

¹ Vector-based Navigation using Grid-like Representations ² in Artificial Agents.

³ Andrea Banino^{1,2,3*}, Caswell Barry^{2*}, Benigno Uria¹, Charles Blundell¹, Timothy Lillicrap¹,
⁴ Piotr Mirowski¹, Alexander Pritzel¹, Martin J. Chadwick¹, Thomas Degrif¹, Joseph Modayil¹,
⁵ Greg Wayne¹, Hubert Soyer¹, Fabio Viola¹, Brian Zhang¹, Ross Goroshin¹, Neil Rabinowitz¹,
⁶ Razvan Pascanu¹, Charlie Beattie¹, Stig Petersen¹, Amir Sadik¹, Stephen Gaffney¹, Helen King¹,
⁷ Koray Kavukcuoglu¹, Demis Hassabis^{1,4}, Raia Hadsell¹, Dharshan Kumaran^{1,3}

⁸ ¹DeepMind, 5 New Street Square, London EC4A 3TW, UK.

⁹ ²Department of Cell and Developmental Biology, University College London, London, UK

¹⁰ ³Centre for Computation, Mathematics and Physics in the Life Sciences and Experimental Biology
(CoMPLEX), University College London, London, UK

¹² ⁴Gatsby Computational Neuroscience Unit, 25 Howland Street, London W1T 4JG, UK

¹³ *equal contribution.

¹⁴ **Deep neural networks have achieved impressive successes in diverse areas ranging from ob-**
¹⁵ **ject recognition to complex games such as Go^{1,2}. Navigation, however, remains a substantial**
¹⁶ **challenge for artificial agents, with deep neural networks trained by reinforcement learn-**
¹⁷ **ing (RL)³⁻⁵ failing to rival the proficiency of mammalian spatial behavior, underpinned by**
¹⁸ **grid cells in the entorhinal cortex⁶. Grid cells are viewed to provide a multi-scale periodic**
¹⁹ **representation that functions as a metric for coding space^{7,8} which is critical for integrat-**
²⁰ **ing self-motion (path integration)^{6,7,9} and planning direct trajectories to goals (vector-based**
²¹ **navigation)^{7,10,11}. We set out to leverage the computational functions of grid cells to develop a**
²² **deep RL agent with mammalian-like navigational abilities. We first trained a recurrent net-**
²³ **work to perform path integration, leading to the emergence of representations resembling**
²⁴ **grid cells, as well as other entorhinal cell types¹². We then showed that this representation**

25 **provided an effective basis for an agent to locate goals in complex, unfamiliar, and change-**
26 **able environments — optimizing the primary objective of navigation through deep RL. The**
27 **performance of agents endowed with grid-like representations surpassed that of an expert**
28 **human and comparison agents, with the metric quantities necessary for vector-based nav-**
29 **igation derived from grid-like units within the network. Further, grid-like representations**
30 **enabled agents to conduct shortcut behaviours reminiscent of those performed by mammals.**
31 **Our findings show that emergent grid-like representations furnish agents with a Euclidean**
32 **spatial metric and associated vector operations, providing a foundation for proficient nav-**
33 **igation. As such, our results support neuro-scientific theories that see grid cells as critical**
34 **for vector-based navigation^{7,10,11}, demonstrating that the latter can be combined with path-**
35 **based strategies to support navigation in complex environments.**

36 The ability to self-localize in the environment and update one's position on the basis of self-
37 motion are core components of navigation¹³. We trained a deep neural network to path integrate
38 within a square arena (2.2m×2.2m), using simulated trajectories modelled on those of foraging ro-
39 dents (see Methods). The network was required to update its estimate of location and head direction
40 based on translational and angular velocity signals, mirroring those available to the mammalian
41 brain^{12,14,15} (see Methods, Fig. 1a&b). Velocity was provided as input to a recurrent network
42 with a Long Short-Term Memory architecture (LSTM) which was trained using backpropagation
43 through time (see Methods and Supplemental Discussion), allowing the network to dynamically
44 combine current input signals with activity patterns reflecting past events (see Methods, Fig. 1a).
45 The LSTM projected to place and head direction units via a linear layer — units with activity

46 defined as a simple linear function of their input (see Extended Data Figure 1 for architecture).
47 Importantly, the linear layer was subject to regularization, in particular dropout¹⁶, such that 50% of
48 the units were randomly silenced at each time step. The vector of activities in the place and head
49 direction units, corresponding to the current position, was provided as a supervised training signal
50 at each time step (see Methods and Extended Data Figure 1). This form of supervision follows
51 evidence that in mammals, place and head direction representations exist in close anatomical prox-
52 imity to entorhinal grid cells¹² and emerge in rodent pups prior to the appearance of mature grid
53 cells^{17,18}. Equally, in adult rodents, entorhinal grid cells are known to project to the hippocampus¹⁹
54 and appear to contribute to the neural activity of place cells¹⁹.

55 As expected, the network was able to path integrate accurately in this setting involving for-
56 aging behavior (mean error after 15s trajectory, 16cm vs 91cm for an untrained network, effect
57 size = 2.83; 95% CI [2.80, 2.86], Fig. 1b&c). Strikingly, individual units within the linear layer
58 of the network developed stable spatial activity profiles similar to neurons within the entorhinal
59 network^{6,12} (Fig. 1d, Extended Data Figure 2). Specifically, 129 of the 512 linear layer units
60 (25.2%) resembled grid cells, exhibiting significant hexagonal periodicity (gridness²⁰) versus a
61 null distribution generated by a conservative fields shuffling procedure (see Methods). The scale
62 of the grid-patterns, measured from the spatial autocorrelograms of the activity maps²⁰, varied
63 between units (range 28cm to 115cm, mean 66cm) and followed a multi-modal distribution, con-
64 sistent with empirical results from rodent grid cells^{21,22} (Fig. 1e). To assess these clusters we
65 fit mixtures of Gaussians, finding the most parsimonious number by minimizing the Bayesian In-
66 formation Criterion (BIC). The distribution was best fit by 3 Gaussians (means 47cm, 70cm, and

67 106cm), indicating the presence of scale clusters with a ratio between neighbouring clusters of
68 approximately 1.5, closely matching theoretical predictions²³ and lying within the range reported
69 for rodents^{21,22} (Fig. 1e, Extended Data Figure 3). The linear layer also exhibited units resembling
70 head direction cells (10.2%), border cells (8.7%), and a small number of place cells¹² as well as
71 conjunctions of these representations (Fig. 1d,f&g, Extended Data Figure 2).

