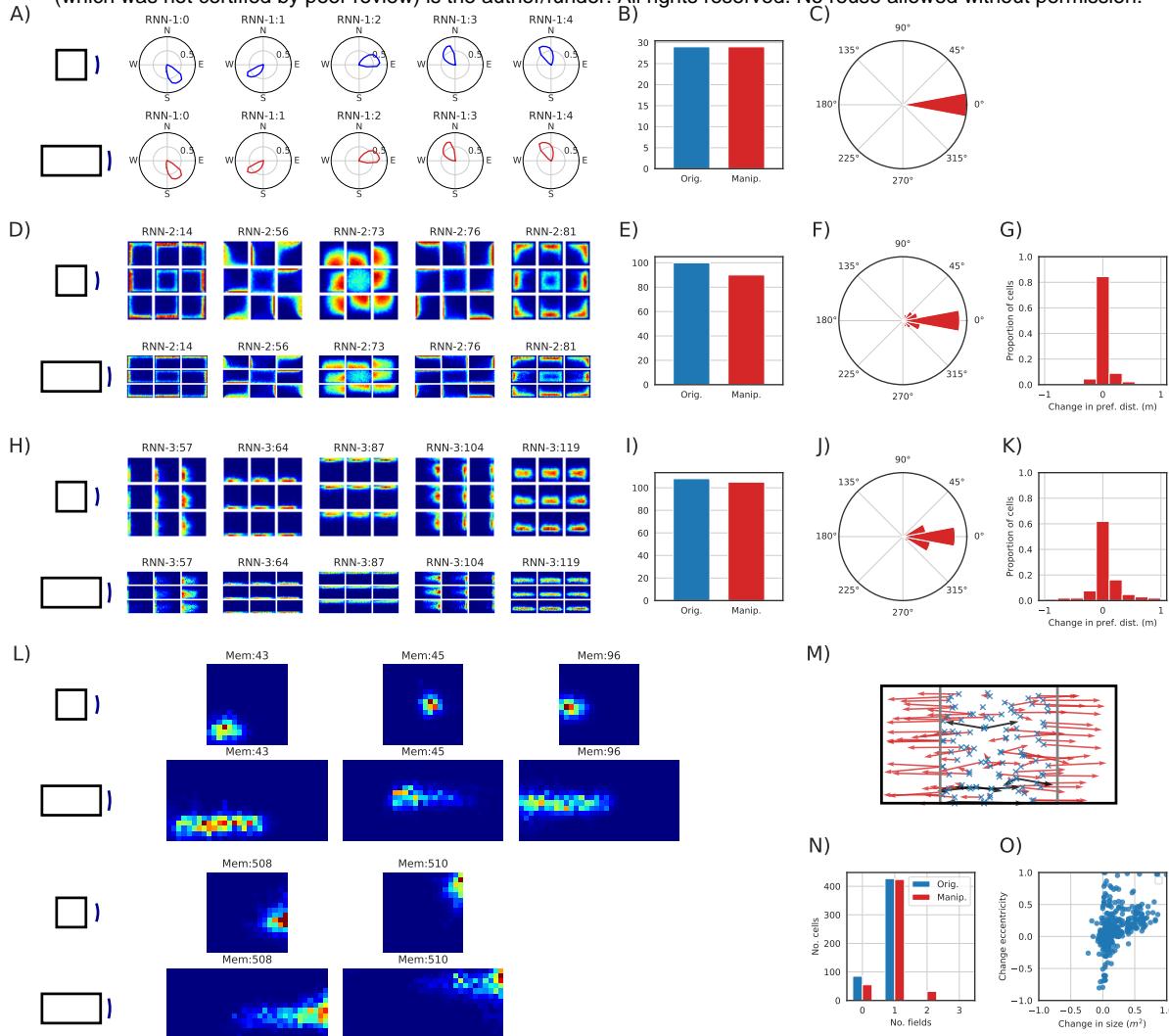
Supplementary Figure 3: Detail of boundary cells in Fig 2. **A)** First row, ratemap of five cells classified as egoBVCs. Second row, egocentric-boundary ratemap (see Supplementary Methods). Third row, allocentric-boundary ratemap (see Supplementary Methods). **B)** Histogram of preferred egocentric direction to boundary of all units classified as egoBVCs. Inset, Bayes information criterion numbers for a uniform distribution over angles (U) and mixtures of Von Mises distributions with different numbers of components. A mixture of two Von Mises distributions (shown in green), with a mode for boundaries in front agent, was selected (lowest BIC) (see Supplementary Methods). **C)** Histogram of preferred distance to boundary of all units classified as egoBVCs. The distribution over distances was well fit by an exponential distribution (shown in green) with mean distance of 35 cm. **D)** First row, ratemap of five cells classified as (allocentric) BVCs. Second row, egocentric-boundary ratemap. Third row, allocentric-boundary ratemap. **E)** Histogram of preferred allocentric direction to boundary of all units classified as BVCs. Inset shows Bayes information criterion numbers for a uniform distribution over angles (U) and mixtures of Von Mises distributions with different numbers of components. A uniform distribution (shown in green) was selected (lowest BIC). **F)** Histogram of preferred distance to boundary of all units classified as BVCs. The distribution over distances was well fit by an exponential distribution (shown in green) with mean distance of 29 cm.