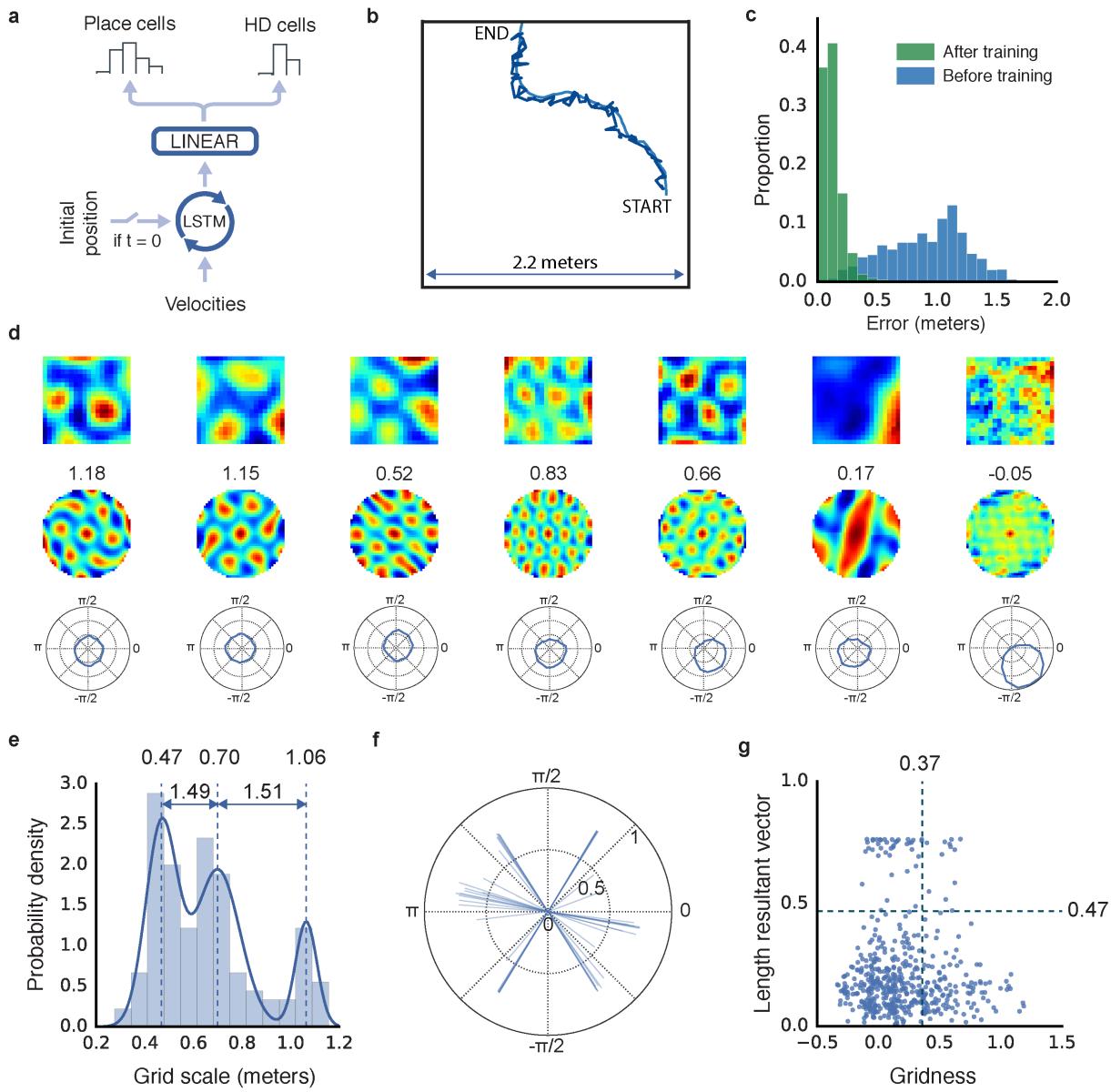


Figure 1: Entorhinal-like representations emerge in a network trained to path integrate. a, Schematic of network architecture (see Extended Data Figure 1 for details) . b, Example trajectory (15s), self-location decoded from place cells resembles actual path (respectively, dark and light-blue). c, Accuracy of decoded location before (blue) and after (green) training. d, Linear layer units exhibit spatially tuned responses resembling grid, border, and head direction cells. Ratemap shows activity over location (top), spatial autocorrelogram of the ratemap with gridness indicated (middle), polar plot show activity vs. head direction (bottom). e, Spatial scale of grid-like units ($n = 129$) is clustered. Distribution is more discrete²² than chance (effect size = 2.98, 95% CI [0.97, 4.91]) and best fit by a mixture of 3 Gaussians (centres 0.47, 0.70 & 1.06m, ratio=1.49 & 1.51). f, Directional tuning of the most strongly directional units ($n = 52$). Lines indicate length and orientation of resultant vector (see Methods), exhibiting a six-fold clustering reminiscent of conjunctive grid cells²⁴. g, Distribution of gridness and directional tuning. Dashed lines indicate 95% confidence interval from null distributions (based on 500 data permutations), 14 (11%) grids exhibit directional modulation (see Methods). Similar results were seen in a circular environment (Extended Data Figure 3).

72 To ascertain how robust these representations were, we retrained the network 100 times,
73 in each instance finding similar proportions of grid-like units (mean 23% SD 2.8%, units with
74 significant gridness scores) and other spatially modulated units (Extended Data Figure 3). Con-
75 versely, grid-like representations did not emerge in networks without regularization (e.g. dropout,
76 see Methods; also see ²⁵, Extended Data Figure 4). Therefore, the use of regularization, including
77 dropout which has been viewed to be a parallel of noise in neural systems¹⁶, was critical to the
78 emergence of entorhinal-like representations. Notably, therefore, our results show that grid-like
79 representations reminiscent of those found in the mammalian entorhinal cortex emerge in a generic
80 network trained to path integrate, contrasting with previous approaches using pre-configured grid
81 cells (e.g. ²⁶; see Supplemental Discussion). Further our results are consistent with the view that
82 grid cells represent an efficient and robust basis for a location code updated by self-motion cues^{6–9}.

83 Next, we sought to test the hypothesis that the emergent representations provide an effective
84 basis function for goal-directed navigation in complex, unfamiliar, and changeable environments,
85 when trained through deep RL. Entorhinal grid cells have been proposed to provide a Euclidean
86 spatial metric and thus support the calculation of goal-directed vectors, enabling animals to follow
87 direct routes to a remembered goal, a process known as vector-based navigation^{7,10,11}. Theoreti-
88 cally, the advantage of decomposing spatial location into a multi-scale periodic code, as provided
89 by grid cells, is that the relative position of two points can be retrieved by examining the differ-
90 ence in the code at the level of each scale — combining the modulus remainders to return the
91 true vector^{7,11} (Fig. 2a). However, despite the obvious utility of such a framework, experimen-
92 tal evidence for the direct involvement of grid representations in goal-directed navigation is still

93 lacking.

94 To develop an agent with the potential for vector-based navigation, we incorporated the “grid
95 network” described above, into a larger architecture that was trained using deep RL (Fig. 2d,
96 Extended Data Figure 5). As before, the grid network was trained using supervised learning but,
97 to better approximate the information available to navigating mammals, it now received velocity
98 signals perturbed with random noise as well as visual input. Experimental evidence suggests that
99 place cell input to grid cells corrects for drift and anchors grids to environmental cues²¹. To parallel
100 this, visual input was processed by a ”vision module” consisting of a convolutional network that
101 produced place and head direction cell activity patterns which were provided as input to the grid
102 network 5% of the time – akin to a moving animal making occasional, imperfect observations of
103 salient environmental cues²⁷ (see Methods, Fig. 2b&c and Extended Data Figure 5). The output
104 of the linear layer of the grid network, corresponding to the agent’s current location, was provided
105 as input to the “policy LSTM”, a second recurrent network controlling both the agent’s actions
106 and outputting a value function. Additionally, whenever the agent reached the goal, the ”goal grid
107 code” — activity in the linear layer — was subsequently provided to the policy LSTM during
108 navigation as an additional input.

109 We first examined the navigational capacities of the agent in a simple setting inspired by the
110 classic Morris water maze (Fig. 2b&c; 2.5m×2.5m square arena; see Methods and Supplemental
111 Results). Notably, the agent was still able to self localize accurately in this more challenging setting
112 where ground truth information about location was not provided and velocity inputs were noisy

¹¹³ (mean error after 15s trajectory, 12cm vs 88cm for an untrained network, effect size = 2.82; 95% CI
¹¹⁴ [2.79, 2.84], Fig. 2e). Further, the agent exhibited proficient goal-finding abilities, typically taking
¹¹⁵ direct routes to the goal from arbitrary starting locations (Fig. 2h). Performance exceeded that
¹¹⁶ of a control place cell agent (Fig. 2f, Supplemental Results and Methods), chosen because place
¹¹⁷ cells provide a robust representation of self-location but are not thought to provide a substrate
¹¹⁸ for long range vector calculations¹¹. We examined the units in the linear layer, again finding a
¹¹⁹ heterogeneous population resembling those found in entorhinal cortex, including grid-like units
¹²⁰ (21.4%) as well as other spatial representations (Fig. 2g, Extended Data Figure 6) — paralleling
¹²¹ the dependence of mammalian grid cells on self-motion information^{15,28} and spatial cues^{6,21}.

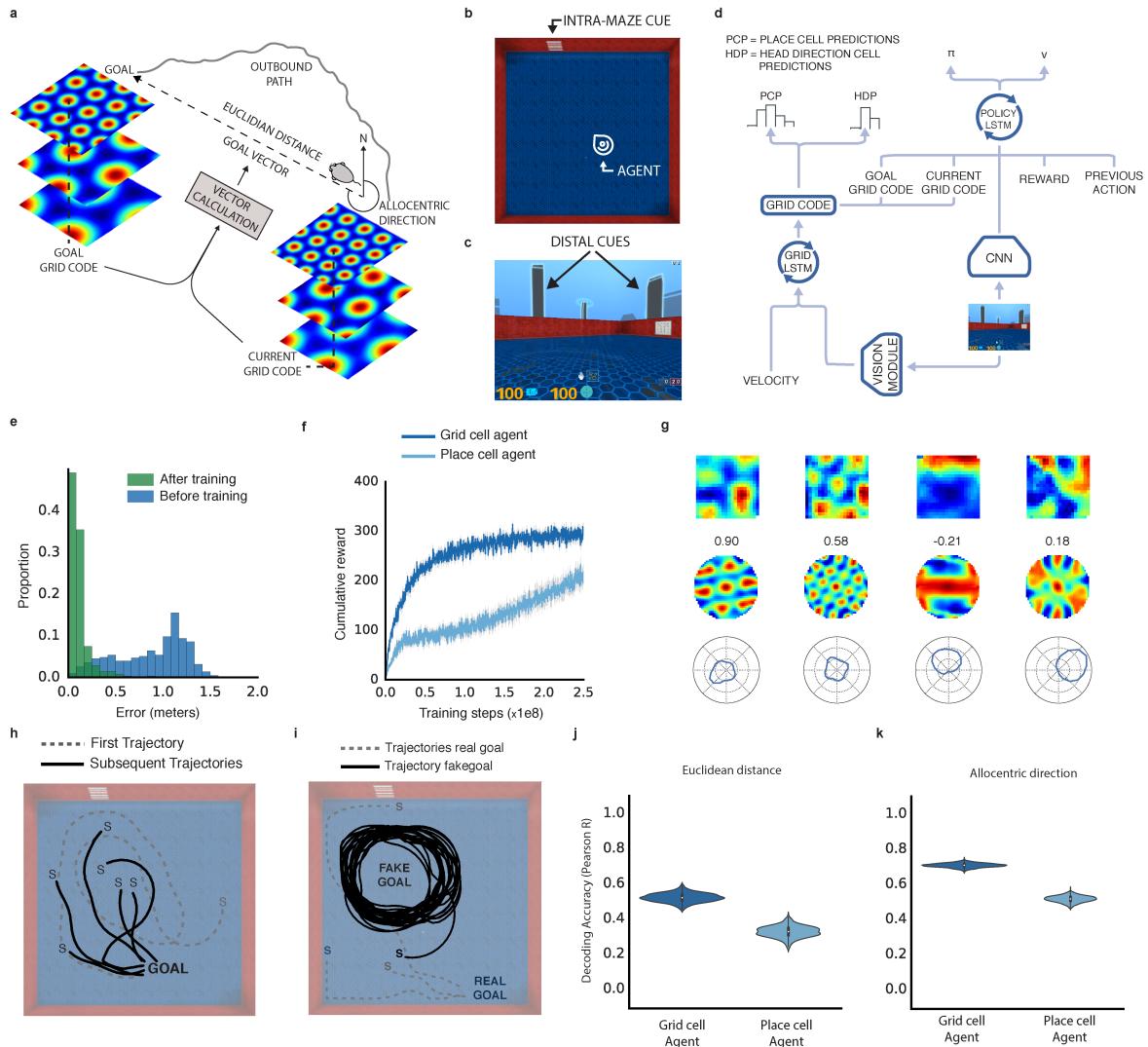


Figure 2: **One-shot open field navigation to a hidden goal.** a, Schematic of vector-based navigation¹¹. b, Overhead view of typical environment (icon indicates agent and facing direction). c, Agent view of (b). d, Schematic of Deep RL architecture (see Extended Data Figure 5) e, Accuracy of self-location decoded from place cell units. f, Performance of grid cell agent and place cell agent (y-axis shows reward obtained within a single episode, 10 points per goal arrival, gray band displays the 68% confidence interval based on 5000 bootstrapped samples). g, As before the linear layer develops spatial representations similar to entorhinal cortex. Left to right, 2 grid cells, 1 border cell, and 1 head direction cell. h, On the first trial of an episode the agent explores to find the goal and subsequently navigates directly to it. i, After successful navigation, the policy LSTM was supplied with a "fake" goal grid-code, directing the agent to this location where no goal was present. j&k, Decoding of goal-directed metric codes (i.e. Euclidean distance and direction) from the policy LSTM of grid cell and place cell agents. The bootstrapped distribution (1000 samples) of correlation coefficients are each displayed with a violin plot overlaid on a Tukey boxplot.

122 We next turn to our central claim, that grid cells endow agents with the ability to perform
 123 vector-based navigation, enabling downstream regions to calculate goal directed vectors by com-
 124 paring current activity with that of a remembered goal^{7,10,11}. In the agent, we expect these calcu-
 125 lations to be performed by the policy LSTM, which receives the current activity pattern over the
 126 linear layer (termed “current grid code”; see Fig. 2d and Extended Data Figure 5) as well as that
 127 present the last time the agent reached the goal (termed “goal grid code”), using them to control
 128 movement. Hence we performed several manipulations, which yielded four lines of evidence in

¹²⁹ support of the vector-based navigation hypothesis (see Supplemental Results for details).

¹³⁰ First, to demonstrate that the goal grid code provided sufficient information to enable the
¹³¹ agent to navigate to an arbitrary location, we substituted it with a "fake" goal grid code sampled
¹³² randomly from a location in the environment (see Methods). The agent followed a direct path to
¹³³ the newly specified location, circling the absent goal (Fig. 2i) — similar to rodents in probe trials
¹³⁴ of the Morris water maze (escape platform removed). Secondly, we demonstrated that withholding
¹³⁵ the goal grid code from the policy LSTM of the grid cell agent had a strikingly deleterious effect
¹³⁶ on performance (see Extended Data Fig. 6c). Thirdly, we demonstrated that the policy LSTM of
¹³⁷ the grid cell agent contained representations of key components of vector-based navigation (Figure
¹³⁸ 2j&k), and that both Euclidean distance (difference in $r = 0.17$; 95% CI [0.11, 0.24]) and allocentric
¹³⁹ goal direction (difference in $r = 0.22$; 95% CI [0.18, 0.26]) were represented more strongly than in
¹⁴⁰ the place cell agent. Notably, a neural representation of goal distance has recently been reported
¹⁴¹ in mammalian hippocampus²⁹. Finally, we provide evidence consistent with a prediction of the
¹⁴² vector-based navigation hypothesis, namely that a targeted lesion (i.e. silencing) to the most grid-
¹⁴³ like units within the goal grid code should have a greater adverse effect on performance and the
¹⁴⁴ representation of vector-based metrics (e.g. Euclidean distance) than a sham lesion (i.e silencing
¹⁴⁵ of non-grid units; average score for 100 episodes: 126.1 vs. 152.5, respectively; effect size = 0.38,
¹⁴⁶ 95% CI [0.34, 0.42] see Supplemental Results).

¹⁴⁷ Having demonstrated the effectiveness of grid-like representations in optimizing one-shot
¹⁴⁸ goal learning in a simple square arena, we assessed the agent's performance in complex, procedurally-

¹⁴⁹ generated multi-room environments, termed “goal-driven” and “goal-doors” (see Methods). No-
¹⁵⁰ tably, these environments are challenging for deep RL agents with external memory (see Extended
¹⁵¹ Data Figure 7e,f,h&i and Supplemental Results). Again, the grid-cell agent exhibited high levels
¹⁵² of performance, was strikingly robust across a range of network hyperparameters (see Extended
¹⁵³ Data Figure 7a,b&c), and reached the goal more frequently than either control agents or a human
¹⁵⁴ expert — a typical benchmark for the performance of deep RL agents in game playing scenarios²
¹⁵⁵ (Fig. 3e&f and see Supplemental Results). Further, when agents were tested, without retraining,
¹⁵⁶ in environments considerably larger than those seen previously, only the grid cell agent was able
¹⁵⁷ to generalise effectively (Fig. 3g&h and see Supplemental Results). Despite the complexity of the
¹⁵⁸ “goal-driven” environment, we could still decode the key metric codes from the grid agent policy
¹⁵⁹ LSTM with high accuracy during the initial period of navigation – with decoding accuracy sub-
¹⁶⁰ stantially higher in the grid cell agent than both the place cell and deep RL control agents (Figure
¹⁶¹ 3j&k and Supplemental Results, Extended Data Figures 8 and 9 for control agent architectures).