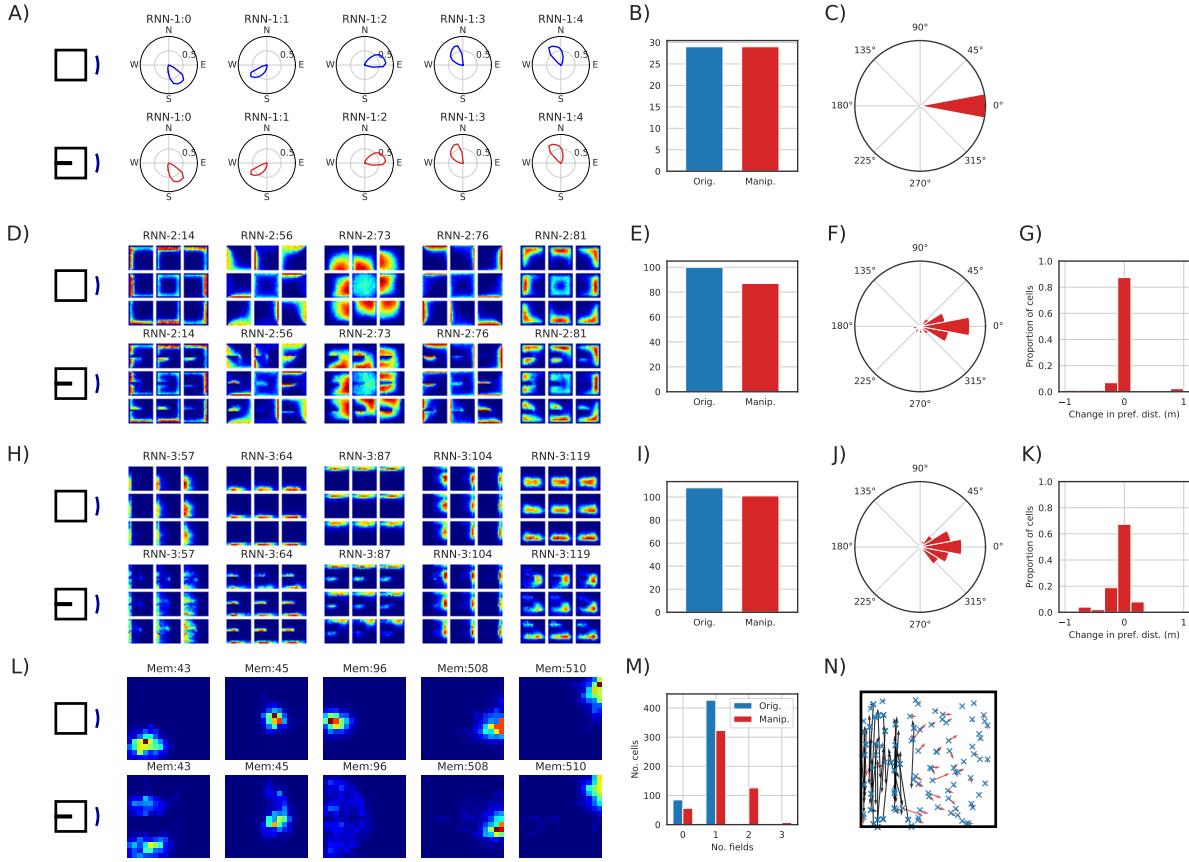
Supplementary Figure 4: Reactivation of memory slots (containing past RNN-3 states) resembles place-cell responses. **A)** Spatial ratemap of the reactivation of eight memory slots in the experiment described in Fig 2. Activity resembles that of biological place cells in the hippocampus. **B)** Spatial stability of memory slot reactivations (see Supplementary Methods). **C)** Resultant vector of memory slot reactivations. **D)** Cumulative total activity of place cells (memory slot reactivations) ordered by their total activity (blue). Place cells showed a high sparsity: 25% of cells account for 75% of activity. Activity in the three RNNs shows almost no sparsity (cells activated equally on average). **E)** Number of fields per place cell (see Supplementary Methods). **F)** Area of each activity field in square meters (total environment area is 4.8 m²). **G)** Eccentricity of fields. A score of 1.0 indicates a perfectly round field; lower scores indicate elongation.



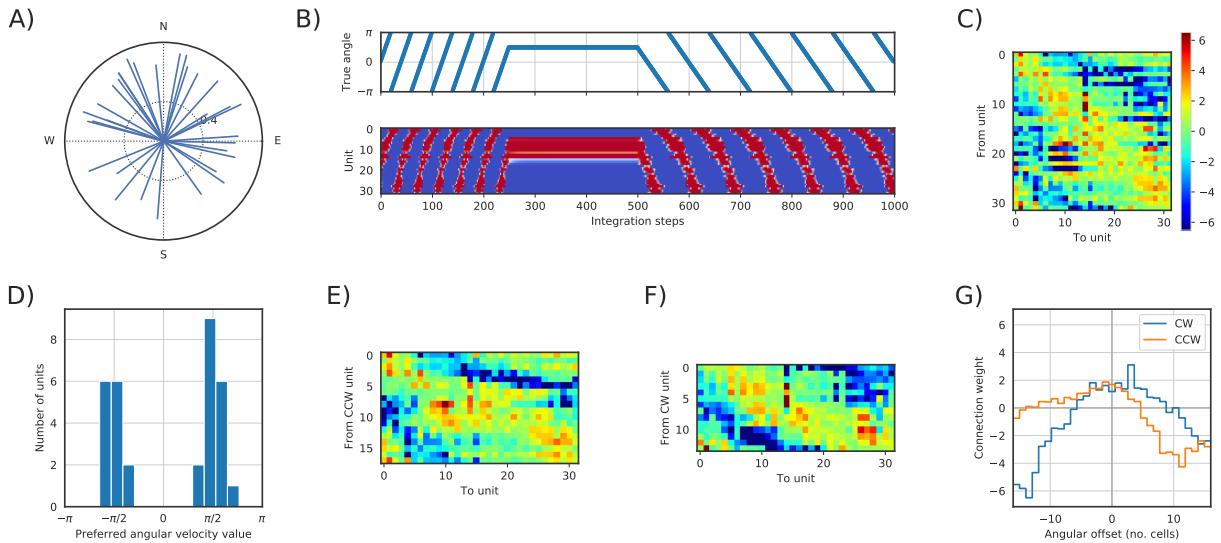
Supplementary Figure 5: Effects of distal cue rotation (45°) on the representations of the model shown in Fig 2. There was no training after the manipulation. **A)** Polar plots of five cells identified as HD cells. Top row, original environment. Second row, after manipulation. HD activity followed the rotation of distal cues. **B)** Number of RNN units identified as HD cells before and after the environment manipulation. **C)** Histogram of change in phase of the resultant vectors of all HD cells. **D)** Ratemap of five cells identified as egoBVCs. Top row, original environment. Second row, after manipulation. Rotation of distal cues did not affect egoBVCs' activity. **E)** Number of RNN units identified as egoBVCs before and after the environment manipulation. **F)** Histogram of change in preferred egocentric angle to boundary of egoBVCs. **G)** Histogram of change in preferred distance to boundary of egoBVCs. **H)** Ratemap of five cells identified as (allocentric) BVCs. Top row, original environment. Second row, after manipulation. BVCs followed the rotation of distal cues. **I)** Number of RNN units identified as BVCs before and after the environment manipulation. **J)** Histogram of change in preferred allocentric angle to boundary of BVCs. **K)** Histogram of change in preferred distance to boundary of BVCs. **L)** Ratemap of five place-cell-like units. Top row, original environment. Second row, after manipulation. **M)** Number of fields for each place cell before and after the environment manipulation. **N)** Displacement of place-cell field centroid after manipulation. Blue cross indicates the original centroid, red arrows show their displacement. Black arrows show the displacement when a single field in the original environment split into several fields after manipulation. **O)** Histogram of change in phase of the place-cell centroid of activity when calculated in polar coordinates taking as origin the centre of the enclosure.



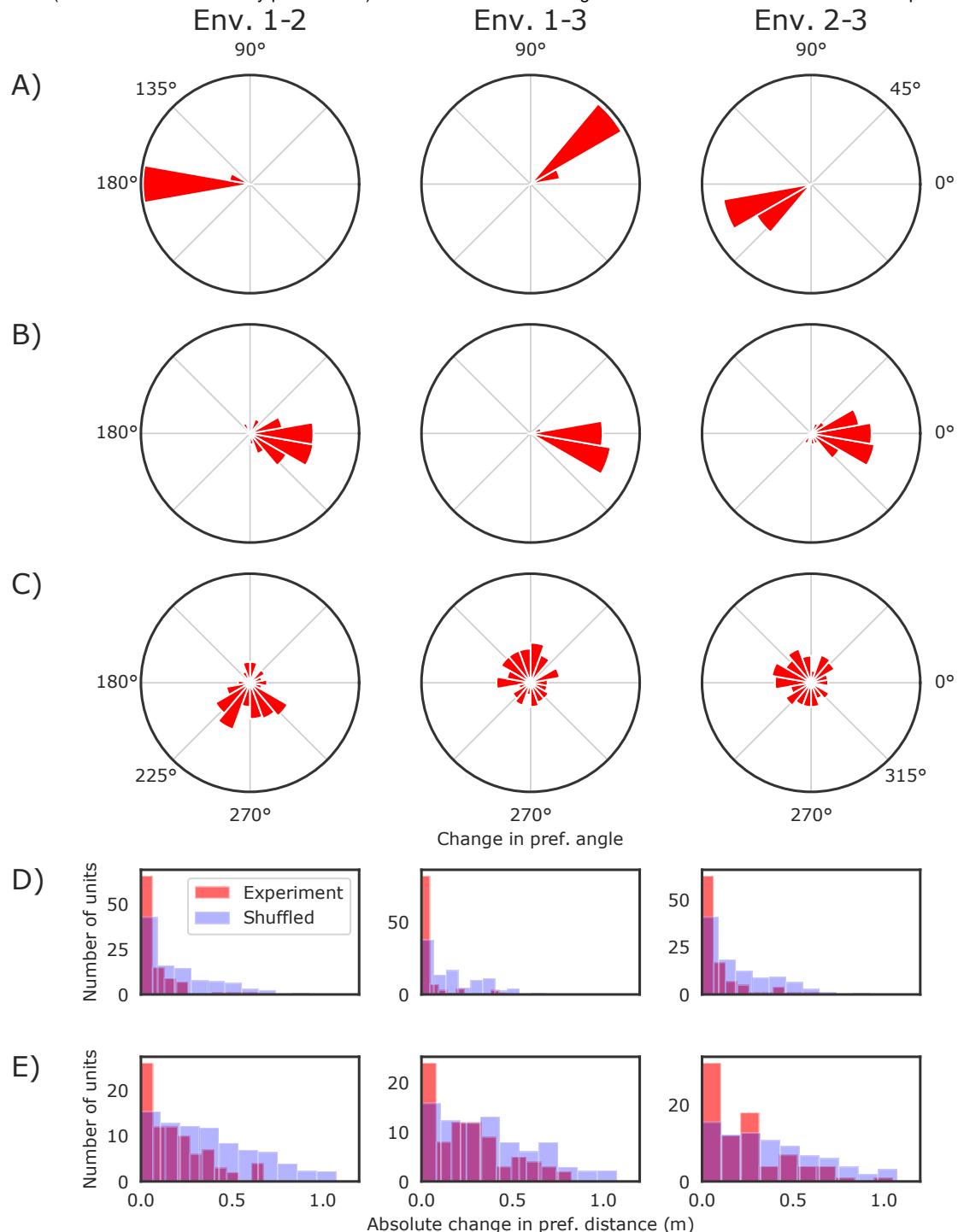
Supplementary Figure 6: Effects of enclosure resizing (width was doubled) on the representations of the model shown in Fig 2. There was no training after the manipulation. **A)** Polar plots of five cells identified as head-direction cells. Top row, original environment. Second row, after manipulation. The manipulation did not affect HD-cells. **B)** Number of RNN units identified as head-direction cells before and after the environment manipulation. **C)** Histogram of change in phase of the resultant vectors of all HD cells. **D)** Ratemap of five cells identified as egoBVCs. Top row, original environment. Second row, after manipulation. Cells maintained their egocentric response after the stretch. **E)** Number of RNN units identified as egoBVCs before and after the environment manipulation. **F)** Histogram of change in preferred egocentric angle to boundary of egoBVCs. **G)** Histogram of change in preferred distance to boundary of egoBVCs. **H)** Ratemap of five cells identified as (allocentric) BVCs. Top row, original environment. Second row, after manipulation. Cells maintained their allocentric response after the stretch. **I)** Number of RNN units identified as BVCs before and after the environment manipulation. **J)** Histogram of change in preferred allocentric angle to boundary of BVCs. **K)** Histogram of change in preferred distance to boundary of BVCs. **L)** Ratemap of five place-cell-like units. Top, original environment; below, after manipulation. The manipulation caused the activity fields of most cells to stretch with the stretched environment axis. However, some cells' activity field split into two fields. **M)** Displacement of place-cell field centroid after manipulation. Blue cross indicates the original centroid, red arrows show their displacement, and black arrows show the displacement when a single field in the original environment split into several fields after manipulation. **N)** Number of fields for each place cell before and after the environment manipulation. **O)** Change in size and eccentricity of each place cell's activity field.



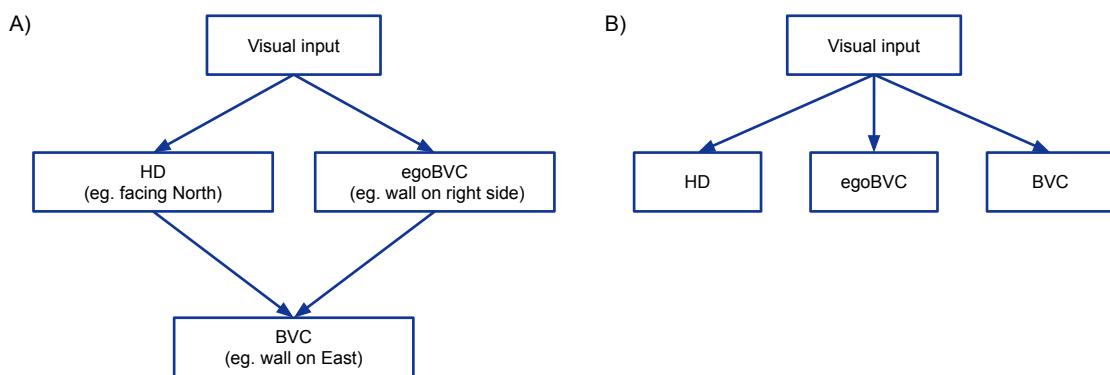
Supplementary Figure 7: Effects of extra barrier insertion on the representations of the model shown in Fig 2. There was no training after the manipulation. **A)** Polar plots of five cells identified as HD cells. Top row, original environment. Second row, after manipulation. The manipulation did not affect HD cells. **B)** Number of RNN units identified as HD cells before and after the environment manipulation. **C)** Histogram of change in phase of the resultant vectors of all HD cells. **D)** Ratemap of five cells identified as egoBVCs. Top row, original environment. Second row, after manipulation. Cells maintained their egocentric response to the new barrier. **E)** Number of RNN units identified as egoBVCs before and after the environment manipulation. **F)** Histogram of change in preferred egocentric angle to boundary of egoBVCs. **G)** Histogram of change in preferred distance to boundary of egoBVCs. **H)** Ratemap of five cells identified as (allocentric) BVCs. Top row, original environment. Second row, after manipulation. Cells maintained their allocentric response to the new barrier. **I)** Number of RNN units identified as BVCs before and after the environment manipulation. **J)** Histogram of change in preferred allocentric angle to boundary of BVCs. **K)** Histogram of change in preferred distance to boundary of BVCs. **L)** Ratemap of five place-cell-like units. Top row, original environment. Second row, after manipulation. The extra barrier caused cells in the region to duplicate their fields. Cells with distant activity fields were not affected. **M)** Number of fields for each place cell before and after the environment manipulation. **N)** Displacement of place-cell field centroid after manipulation. Blue cross indicates the original centroid, red arrows show their displacement. Black arrows show the displacement when a single field in the original environment split into several fields after manipulation.