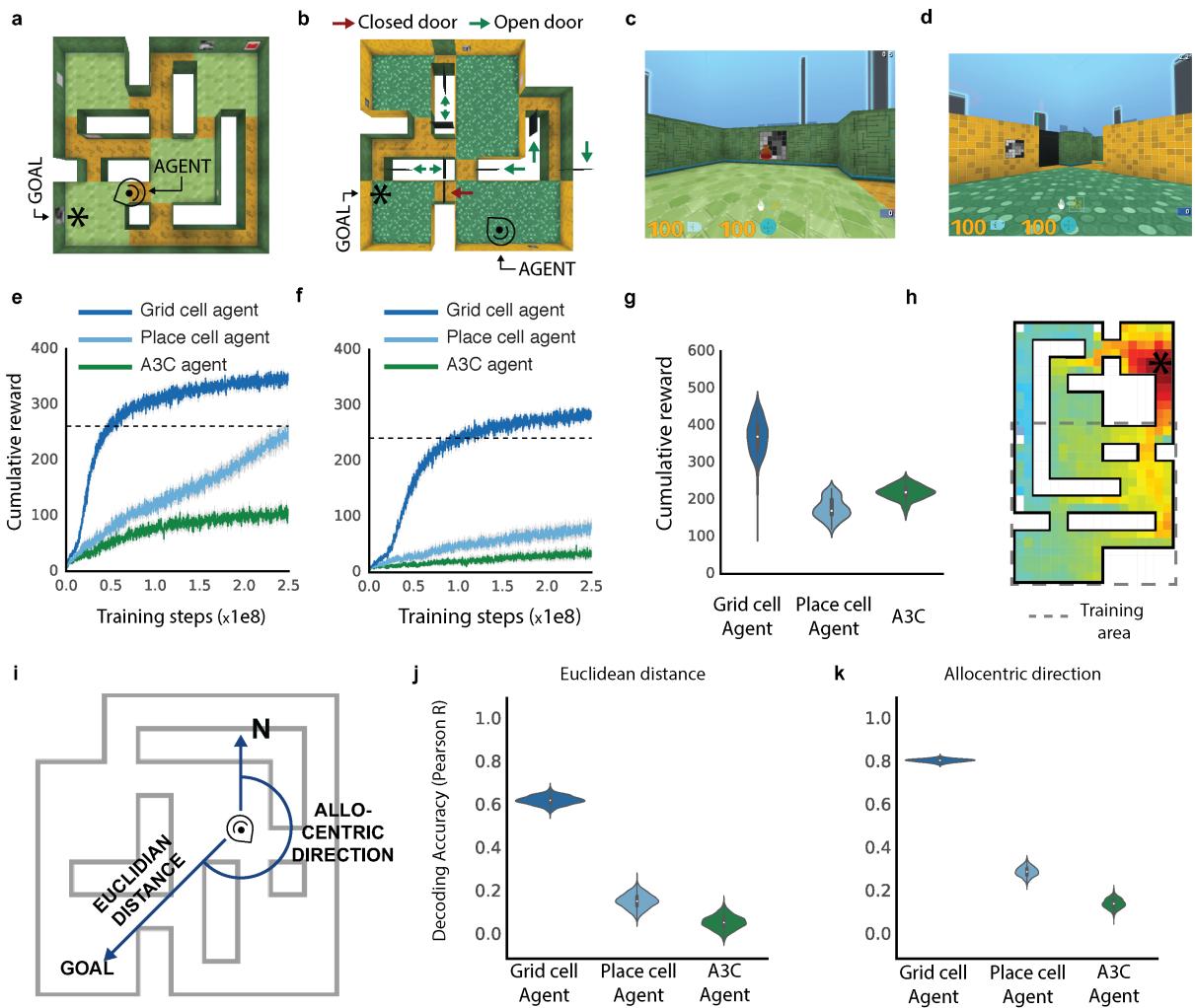


Figure 3: **Navigation in complex environments** a-b, Overhead view of multi-room environments “goal-driven” (a) and “goal-doors” (i.e. with stochastic doors) (b) Goal (*) and agent locations (head icon) are displayed. c-d, Agent views of (a) & (b) showing red goal and closed black door. e-f, Agent training performance curves for (a) & (b), and performance of human expert (dotted line). Performance is mean cumulative reward over 100 episodes. The gray band displays the 68% confidence interval based on 5000 bootstrapped samples g, Distribution of test performance over 100 episodes, showing ability of agents to generalize to a larger version of goal-driven environment, displayed with a violin plot overlaid on a Tukey boxplot for each agent. h, The value function of the grid cell agent is projected onto an example larger goal doors environment as a heatmap. Dotted lines show the extent of the original training environment. Despite the larger size, the value function clearly approximates Euclidean distance to goal. i, Schematic displaying the key metrics required for vector-based navigation to a goal. j-k, Decoding of vector-based metric codes from the policy LSTM of agents during navigation. The bootstrapped distribution (1000 samples) of correlation coefficients are displayed with a violin plot overlaid on a Tukey boxplot in each case.

163 Finally, a core feature of mammalian spatial behaviour is the ability to exploit novel short-
164 cuts and traverse unvisited portions of space, a capacity thought to depend on vector-based
165 navigation^{9,11}. Strikingly, the grid cell agent – but not comparison agents – robustly demonstrated
166 these abilities in specifically designed neuroscience-inspired mazes, taking direct routes to the goal
167 as soon as they became available (Fig. 4, Extended Data Figure 10 and Supplemental Results).