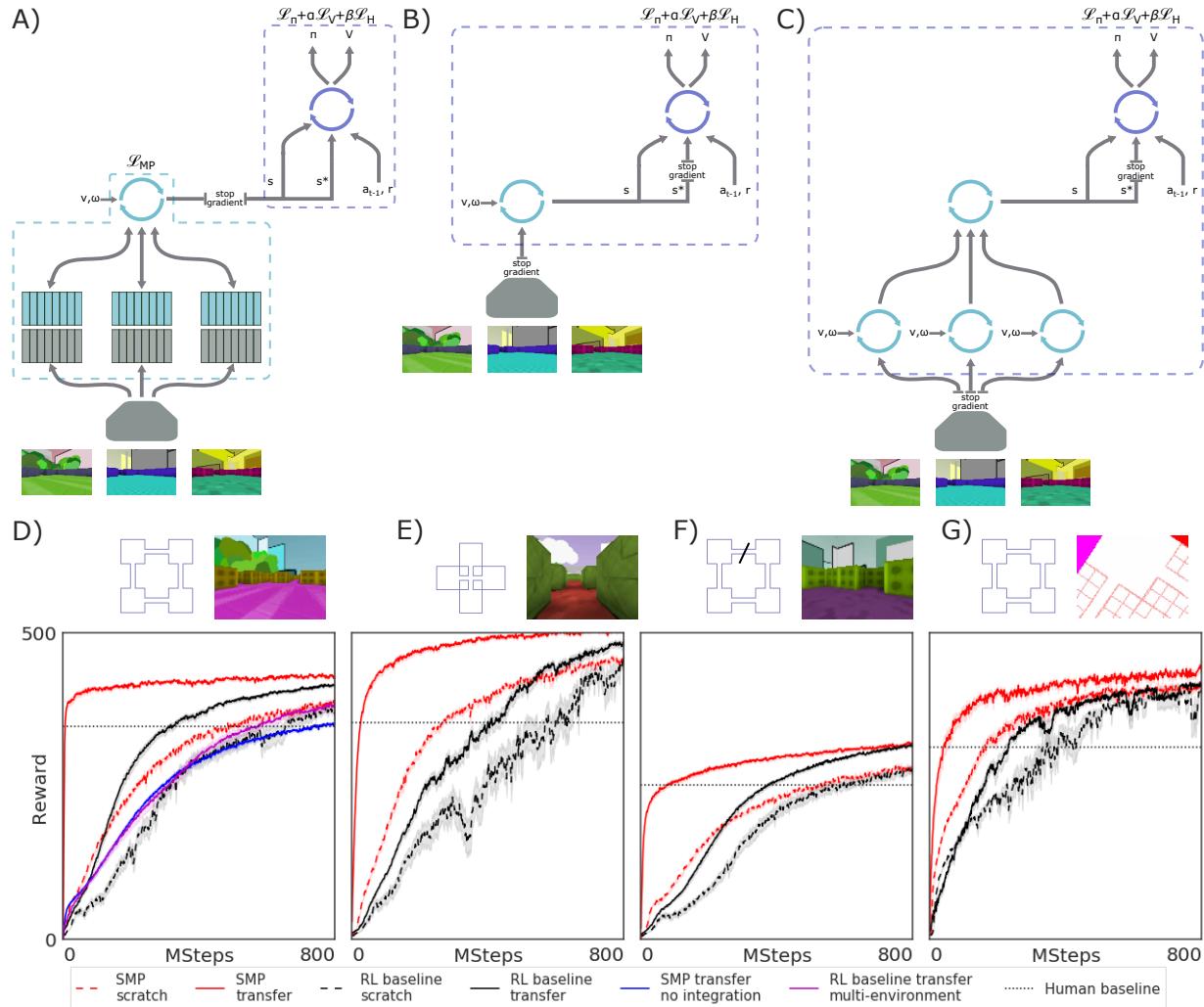
Supplementary Figure 8: Head direction cell connectivity learnt by a Vanilla-RNN network. **A)** Polar plot of the resultant vectors of cells in RNN-1. **B)** Activation of each unit over 1000 steps of blind integration. Top: True integrated angle. Bottom: Activation of HD-units in RNN-1 through time (units ordered by the phase of their resultant vectors). For the first 250 steps the angular velocity was set to $\pi/20$ rad/step, the following 250 steps 0 rad/step, the remaining 500 steps $-\pi/40$ rad/step. **C)** Weights in the RNN dynamics matrix, \mathbf{W} . Columns and rows ordered by the phase of their resultant vector. **D)** Histogram of preferred angular velocities for each cell. **E)** Matrix of weights from CCW cells to all other cells. **F)** Matrix of weights from CW cells to all other cells. **G)** Average weights connecting cells in each of the two rings to all other cells ordered by their angular offset.



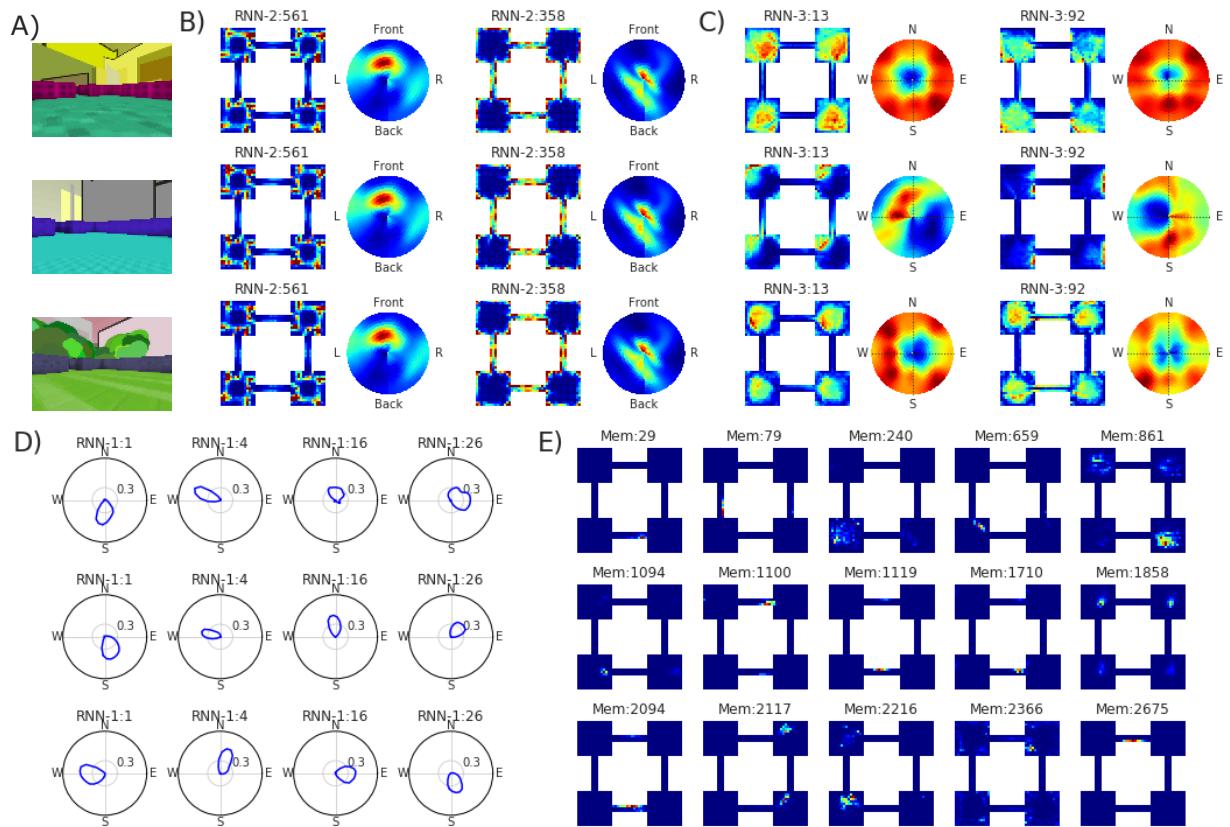
Supplementary Figure 9: Stability of representations learnt across three environments: square (1), circular (2) and trapezoidal (3). First column compares environments 1 and 2, second column environments 1 and 3, third column environments 2 and 3. **A)** Histogram of HD cell preferred direction differences between environment pairs. HD cells rotate coherently between environments. **B)** Histogram of egoBVC preferred egocentric direction differences between environments. The egoBVCs preserve their angular tuning between environments. **C)** Histogram of BVC preferred allocentric direction differences between environments. The BVCs do not rotate coherently between environments. **D)** Histogram of egoBVC preferred distance differences between environments (red) and normalised histogram of differences between randomly shuffled units (blue). The egoBVCs preserve their distance tuning across environments. **E)** Same as D) for BVCs. The distance tuning is significantly preserved across environments, although not as tightly as in the case of egoBVCs.



Supplementary Figure 10: Hypothesised BVC mechanisms. **A)** Traditional BVC mechanism: BVCs result from the conjunction of HD and egoBVC signals. **B)** Proposed BVC mechanism: BVCs are driven by the reactivation of visual memories and temporal coherence.



Supplementary Figure 11: Reinforcement learning architectures and baselines. **A)** Agent training with the *Spatial Memory Pipeline*. The visual inputs from three visually different training environments were stored in separate memory slot banks. The *Spatial Memory Pipeline* (light blue frame) was trained as in the unsupervised experiments to minimise the loss in prediction of memory reactivations. The policy (purple frame) was trained to minimise a combination of policy gradient, value and entropy losses. **B)** Agent training baseline with an LSTM replacing the *Spatial Memory Pipeline*. **C)** Agent training baseline with one LSTM per training environment and a common LSTM replacing the *Spatial Memory Pipeline*. **D-G)** Learning curves in tasks shown in Fig 4C-F. Error regions represent the standard error of the mean episode returns across 5 (D-F) or 1 (G) transfer environment and 25 trained agents (transfer) or 5 seeds (scratch). Red, *Spatial Memory Pipeline*; black, RL baseline. Dashed, learning from scratch; solid, transferred from trained agent. In D), additionally, transfer curves for the training baseline with one LSTM per training environment (magenta), and for a *Spatial Memory Pipeline* with correction at every time step (blue).



Supplementary Figure 12: Representations across three reinforcement learning enclosures, one enclosure in each row. **A)** Agent view of the enclosures. **B)** Ratemaps of two egoBVCs in RNN-2, with their egocentric-boundary ratemap (see Supplementary Methods). **C)** Ratemaps of two BVCs in RNN-3, with their allocentric-boundary ratemap (see Supplementary Methods). **D)** Polar plots of four HD cells in RNN-1. **E)** Ratemaps of five place cells in each environment. Note that, unlike the egoBVC, BVC or HD units, there is no relationship between the place cell units across environments, since they correspond to separate memory banks.

Configuration	Value	Meaning
b_m	0.13	Linear velocity Rayleigh scale while moving (m/s)
b_s	0.0	Linear velocity Rayleigh scale while stopped (m/s)
$\mu^{(\phi)}$	0	Rotation velocity Gaussian distribution mean (deg/s)
$\sigma^{(\phi)}$	330	Rotation velocity Gaussian distribution standard deviation (deg/s)
d	0.03	Distance to wall to activate wall avoidance (m)
a	90	Angle limit to activate wall avoidance (deg)
slowdown	0.25	Velocity reduction factor when avoidance activated
dt	0.02	Simulation step time increment (s)
$\mu_{(stop)}$	50	Mean period for stop/move switching (simulation steps)
n	7	Number of simulation steps per trajectory sample
N	500	Number of samples per trajectory

Supplementary Table 1: Hyperparameters of rat-like motion model³² to generate trajectories for unsupervised experiments.

Hyperparameter	Single-environment	Multi-environment
BPTT unroll length	50	50
Batch size	32	32
Training batches	200,000	200,000
Optimiser	Adam	RMSProp
Learning rate (RNNs)	$3 \cdot 10^{-3}$	$3 \cdot 10^{-4}$
Learning rate (memory embeddings)	$1 \cdot 10^{-2}$	$3 \cdot 10^{-2}$
Other optimiser params	$\beta_1 = 0.9, \beta_2 = 0.999$	decay=0.9, momentum=0
Dropout in RNN states predicting memories	0.5	0.5

Supplementary Table 2: Hyperparameters for training in unsupervised experiments.

Hyperparameter	Value
BPTT unroll length	100
Batch size	64
Optimiser (all losses)	Adam
Learning rate (actor-critic loss)	$1 \cdot 10^{-4}$ or $5 \cdot 10^{-5}$
Baseline loss coefficient	0.5
Entropy loss coefficient	$1 \cdot 10^{-3}$
Learning rate (RNNs)	$1 \cdot 10^{-3}$
Learning rate (memory embeddings)	$3 \cdot 10^{-3}$
Other optimiser params (all losses)	$\beta_1 = 0.9, \beta_2 = 0.999$
Dropout in RNN states predicting memories	0.5

Supplementary Table 3: Hyperparameters for training in reinforcement learning experiments. The learning rate for the actor-critic loss was $1 \cdot 10^{-4}$ for all experiments except the top-down schematic representation with RL-baseline, where a slower learning rate of $5 \cdot 10^{-5}$ was needed to curb instability.

Configuration	Value (unsupervised)	Value (reinforcement learning)
Image shape	64 x 64 x 3	96 x 72 x 3
Number of conv. layers	4	4
Conv. kernel sizes	5, 5, 3, 3	5, 5, 3, 3
Conv. strides	2, 2, 1, 1	2, 2, 1, 1
Conv. output channels	16, 16, 32, 32	32, 32, 64, 64
Conv. padding	Same	Same
Output vector ($y_t^{(enc)}$) size	64	128

Supplementary Table 4: Configuration of the visual encoding level of the *Spatial Memory Pipeline* in unsupervised, and reinforcement learning experiments.