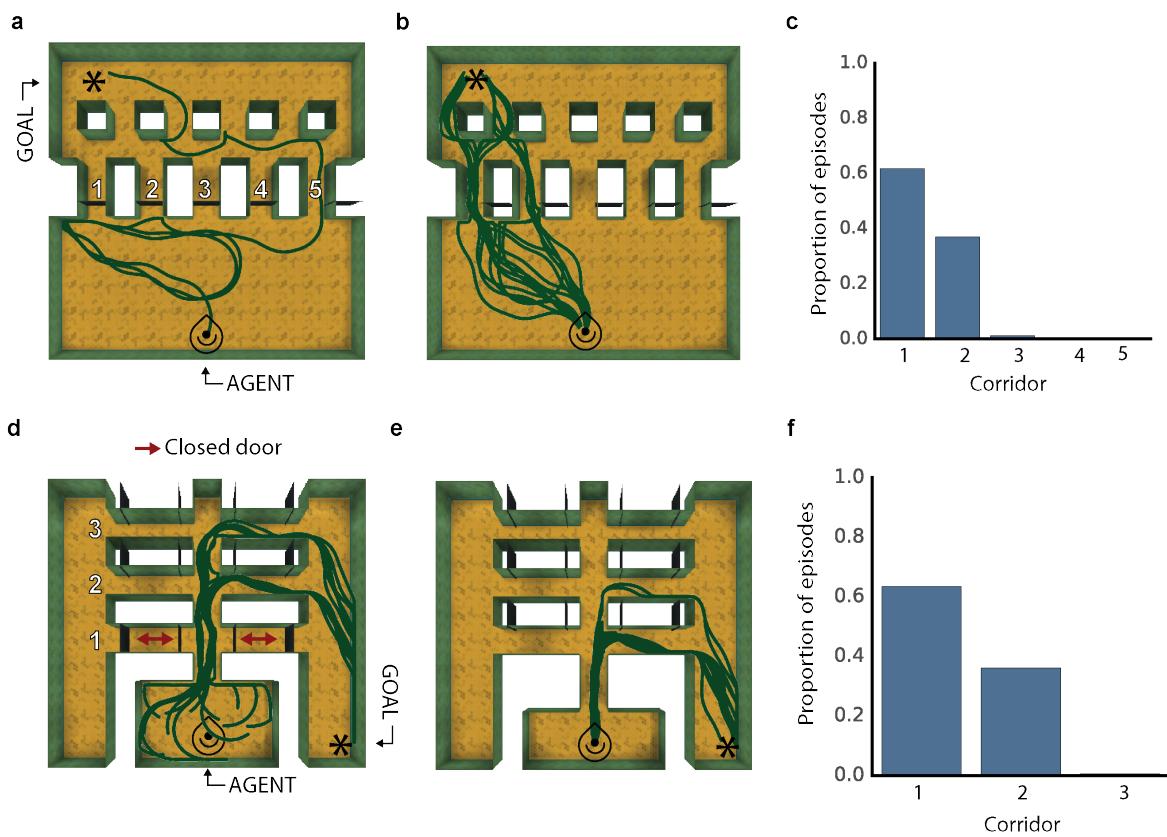


Figure 4: **Flexible use of short-cuts** a, Example trajectory from grid cell agent during training in the linear sunburst maze (only door 5 open; icon indicates start location). b, Testing configuration with all doors open: grid cell agent uses the newly available shortcuts (100 episodes shown). c, Histogram showing agent's strong preference for most direct doors. d, Example grid cell agent trajectories (100) during training in the double E-maze (corridor 1 doors closed). e, Testing configuration with corridor 1 open, and 100 grid agent trajectories. f, Histogram analogous to panel c, agent prefers newly-available shortest route. See Extended

¹⁶⁸ Data Figure 10 for performance of place cell agent.

169 Conventional simultaneous localization and mapping (SLAM) techniques typically require
170 an accurate and complete map to be built, with the nature and position of the goal externally
171 defined³⁰. In contrast, the deep reinforcement learning approach described in this work has the abil-
172 ity to learn complex control policies end-to-end from a sparse reward, taking direct routes involving
173 shortcuts to goals in an automatic fashion - abilities that exceed previous deep RL approaches³⁻⁵,
174 and that would have to be hand-coded in any SLAM system.

175 Our work, in demonstrating that grid-like representations provide an effective basis for flexi-
176 ble navigation in complex novel environments, supports theoretical models of grid cells in vector-
177 based navigation previously lacking strong empirical support^{7,10,11}. We also show that vector-based
178 navigation can be effectively combined with a path-based barrier avoidance strategy to enable the
179 exploitation of optimal routes in complex multi-compartment environments. In sum, we argue that
180 grid-like representations furnish agents with a Euclidean geometric framework — paralleling their
181 proposed computational role in mammals as an early-developing Kantian-like spatial scaffold that
182 serves to organize perceptual experience^{17,18}.

183

- 184 1. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
- 185 2. Silver, D. *et al.* Mastering the game of go with deep neural networks and tree search. *Nature*
186 **529**, 484–489 (2016).
- 187 3. Oh, J., Chockalingam, V., Singh, S. P. & Lee, H. Control of memory, active perception, and
188 action in minecraft. In *Proc. of International Conference on Machine Learning, ICML* (2016).

- 189 4. Kulkarni, T. D., Saeedi, A., Gautam, S. & Gershman, S. J. Deep successor reinforcement
190 learning. *CoRR* **abs/1606.02396** (2016). URL <http://arxiv.org/abs/1606.02396>.
- 191 5. Mirowski, P. *et al.* Learning to navigate in complex environments. *International Conference*
192 *on Learning Representations* (2017).
- 193 6. Hafting, T., Fyhn, M., Molden, S., Moser, M.-B. & Moser, E. I. Microstructure of a spatial
194 map in the entorhinal cortex. *Nature* **436**, 801–806 (2005).
- 195 7. Fiete, I. R., Burak, Y. & Brookings, T. What grid cells convey about rat location. *Journal of*
196 *Neuroscience* **28**, 6858–6871 (2008).
- 197 8. Mathis, A., Herz, A. V. & Stemmler, M. Optimal population codes for space: grid cells
198 outperform place cells. *Neural Computation* **24**, 2280–2317 (2012).
- 199 9. McNaughton, B. L., Battaglia, F. P., Jensen, O., Moser, E. I. & Moser, M.-B. Path integration
200 and the neural basis of the ‘cognitive map’. *Nature Reviews Neuroscience* **7**, 663–678 (2006).
- 201 10. Erdem, U. M. & Hasselmo, M. A goal-directed spatial navigation model using forward trajec-
202 tory planning based on grid cells. *European Journal of Neuroscience* **35**, 916–931 (2012).
- 203 11. Bush, D., Barry, C., Manson, D. & Burgess, N. Using grid cells for navigation. *Neuron* **87**,
204 507–520 (2015).
- 205 12. Barry, C. & Burgess, N. Neural mechanisms of self-location. *Current Biology* **24**, R330–R339
206 (2014).

- 207 13. Mittelstaedt, M.-L. & Mittelstaedt, H. Homing by path integration in a mammal. *Naturwissen-*
208 *schaften* **67**, 566–567 (1980).
- 209 14. Bassett, J. P. & Taube, J. S. Neural correlates for angular head velocity in the rat dorsal
210 tegmental nucleus. *Journal of Neuroscience* **21**, 5740–5751 (2001).
- 211 15. Kropff, E., Carmichael, J. E., Moser, M.-B. & Moser, E. I. Speed cells in the medial entorhinal
212 cortex. *Nature* **523**, 419–424 (2015).
- 213 16. Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a
214 simple way to prevent neural networks from overfitting. *Journal of Machine Learning Re-*
215 *search* **15**, 1929–1958 (2014).
- 216 17. Wills, T. J., Cacucci, F., Burgess, N. & O’keefe, J. Development of the hippocampal cognitive
217 map in preweanling rats. *Science* **328**, 1573–1576 (2010).
- 218 18. Langston, R. F. *et al.* Development of the spatial representation system in the rat. *Science* **328**,
219 1576–1580 (2010).
- 220 19. Zhang, S.-J. *et al.* Optogenetic dissection of entorhinal-hippocampal functional connectivity.
221 *Science* **340**, 1232627 (2013).
- 222 20. Sargolini, F. *et al.* Conjunctive representation of position, direction, and velocity in entorhinal
223 cortex. *Science* **312**, 758–762 (2006).
- 224 21. Barry, C., Hayman, R., Burgess, N. & Jeffery, K. J. Experience-dependent rescaling of en-
225 torhinal grids. *Nature neuroscience* **10**, 682–684 (2007).

- 226 22. Stensola, H. *et al.* The entorhinal grid map is discretized. *Nature* **492**, 72–78 (2012).
- 227 23. Stemmler, M., Mathis, A. & Herz, A. V. Connecting multiple spatial scales to decode the
228 population activity of grid cells. *Science advances* **1**, e1500816 (2015).
- 229 24. Doeller, C. F., Barry, C. & Burgess, N. Evidence for grid cells in a human memory network.
230 *Nature* **463**, 657–661 (2010).
- 231 25. Kanitscheider, I. & Fiete, I. Training recurrent networks to generate hypotheses about how the
232 brain solves hard navigation problems. *arXiv preprint arXiv:1609.09059* (2016).
- 233 26. Milford, M. J. & Wyeth, G. F. Mapping a suburb with a single camera using a biologically
234 inspired slam system. *IEEE Transactions on Robotics* **24**, 1038–1053 (2008).
- 235 27. Hardcastle, K., Ganguli, S. & Giocomo, L. M. Environmental boundaries as an error correction
236 mechanism for grid cells. *Neuron* **86**, 827–839 (2015).
- 237 28. Chen, G., King, J. A., Burgess, N. & O’Keefe, J. How vision and movement combine in
238 the hippocampal place code. *Proceedings of the National Academy of Sciences* **110**, 378–383
239 (2013).
- 240 29. Sarel, A., Finkelstein, A., Las, L. & Ulanovsky, N. Vectorial representation of spatial goals in
241 the hippocampus of bats. *Science* **355**, 176–180 (2017).
- 242 30. Dissanayake, M. G., Newman, P., Clark, S., Durrant-Whyte, H. F. & Csorba, M. A solution
243 to the simultaneous localization and map building (slam) problem. *IEEE Transactions on
244 Robotics and Automation* **17**, 229–241 (2001).