Configuration	Value (unsupervised)	Value (reinforcement learning)
S	512	1024
R	3	3
\mathbf{y}_t	$\mathbf{y}_t^{(enc)}$	$\mathbf{y}_t^{(enc)}$
F_1, F_2, F_3	Sigmoid-LSTM	Sigmoid-LSTM
G_1, G_2, G_3	Sigmoid-LSTM	Sigmoid-LSTM
N_1	32	32
N_2	128	768
N_3	128	128
$\mathbf{v}_{1,t}$	$10 \cdot (\cos(\omega_t), \sin(\omega_t))$	$10 \cdot (\cos(\omega_t), \sin(\omega_t))$
$\mathbf{v}_{2,t}$	$10 \cdot (\cos(\omega_t), \sin(\omega_t), s_t)$	$10 \cdot (\cos(\omega_t), \sin(\omega_t), s_t, s_{\perp,t})$
$\mathbf{v}_{3,t}$	()	()
H_{react}	0.5 nats	1.0 nats
$P_{correction}$	0.1	0.1
$P_{storage}$	0.0000625	0.0001

Supplementary Table 5: Configuration of the first integration level of the *Spatial Memory Pipeline* in unsupervised and reinforcement learning experiments. Where $\mathbf{y}_t^{(enc)}$ is the output of the visual encoder, ω_t is the angular velocity, s_t speed parallel to the direction of heading, and (only for reinforcement learning experiments) $s_{\perp,t}$ the speed perpendicular to the direction of heading.

765 **Supplementary Results for A model of egocentric to allocentric understanding in mammalian
766 brains**

767 **Emergent representations in a circular environment** The appearance of spatial representations
768 shown in Fig 2 does not depend on the square geometry of the environment. The emergence
769 of such representations is robust to training in enclosures with different environment shapes. To
770 demonstrate this, the model was trained from scratch in a simulated circular environment of radius
771 1.5 meters, white walls, no internal cues, and distal cues consisting of a city-like scene. The
772 model developed similar representations (Supplementary Fig 2) to those appearing in the square
773 enclosure.

774 The majority of units in the first RNN module (RNN-1), which received only angular velocities
775 and corrections from the visual memories, exhibited activity modulated by the agent's direction
776 of facing (Supplementary Fig 2A-B). In total 94% (30/32) of units were classified as head direc-
777 tion cells (resultant vector length >0.48) with unimodal responses distributed uniformly in the
778 unit circle (Supplementary Fig 2H, uniform distribution was selected under BIC over mixtures
779 of Von Mises).

780 The second module (RNN-2) received angular velocity and speed, together with corrections from
781 the visual memories. Units in this RNN encoded distances relative to boundaries at a particular
782 heading (Supplementary Fig 2C). As in the square arena, they resembled egoBVC responses,
783 with 94% (120/128) of units identified as egoBVCs (ego-boundary score >0.1 , 99th percentile
784 of bin-shuffled distribution, Supplementary Fig 2I). None of the 128 units were identified as
785 BVCs. Here cells did not exhibit HD cell-like patterns of activity (mean resultant vector 0.01,
786 0/128 units classified as HD cells).

787 The third (RNN-3) received no self-motion inputs, thus was dependent upon corrections from
788 visual reactivations as its sole input and learning signal. Units in this RNN were characterised by
789 spatially stable responses (Supplementary Fig 2F). As in the square enclosure, the activity profile
790 of this module was reminiscent of BVCs (Supplementary Fig 2D). 83% (106/128) of units were
791 classified as BVCs (BVC score >0.11 , 99th percentile of shuffled distribution) – 16% (20/128)
792 met the criteria for egoBVCs, but 13 of those 20 had a higher BVC score than egoBVC score.
793 Units in this layer were not modulated by heading direction (mean resultant vector 0.11, 0/128
794 units classified as HD cells).

795 **Effects of distal cue rotation on the model's representations** First, we rotated the visible
796 distal cues by 45° clockwise (Supplementary Fig 5). The preferred orientation of HD units
797 in RNN-1 rotated en masse, tracking the cue rotation the same way that rodent head direction
798 cells are controlled by visual cues (resvec phase rotated by 43° on average, 1° SD) – tuning
799 width was unchanged (average 76.5° , 17.8° SD). The manipulation did not significantly affect
800 egoBVC activity (preferred egocentric directions to wall rotated by 3° on average, 6° SD). The
801 firing correlates of BVCs rotated with the head direction system (35° , 9° SD). The activity field
802 of the place-cell-like memory slots rotated around the centre of the environment (27° on average,
803 20° SD).

804 **Effects of enclosure stretch on the model's representations** We transformed the training en-
805 vironment stretching it by 100% along the horizontal axis to form a 4.4 m by 2.2 m rectangle
806 (Supplementary Fig 6). Again, the response characteristics of HD cells as well as egoBVCs

807 and BVCs were unchanged, and simply extended along the elongated walls like their biological
808 counterparts^{5,6,16,17}. Place cell responses closely mirrored those observed in rodents³⁸, tending to
809 be stretched along the axis of elongation and in some cases becoming bimodal (Supplementary
810 Fig 6L-O).

811 **Effects of barrier insertion on the model's representations** In a similar fashion, we introduced
812 a barrier into the virtual environment (Supplementary Fig 7). The responses of HD cells were
813 largely unaffected, maintaining the same direction of tuning (0° average change in phase, 0.2°
814 SD). Similarly, egoBVCs and BVCs retained their basic firing correlates, responding to the new
815 barrier as they had to the existing walls, resulting in the inception of additional firing fields
816 (Supplementary Fig 7D,H). Biological BVCs and egoBVCs are known to respond similarly;
817 indeed, the predictable response of these cells to extra walls is considered to be a diagnostic
818 feature^{5,6}. Place cell responses were more complex, some – typically those further from the
819 barrier – were unaffected (60%), whereas 22% formed a duplicate field on either side of the
820 barrier (Supplementary Fig 7L-N). Notably, similar outcomes have been reported in empirical
821 studies^{60,37}.

822 **Preservation of spatial characteristics across environments** To study the properties of the
823 *Spatial Memory Pipeline* representations across environments we trained concurrently in three
824 different enclosures (Fig 4A). In RNN-1, although the preferred firing directions of HD cells
825 changed between environments, the responses of all units were rotated by a similar amount,
826 maintaining the relative angular tuning between cell pairs (resultant vector phase rotation be-
827 tween environments 1 and 2 was 179° on average, 0.5° SD; between environments 1 and 3, -41°
828 on average, 0.6° SD; Fig 4A, Supplementary Fig 9A). The preservation of HD-cell characteris-
829 tics across environments is consistent with the presence of ring-attractor dynamics, as analysed
830 in single-environment experiments (Fig 3).

831 The ensemble response properties of egoBVCs in RNN-2 were similarly preserved. Between
832 the three different environments individual egoBVCs maintained their distance and directional
833 tuning, without rotation (mean resultant vector phase rotation 2° on average, 25° SD, between
834 environments 1 and 2; 0° on average, 6° SD between 1 and 3, Supplementary Fig 9B; mean
835 distance tuning difference 0.07 m between 1 and 2, 0.04 m between 1 and 3, $p < 10^{-4}$ compared
836 to differences with shuffled units, Supplementary Fig 9D). These representations do not depend
837 on arbitrary distal cues, and are therefore preserved.

838 The BVCs in RNN-3 tended also to preserve distance tuning (0.19 m mean difference in distance
839 tuning between environments 1 and 2, 0.26 m between 1 and 3, SD 0.22, $p < 10^{-4}$ compared
840 to differences with shuffled units, Supplementary Fig 9E), but not directional tuning across envi-
841 ronments (phase rotation had SD 69° between environments 1 and 2, and 89° between 1 and 3;
842 Supplementary Fig 9C).

843 **Robustness to random initialization** The representations exemplified in Fig 2A-E appear in-
844 dependently of the initial values of the parameters of the *Spatial Memory Pipeline*. In all of 20
845 experiments run with identical hyperparameters but different random parameter initializations,
846 HDs, egoBVCs and BVCs appeared in numbers similar to those reported in the main text. BVCs
847 always appear exclusively in RNN-3. HDs appear typically in RNN-1 as described (>31 out of
848 32 units classified as HD in 12 out of 20 experiments), but sometimes they are also found in
849 RNN-2. Specifically, in 7/20 experiments HDs appeared exclusively in RNN-1, as in the main

850 text example; in 5/20 they appeared both in RNN-1 and RNN-2; and in the remaining 8/20 they
851 appeared exclusively in RNN-2. In such cases, all RNN-1 units were classified as egoBVCs.
852 Since angular velocity information is available to RNN-2, it is not unexpected that HDs can be
853 learned there too, and, to the effects of prediction optimization, separating RNN-1 and RNN-2 is
854 unnecessary. Doing so, however, allows us to simplify the analysis of the attractor dynamics of
855 the HD system.

856 Similarly, the experiment described in Fig 4A and Supplementary Fig 9 was typical, but in some
857 of the experiments HDs appeared also in RNN-2. Specifically, across 15 multi-environment
858 experiments (using 5 different random seeds and 3 different RNN learning rates, 10^{-3} , $3 \cdot 10^{-4}$,
859 and 10^{-4}), 8 had HD cells purely in RNN-1, as in the experiment shown, while the other 7 had a
860 significant number of them in RNN-2 instead.

861 **Single-RNN ablation** Splitting the model into three separate RNNs with different velocity in-
862 puts is important to obtain all of the representations. If a single RNN with both linear and angular
863 velocity input is used, with the same number of units as the total in our 3-RNN model (288), ap-
864 proximately 1/3 of the units will be HDs and 1/3 will be egoBVCs; the remaining 1/3 do not
865 satisfy any of the classification criteria. Remarkably, no BVCs appear. Conversely, if the sin-
866 gle RNN receives no velocity inputs, approximately 60% of the units are classified as BVCs,
867 and nearly 30% of units classify as egoBVCs, but no HDs appear. Thus, having both an RNN
868 with velocity inputs and one without is necessary to produce HDs, egoBVCs and alloBVCs in
869 the model. As noted earlier, the split between RNN-1 and RNN-2 is not strictly necessary but
870 simplifies the analysis of the HD system.

871 **RNN-1 ring attractor analysis** To understand the mechanism by which our model tracks head
872 direction, we repeated the experiment shown in Fig 3 substituting the efficient but complex
873 LSTM integrators with Vanilla-RNNs. In Vanilla-RNNs each activity state is a simple function
874 of the previous state and inputs, thus is amenable to a mechanistic analysis⁴² (see Supplementary
875 Methods). As expected, this new experiment also developed head-direction cells in RNN-1, all
876 32 units being classified as head direction cells (resultant vector length >0.4 , 99-th percentile of
877 shuffled data, Supplementary Fig 8A). Like the LSTM-based model, the Vanilla-RNN effectively
878 integrated angular velocity over many time steps (Supplementary Fig 8B).

879 The simplicity of Vanilla-RNNs allowed us to examine the connectivity that supported angular
880 velocity integration, revealing a striking similarity with that found in the fly⁴¹ and hypothesised
881 for mammals⁴⁰. The weight matrix of dynamics, \mathbf{W} , when displayed ordered according to each
882 units' preferred firing direction (Supplementary Fig 8C), resembled a circulant matrix, with a
883 diagonal band of excitatory connections surrounded by diagonal bands of inhibition. That is,
884 the connectivity between units forms a ring with local excitation and long-range inhibition. The
885 angular-velocity tuning of the cells, calculated as the arc-tangent of the 2-dimensional vector
886 of control weights, \mathbf{V} , that receive as input the cosine and sine of the angular velocity, showed
887 a clear split of the units in two groups, 18/32 units being activated by positive angular veloci-
888 ties (ccw-cells) and 14/32 activated by negative angular velocities (cw-cells) (Supplementary
889 Fig 8D). Plotting the weight matrix of dynamics separately for each group revealed the mech-
890 anism by which angular velocity was integrated (Supplementary Fig 8E-F). Each cell prefer-
891 entially excited other cells offset around the ring in the same direction as their angular-velocity
892 tuning, an asymmetry that became more obvious when the average weights to units with different

893 relative firing directions were calculated (Supplementary Fig 8G).

894 **Supplementary Methods for A model of egocentric to allocentric understanding in mam-**
895 **malian brains**

896 **Decoding of position and heading angle** For decoding of position and head direction (Fig 1D)
897 we trained a single-layer MLP (256 hidden units with tanh non-linearity) to predict either the true
898 (x, y) position of the agent in the environment, or the cosine and sine of its true head direction,
899 from either the vector of log probabilities of activation of the visual slots or the concatenation
900 of all the RNN outputs. We used RMSProp as optimiser (learning rate 10^{-4} , decay 0.9, no mo-
901 mentum) with square loss. The decoder was trained on 10 million batches of 25 frames sampled
902 at random out of 300,000 frames from the same trajectories used for unsupervised training of
903 the Spatial Memory Pipeline. The decoding error shown in Fig 1D for position is the mean Eu-
904 clidean distance over the whole training set between the (x, y) output of the decoder and the true
905 position. For head direction, it is the mean absolute difference between the true head direction
906 and the arc-tangent of the cosine and sine prediction of the decoder. Error bars show the standard
907 error of the mean calculated as the standard deviation of 100 bootstrapped means computed from
908 5000 decoded samples.

909 **Quantitative categorisation of spatial representations** Where possible we have used the same
910 functional characterisations of cells used for rodent data. Resultant vector lengths of allocentric
911 direction-of-facing were used to characterise head-direction cells⁶¹. Resultant vector lengths of
912 egocentric-boundary ratemaps were used to characterise egocentric boundary cells (egoBVCs)⁶².
913 We used the same procedure with allocentric-boundary ratemaps to detect allocentric boundary
914 cells (BVCs). Our results were based on datasets of 800 trajectories each consisting of 500
915 timesteps.

916 Our simulation datasets of 400,000 timesteps correspond to much longer duration than is com-
917 mon in rodent studies. For this reason we avoided using thresholds based on random shifts of
918 the data, as this results in very low thresholds due to a null hypothesis that removes most of the
919 dependency of activations from any environmental features. We used a bin-shuffling procedure
920 that results in much more stringent, and qualitatively pleasing, thresholds. Thresholds were cal-
921 culated as the 99-th percentile of resultant vector lengths for 1000 bin shuffles for each unit. All
922 units from all RNNs were used to calculate the threshold of each cell category.

923 **Spatial ratemaps** We measured and plotted the dependence of a cell's activity on the agent's
924 location by using spatial ratemaps. We partitioned the environment into a grid of 14cm by 14cms
925 square bins aligned with the cardinal directions. The spatial ratemap for a cell is the matrix of its
926 average activity when the agent is located inside each of these bins.