245 **Methods**

246 **Path integration: Supervised learning experiments.**

247 **Simplified 2D environment.** Simulated rat trajectories of duration T were generated in square
 248 and circular environments with walls of length L (diameter in the circular case). The simulated rat
 249 started at a uniformly sampled location and facing angle within the enclosure. A rat-like motion
 250 model³¹ was used to obtain trajectories that uniformly covered the whole environment by avoiding
 251 walls (see Table 1 in supplementary methods for the model's parameters).

252 **Ground truth place cell distribution.** Place cell activations, $\vec{c} \in [0, 1]^N$, for a given position
 253 $\vec{x} \in \mathbb{R}^2$, were simulated by the posterior probability of each component of a mixture of two-
 254 dimensional isotropic Gaussians,

$$c_i = \frac{e^{-\frac{\|\vec{x} - \vec{\mu}_i^{(c)}\|_2^2}{2(\sigma^{(c)})^2}}}{\sum_{j=1}^N e^{-\frac{\|\vec{x} - \vec{\mu}_j^{(c)}\|_2^2}{2(\sigma^{(c)})^2}}}, \quad (1)$$

255 where $\vec{\mu}_i^{(c)} \in \mathbb{R}^2$, the place cell centres, are N two-dimensional vectors chosen uniformly at random
 256 before training, and $\sigma^{(c)}$, the place cell scale, is a positive scalar fixed for each experiment.

257 **Ground truth head-direction cell distribution.** Head-direction cell activations, $\vec{h} \in [0, 1]^M$, for
 258 a given facing angle φ were represented by the posterior probability of a each component of a
 259 mixture of Von Mises distributions with concentration parameter $\kappa^{(h)}$,

$$h_i = \frac{e^{\kappa^{(h)} \cos(\varphi - \mu_i^{(h)})}}{\sum_{j=1}^M e^{\kappa^{(h)} \cos(\varphi - \mu_j^{(h)})}}, \quad (2)$$

260 where the M head direction centres $\mu_i^{(h)} \in [-\pi, \pi]$, are chosen uniformly at random before train-
261 ing, and $\kappa^{(h)}$ the concentration parameter is a positive scalar fixed for each experiment.

262 **Supervised learning inputs.** In the supervised setup the grid cell network receives, at each step t ,
263 the egocentric linear velocity $v_t \in \mathbb{R}$, and the sine and cosine of its angular velocity $\dot{\varphi}_t$.

Grid cell network architecture The grid cell network architecture (Extended Data Figure 1) consists of three layers: a recurrent layer, a linear layer, and an output layer. The single recurrent layer is an LSTM (*long short-term memory*³²) that projects to place and head direction units via the linear layer. The linear layer implements regularisation through dropout¹⁶. The recurrent LSTM layer consists of one cell of 128 hidden units, with no peephole connections. Input to the recurrent LSTM layer is the vector $[v_t, \sin(\dot{\varphi}_t), \cos(\dot{\varphi}_t)]$. The initial cell state and hidden state of the LSTM, \vec{l}_0 and \vec{m}_0 respectively, are initialised by computing a linear transformation of the ground truth place and head-direction cells at time 0:

$$\vec{l}_0 = W^{(cp)} \vec{c}_0 + W^{(cd)} \vec{h}_0 \quad (3)$$

$$\vec{m}_0 = W^{(hp)} \vec{c}_0 + W^{(hd)} \vec{h}_0 \quad (4)$$

264 The parameters of these two linear transformations ($W^{(cp)}$, $W^{(cd)}$, $W^{(hp)}$, and $W^{(hd)}$) were opti-
265 mised during training. The output of the LSTM, \vec{m}_t is then used to produce predictions of the place
266 cells \vec{y}_t and head direction cells \vec{z}_t by means of a linear decoder network.

267 The linear decoder consists of three sets of weights and biases: first, weights and biases that
268 map from the LSTM hidden state \vec{m}_t to the linear layer activations $\vec{g}_t \in \mathbb{R}^{512}$. The other two
269 sets of weights map from the linear layer activations \vec{g}_t to the predicted head directions, \vec{z}_t , and

²⁷⁰ predicted place cells, \vec{y}_t , respectively via softmax functions³³. Dropout¹⁶ with drop probability 0.5
²⁷¹ was applied to each \vec{g}_t unit. Note that there is no intermediary non-linearity in the linear decoder.

Supervised learning loss. The grid cell network is trained to predict the place and head-direction
cell ensemble activations, \vec{c}_t and \vec{h}_t respectively, at each time step t . During training, the network
was trained in a single environment where the place cell centres were constant throughout. The pa-
rameters of the grid cell network are trained by minimising the cross-entropy between the network
place cell predictions, \vec{y} , and the synthetic place-cells targets, \vec{c} , and the cross-entropy between
head-direction predictions, \vec{z} , and their targets, \vec{h} :

$$L(\vec{y}, \vec{z}, \vec{c}, \vec{h}) = - \sum_{i=1}^N c_i \log(y_i) - \sum_{j=1}^M h_j \log(z_j), \quad (5)$$

²⁷² Gradients of (5) with respect to the network parameters were calculated using backpropagation
²⁷³ through time³⁴, unrolling the network into blocks of 100 time steps. The network parameters
²⁷⁴ were updated using stochastic-gradient descent (RMSProp³⁵), with weight decay³⁶ for the weights
²⁷⁵ incident upon the bottleneck activations. Hyperparameter values used for training are listed in
²⁷⁶ Table 1.

²⁷⁷ **Gradient clipping** In our simulations gradient clipping was used for parameters projecting from
²⁷⁸ the dropout linear layer, g_t , to the place and head-direction cell predictions \vec{y}_t and \vec{z}_t . Gradient
²⁷⁹ clipping clips each element of the gradient vector to lie in a given interval $[-g_c, g_c]$. Gradient
²⁸⁰ clipping is an important tool for optimisation in deep and recurrent artificial neural networks where
²⁸¹ it helps to prevent exploding gradients³⁷. Gradient clipping also introduces distortions into the
²⁸² weight updates which help to avoid local minima³⁸.

283 **Navigation through Deep RL**

284 **Environments and Task** We assessed the performance of agents on three environments seen by

285 the agent from a first-person perspective in the DeepMind Lab³⁹ platform.

286 **Custom Environment: Square Arena** This comprised a 10×10 square arena - which correspond

287 to 2.5×2.5 meters assuming an agent speed of 15 cm/s (Fig. 2b, c). The arena contained a single,

288 coloured, intra-arena cue whose position and colour changed each episode — as did the texture

289 of the floor, the texture of the walls and the goal location. As in the goal-driven and goal-door

290 environments described below, there were a set of distal cues (i.e. buildings) that paralleled the de-

291 sign of virtual reality environments used in human experiments²⁴. These distal cues were rendered

292 at infinity — so as to provide directional but not distance information — and their configuration

293 was consistent across episodes. At the start of each episode the agent (described below) started in

294 a random location and was required to explore in order to find an unmarked goal, paralleling the

295 task of rodents in the classic Morris water maze. The agent always started in the central 6×6 grids

296 (i.e. 1.5×1.5 meters) of the environment. Noise in the velocity input \vec{u}_t was applied throughout

297 training and testing (i.e. Gaussian noise ϵ , with $\mu = 0$ and $\sigma = 0.01$). The action space is discrete

298 (six actions) but affords fine-grained motor control (i.e. the agent could rotate in small increments,

299 accelerate forward/backward/sideways, or effect rotational acceleration while moving).

300 **DeepMind Lab Environments: Goal-Driven and Goal-Doors** Goal-driven and Goal-Doors are

301 complex, visually-rich multi-room environments (see Fig. 3a-d). Mazes were formed within an

302 11×11 grid, corresponding to 2.7×2.7 meters, (see below for definition of larger 11×17 mazes).

303 Mazes were procedurally generated at the beginning of each episode; thus, the layout, wall textures,
304 landmarks (i.e. intra-maze cues on walls) and goal location were different for each episode but
305 consistent within an episode. Distal cues, in the form of buildings rendered at infinity, were as
306 described for the square arena (see above).

307 The critical difference between goal-driven and goal-doors tasks is that the latter had the
308 additional challenge of stochastic doors within the maze. Specifically, the state of the doors (i.e.
309 open or closed) randomly changed during an episode each time the agent reached the goal. This
310 meant that the optimal path to the goal from a given location changed during an episode – requiring
311 the agent to recompute trajectories.

312 In both tasks the agent starts at a random location within the maze and its task is to explore
313 to find the goal. The goal in both levels was always represented by the same object (see Fig. 3c).
314 After getting to the goal the agent received a reward of 10 points after which it was teleported to
315 a new random location within the maze. In both levels, episodes lasted a fixed duration of 5,400
316 environment steps (90 seconds).

317 **Generalisation on larger environments.** We tested the ability of agents trained on the standard
318 environment (11×11) to generalise to larger environments (11×17 , corresponding to 2.7×4.25
319 meters). The procedural generation and composition of these environments was done as with the
320 standard environments. Each agent was trained in the 11×11 goal-driven maze for a total of 10^9
321 environment steps, and the best performing replica (i.e. highest asymptotic performance averaged
322 over 100 episodes in 11×11) was selected for evaluation in the larger maze. Note that the weights

323 of the agent were frozen during evaluation on the larger maze. Evaluation was over 100 episodes
324 of fixed duration 12,600 environment steps (210 seconds).

325 **Probe mazes to assess shortcut behaviour** To test the agent's ability to follow novel, goal-
326 directed routes, we created a series of environments inspired by mazes designed to test the shortcut
327 abilities of rodents.

328 The first maze is a linearised version of Tolman's sunburst maze (Fig. 4a) used to determine
329 if the agent was able to follow an accurate heading towards the goal when a path became available
330 (see Supplementary Methods for details). In this maze, after reaching the goal, the agent was
331 teleported to the original position with the same heading orientation. Here we tested agents trained
332 in the "goal doors" environments. Specifically, the network weights were held frozen during testing
333 and all the agents were tested for 100 episodes, each one lasting for a fixed duration of 5,400
334 environment steps (90 seconds).

335 The second environment, the double E-maze (Fig. 4d), was designed to test the agent's ability
336 to traverse an entirely novel portion of space (see Supplementary Methods for details). In this maze
337 we had a training and a testing condition. During the former agents were trained as in the other
338 mazes (e.g. goal-driven; training details given below), whereas at test time weights were frozen.
339 The agent always started in the central room (e.g. see Fig. 4d). The maze had stochastic doors
340 with two different configurations, one for the training phase and one for testing phase. During
341 training the state of the doors (i.e. open or closed) randomly changed during an episode each time
342 the agent reached the goal. Critically, during training the corridors presenting the shortest route to

343 the goal (i.e. the ones closer to the central room) were closed at both ends, preventing access or
344 observation of the interior. At test time, after the agent reached the goal the first time, all doors
345 were opened. All the agents were tested for 100 episodes, each one lasting for a fixed duration of
346 5,400 environment steps (90 seconds).

347 **Agent Architectures**

348 **Architecture for the Grid Cell Agent.** The agent architecture (see Extended Data Figure 5) was
349 composed of a visual module, of the grid cell network (described above), and of an actor-critic
350 learner⁴⁰. The visual module was a neural network with input consisting of a three channel (RGB)
351 64×64 image $\phi \in [-1, 1]^{3 \times 84 \times 84}$. The image was processed by a convolutional neural network
352 (see Supplementary Methods for the details of the convolutional neural network), which produced
353 embeddings, \vec{e} , which in turn were used as input to a fully connected linear layer trained in a super-
354 vised fashion to predict place and head-direction cell ensemble activations, \vec{c} and \vec{h} (as specified
355 above), respectively. The predicted place and head direction cell activity patterns were provided
356 as input to the grid network 5% of the time on average, akin to occasional imperfect observations
357 made by behaving animals of salient environmental cues²⁷. Specifically, the output of the convolu-
358 tional network \vec{e} is then passed through a masking layer which zeroed the units with a probability
359 of 95%.

360 The grid cell network of the agent was implemented as in the supervised learning set up
361 except that the LSTM (“GRID LSTM”) was not initialised based upon ground truth place cell
362 activations but rather set to zero. The input to the grid cell network were the two translational

363 velocities, u and v , as in DeepMind Lab it is possible to move in a direction different from the
364 facing direction, and the sine and cosine of the angular velocity, $\dot{\varphi}$, (these velocities are provided
365 by DeepMind Lab) — and additionally the \vec{y} and \vec{z} output by the vision module. In contrast to the
366 supervised learning case, here the grid cell network had to use \vec{y} and \vec{z} to learn how to reset its
367 internal state each time it was teleported to an arbitrary location in the environment (e.g. after visit
368 to goal). As in the supervised learning experiments described above, the configuration of place
369 fields (i.e. location of place field centres in the 11×11 environments, “goal-driven” and “goal-
370 doors”, 10×10 square arena, and 13×13 double E) were constant throughout training (i.e. across
371 episodes).

372 For the actor-critic learner the input was a three channel (RGB) 64×64 image $\phi_t \in$
373 $[-1, 1]^{3 \times 64 \times 64}$, which was processed by a convolutional neural network followed by a fully con-
374 nected layer (see Supplementary Methods for the details of the convolutional neural network). The
375 output of the fully connected layer of the convolutional network \vec{e}_t^1 was then concatenated with
376 the reward r_t , the previous action a_{t-1} , the current “grid code”, \vec{g}_t , goal “grid code”, \vec{g}_* (i.e.
377 linear layer activations observed last time the goal was reached) — or zeros if the goal had not
378 yet been reached in the episode. Note we refer to these linear layer activations as “grid codes”
379 for brevity, even though units in this layer comprise also units resembling head direction cells, and
380 border cells (e.g. see Extended Figure 6a). This concatenated input was provided to an LSTM with
381 256 units. The LSTM had 2 different outputs. The first output, the actor, is a linear layer with 6
382 units followed by a softmax activation function, that represents a categorical distribution over the
383 agent’s next action. The second output, the critic, is a single linear unit that estimates the value

³⁸⁴ function. Note that we refer to this as the "policy LSTM" for brevity, even though it also outputs
³⁸⁵ the value function.

³⁸⁶ **Comparison agents** We compared the performance of the grid cell agent against two agents
³⁸⁷ specifically because they use a different representational scheme for space (i.e. place cell agent,
³⁸⁸ place cell prediction agent), and relate to theoretical models of goal-directed navigation from the
³⁸⁹ neuroscience literature (e.g. ^{41,42}). We also compared the grid cell agent against a baseline deep
³⁹⁰ RL agent, Asynchronous Advantage Actor-Critic (A3C)⁴⁰.

³⁹¹ **Place cell agent.** The place cell agent architecture is shown in Extended Data Figure 8b. In con-
³⁹² trast to the grid cell agent, the place cell agent used ground truth information: specifically, the
³⁹³ ground-truth place, \vec{c}_t , and head-direction, \vec{h}_t , cell activations (as described above). These activity
³⁹⁴ vectors were provided as input to the policy LSTM in an analogous way to the provision of grid
³⁹⁵ codes in the grid cell agent.

³⁹⁶ Specifically, the output of the fully connected layer of the convolutional network \vec{e}_t was
³⁹⁷ concatenated with the reward r_t , the previous action a_{t-1} , the ground-truth current place code,
³⁹⁸ \vec{c}_t , and current head-direction code, \vec{h}_t — together with the ground truth goal place code, \vec{c}_* , and
³⁹⁹ ground truth head direction code, \vec{h}_* , observed last time the goal was reached — or zeros if the
⁴⁰⁰ goal had not yet been reached in the episode (see Extended Data Figure 8b). The convolutional
⁴⁰¹ network had the same architecture described for the grid cell agent.

⁴⁰² **Place cell prediction agent.** The architecture of the place cell prediction agent (Extended Data
⁴⁰³ Figure 9a) is similar to the grid cell agent described above: the key difference is the nature of the

404 input provided to the policy LSTM as described below. The place cell prediction agent had a grid
405 cell network — with the same parameters as that of the grid cell agent. However, instead of using
406 grid codes from the linear layer of the grid network \vec{g} , as input for the policy LSTM (i.e. as in
407 the grid cell agent), we used the predicted place cell population activity vector \vec{y} and the predicted
408 head direction population activity vector \vec{z} (i.e. the activations present on the output place and head
409 direction unit layers of the grid cell network at each timestep) (see Supplementary Methods).