927 When we aimed to show the dependence of cell's activity on both spatial location and head-
928 direction, the per-octant spatial ratemaps were calculated and plotted surrounding the spatial-
929 ratemap. These are eight spatial ratemaps where the data is restricted to times when the agent's
930 head direction is contained in each of the eight intervals of 45° centered at the cardinal and
931 inter-cardinal directions.

932 **Resultant vector of binned data** Resultant vectors were calculated as a single complex number
933 from mean activities binned by angle:

$$R_u = \sum_b \frac{F_{b,u}}{\sum_{b'} F_{b',u}} e^{ib\frac{2\pi}{B}} \quad (10)$$

934 where $F_{b,u}$ is the average activity of unit u for the b -th bin out of a total of B bins.

935 The length of the resultant vector corresponds to the modulus of R_u and serves as a statistical
 936 measure of circular concentration (inverse variance). While the argument of R_u is a statistical
 937 measure of preferred direction of activation.

938 **Head-direction cells** Head direction cells characteristically fire in a small range of allocentric
 939 direction. This preference for a single direction can be measured by the length of the resultant
 940 vector of activations.

In order to calculate the resultant vector of each unit, we partitioned allocentric directions into 20 bins. The angular ratemap, $F_{b,u}$, of the u -th unit, with instantaneous activity $a_{u,t}$ at time t , was calculated as:

$$F_{b,u} = \mathbb{E}_{t|\alpha_t \in b\text{-th bin}} a_{u,t} \quad (11)$$

941 where α_t the agent's direction of facing at time t , and $\alpha_t \in b$ -th bin if $(b - 0.5)\frac{2\pi}{20} < \alpha_t \leq$
 942 $(b + 0.5)\frac{2\pi}{20}$.

943 The tuning-curve width for units classified as head-direction cells was calculated as $\frac{2\pi}{20}$, the
 944 width of a single bin, times the number of bins where the mean activity was greater than 0.5 of
 945 the greatest mean-bin activity for that unit.

946 **Egocentric boundary score** Egocentric boundary cells were characterised by the resultant vector
 947 of a the egocentric-boundary ratemap (EBR)⁶². The EBR uses the agent's position and
 948 direction of facing as frame of reference and computes the average instantaneous activity when
 949 a boundary is present at a particular distance and egocentric direction. In our unsupervised ex-
 950 periments the EBR was calculated for bins of 4° by 2.5cm. The maximum distance considered
 951 was half of the maximum distance between opposing walls. Examples of EBRs can be seen in
 952 the middle row of Supplementary Fig 3A and D.

953 The length of resultant vector of the EBR (averaged over distance) was used to detect egocentric
 954 boundary cells. For a cell classified as an egoBVC, the angle of the resultant vector was taken as
 955 its preferred direction and the distance of the maximum activity along this direction in the EBR
 956 as its preferred distance.

957 **Allocentric boundary score** We characterised allocentric boundary cells using an allocentric-
 958 boundary ratemap (ABR). In the ABR the agent's position is used as frame of reference but not
 959 its direction of facing, the relative angle to boundaries is locked to the allocentric north direction.
 960 The ABR computes the average instantaneous activity where a boundary is present at a particular
 961 distance and allocentric direction. In our unsupervised experiments the ABR was calculated for

962 bins of 4° by 2.5cm. The maximum distance considered was half of the maximum distance
963 between opposing walls. Examples of ABRs can be seen in the bottom row of Supplementary
964 Fig 3A and D

965 The length of resultant vector of the ABR (averaged over distance) was used to detect allocentric
966 boundary cells (BVC). For a cell classified as a BVC, the angle of the resultant vector was taken
967 as its preferred direction and the distance of the maximum activity along this direction in the
968 ABR as its preferred distance.

969 **Spatial stability score** Where reported, the spatial stability of a unit was measured as the Pearson
970 correlation of its spatial ratemap for two halves of the data. In our experiments each half
971 of the data was comprised of 200,000 data points, which under the motion model used would
972 amount to 3000 seconds for each half.

973 **Place cell field characterisation** The activity field of a cell was calculated using the following
974 procedure on its spatial ratemap: (1) a Gaussian filter of radius 1 bin was applied, (2) bins
975 with value higher than half the maximum value were considered part of the activity fields, (3)
976 fields were segmented using the `skimage.measure.label` function, (4) fields smaller than
977 3 adjacent bins were discarded, and (5) the attributes of each field (position, size, and eccentricity)
978 were calculated using the `skimage.measure.regionprops` function with the original
979 ratemap as `intensityimage` parameter. We used version 0.14.0 of `skimage` throughout.

980 **Model selection criterion** Where a probability distribution is reported as fitting data (e.g. the
981 distribution of HD angles, or the angular preferences of egoBVCs) we used the Bayesian information
982 criterion to compare several alternative distributions. This criterion penalises the log-
983 likelihood of the model, L , by a term that depends on the number of parameters, k , and number
984 of data points, n :

$$BIC = k \ln n - 2L \quad (12)$$

985 A lower BIC signals a better model fit.

986 **Reinforcement learning baselines** Supplementary Fig 11A depicts the architecture of our agent
987 as described in Methods. The baselines we used to compare performance in the water maze
988 transfer tasks are the following:

- 989 • An agent with the same architecture, receiving visual corrections at every time step (instead
990 of every 10 steps on average). This is the baseline labelled *SMP transfer no integration* in
991 Supplementary Fig 11D.
- 992 • An agent where the *Spatial Memory Pipeline* was replaced by a generic recurrent network
993 (LSTM) (Supplementary Fig 11B) integrating visual and velocity inputs. The size of the
994 LSTM output was 928 units, to make it the same as the concatenation of all the RNN
995 outputs in the *Spatial Memory Pipeline*. Unlike the memory pipeline, the LSTM did not
996 have a separate unsupervised loss for training, but was trained directly from the gradients
997 of the policy loss.
- 998 • An agent where the *Spatial Memory Pipeline* was replaced by a two-layer network of
999 LSTMs (Supplementary Fig 11C), where the first layer consisted of three LSTMs (output

1000 size 512 units), each integrating the visual and velocity inputs from one of the three training
1001 environments, and the second layer consisted of a single LSTM (output size 928 units)
1002 integrating the batched outputs of the first-layer LSTMs. This network paralleled the *Spatial*
1003 *Memory Pipeline* architecture, with the first-layer LSTMs playing the role of the per-
1004 environment memory banks and the second-layer LSTM the role of the memory pipeline
1005 RNNs. For transfer, the first layer was replaced by a single 512-unit LSTM, and the rest of
1006 parameters were kept from the trained agent. As with the single-layer LSTM baseline, the
1007 LSTMs did not have a separate unsupervised loss for training, and were trained directly
1008 from the gradients of the policy loss.

1009

- 1010 60. Rivard, B., Li, Y., Lenck-Santini, P.-P., Poucet, B. & Muller, R. U. Representation of objects
1011 in space by two classes of hippocampal pyramidal cells. *The Journal of general physiology*
1012 **124**, 9–25 (2004).
- 1013 61. Knight, R. *et al.* Weighted cue integration in the rodent head direction system. *Philosophical*
1014 *Transactions of the Royal Society B: Biological Sciences* **369**, 20120512 (2014).
- 1015 62. Alexander, A. *et al.* Egocentric boundary vector tuning of the retrosplenial cortex. *bioRxiv*
1016 702712 (2019).