410 The critical difference between the place cell agent and the place cell prediction agent (see
411 Extended Data Figure 8b and 9a respectively) is that the former used ground truth information (i.e.
412 place and head direction cell activations for current location and goal location) - whereas the latter
413 used the population activity produced across the output place and head direction cell layers (i.e.
414 for current location and goal location) by the linear layer of the same grid network as utilised by
415 the grid cell agent.

416 **A3C** We implemented the asynchronous advantage actor-critic architecture described in⁴⁰ with
417 convolutional network having the same architecture described for the grid cell agent (Extended
418 Data Figure 8a).

419 **Other Agents** We also assessed the performance of two deep RL agents with external mem-
420 ory (Extended Data Figure 9b), which served to establish the challenging nature of the multi-
421 compartment environments (goal-doors and goal-driven). First, we implemented a memory net-
422 work agent (“NavMemNet”) consisting of the FRMQN architecture³ but instead of Q-learning we
423 used the Asynchronous Advantage Actor-Critic (A3C) algorithm described below. Further, the

424 input to memory was generated as an output from the LSTM controller (Extended Data Figure
 425 9b), rather than constituting embeddings from the convolutional network (i.e. as in³). The convo-
 426 lutional network had the same architecture described for the grid cell agent and the memory was
 427 formed of 2 banks (keys and values), each one with 1350 slots.

428 Second, we implemented a Differentiable Neural Computer (“DNC”) agent which uses
 429 content-based retrieval and writes to the most recently used or least recently used memory slot.⁴³

430 **Training algorithms** We used the Asynchronous Advantage Actor-Critic (A3C) algorithm⁴⁰,
 431 which implements a policy, $\pi(a|s, \theta)$, and an approximation to its value function, $V(s, \theta)$, using a
 432 neural network parameterised by θ . A3C adjusts the network parameters using n -step lookahead
 433 values, $\hat{R}_t = \sum_{i=0...n-1} \gamma^i r_{t+i} + \gamma^n V(s_{t+n}, \theta)$, to minimise: $\mathcal{L}_{\text{A3C}} = \mathcal{L}_{\pi} + \alpha \mathcal{L}_{\text{V}} + \beta \mathcal{L}_{\text{H}}$, where
 434 $\mathcal{L}_{\pi} = -\mathbb{E}_{s_t \sim \pi} [\hat{R}_t]$, $\mathcal{L}_{\text{V}} = \mathbb{E}_{s_t \sim \pi} \left[(\hat{R}_t - V(s_t, \theta))^2 \right]$, $\mathcal{L}_{\text{H}} = -\mathbb{E}_{s_t \sim \pi} [H(\pi(\cdot|s_t, \theta))]$. Where \mathcal{L}_{H} is
 435 a policy entropy regularisation term (see Supplementary Methods for details of the reinforcement
 436 learning approach). The grid cell network and the vision module were trained with the same loss
 437 reported for supervised learning: $\mathcal{L}(\vec{y}, \vec{z}, \vec{c}, \vec{h}) = -\sum_{i=1}^N c_i \log(y_i) - \sum_{j=1}^M h_j \log(z_j)$

438 **Agent training details.** We follow closely the approach of⁴⁰. Each experiment used 32 actor-critic
 439 learner threads running on a single CPU machine. All threads applied updates to their gradients
 440 every 4 actions (i.e. action repeat of 4) using RMSProp with shared gradient statistics⁴⁰. All the
 441 experiments were run for a total of 10^9 environment steps.

442 In architectures where the grid cell network and the vision module were present we used
 443 a shared buffer^{44,45} where we stored the agents experiences at each time-step, $e_t = (\phi_t, u_t, v_t)$,

444 collected over many episodes. All the 32 actor-critic workers were updating the same shared
445 buffer which had a total size of $20e6$ slots. The vision module was trained with mini batches of
446 size 32 frames ($\vec{\phi}$) sampled randomly from the replay buffer. The grid cell network was trained
447 with mini batches of size 10, randomly sample from the buffer, each one comprising a sequence
448 of 100 consecutive observations, $[\vec{\phi}, \vec{u}, \vec{v}]$. These mini batches were firstly forwarded through the
449 vision module to get \vec{c} , and \vec{h} , which were then passed trough a masking layer which masked them
450 to 0 with a probability of 95% (i.e. as described above in section on grid cell architecture). The
451 output of this masking layer was then concatenate with \vec{u} , \vec{v} , $\sin\vec{\phi}$, $\cos\vec{\phi}$, which were then used
452 as inputs to the grid network, as previously described (see Extended Data Figure 5 for details).
453 Both networks were trained using one single thread, one to train the vision module and another to
454 train the grid network (so in total we used 34 threads). Also, there was no gradient sharing between
455 the actor-critic learners, the vision module and the grid network.

456 The hyperparameters of the grid cell network were kept fixed across all the simulations and
457 were derived from the best performing network in the supervised learning experiments. For the
458 hyperparameter details of the vision module, the grid network and the actor-critic learner please
459 refer to Table 2. For each of the agents in this paper, 60 replicas were run with hyperparameters
460 sampled from the same interval (see Table 2) and different initial random seeds.

461 **Details for lesion experiment** To conduct a lesioning experiment in the agent we trained the grid
462 cell agent with dropout applied on the goal grid code input \vec{g}_* . Specifically, every 100 training steps
463 we generated a random mask to silence 20% of the units in the goal grid code (\vec{g}_*) - i.e. units were
464 zeroed. This procedure was implemented to ensure that the policy LSTM would become robust

465 through training to receiving a lesioned input (i.e. would not catastrophically fail), and still be able
466 to perform the task.

467 We then selected the agent with the best performance over 100 episodes, and we computed
468 the grid score of all units found in \vec{g} . The critical comparison to test the importance of grid-like
469 units to vector-based navigation was as follows. In one condition we ran 100 testing episodes
470 where we silenced the 25% units in \vec{g}_* with the highest grid scores. In the other condition, we ran
471 100 testing episodes with the same agent with 25% random units in \vec{g}_* silenced. In this second case
472 we ensured head direction cells with a resultant vector length of more than 0.47 were not silenced,
473 to preserve crucial head direction signals. We then compared the performance, and representation
474 of metrics relating to vector-based navigation, of the agents under these two conditions.

475 **Details of experiment using "fake" goal grid code** To demonstrate that the goal grid code pro-
476 vided sufficient information to enable the agent to navigate to an arbitrary location we took an
477 agent trained in the square arena, we froze the weights and we ran it in the same square arena for
478 5,400 steps. Critically, after the 6th time that the agent reach the goal, we sampled the grid code
479 from a random point that the agent visited in the environment (called fake goal grid code). We
480 then substituted the true goal grid code with this fake goal grid code, to show that this would be
481 sufficient to direct the agent to a location where there was no actual goal.

482 **Agent Performance** For evaluating agent performance during training (as in Fig. 2f, Fig. 3e,f)
483 we selected the 30 replicas (out of 60) which had the highest average cumulative reward across
484 100 episodes. Also we assessed the robustness of the architecture over different initial random

485 seeds and the hyperparameters in Table 2 by calculating the area under the curve (AUC). To plot
486 the AUC we ran 60 replicas with hyperparameters sampled from the same interval (see Table 2)
487 and different initial random seeds (Extended Data Figure 7a-c).

488 **Neuroscience-based analyses of network units**

489 **Generation of activity maps** Spatial (ratemaps) and directional activity maps were calculated
490 for individual units as follows. Each point in the trajectory was assigned to a specific spatial and
491 directional bin based on its location and direction of facing. Spatial bins were defined as a 32×32
492 square grid spanning each environment and directional bins as 20 equal width intervals. Then, for
493 each unit, the mean activity over all the trajectories points assigned to that bin was found. These
494 values were displayed and analysed further without additional smoothing.

495 **Inter-trial stability** For each unit the reliability of spatial firing between baseline trials was as-
496 sessed by calculating the spatial correlation between pairs of rate maps taken at 2 different logging
497 steps in training ($t = 2e5$; $t' = 3e5$). The total training time was $3e5$ so the points were selected
498 with enough time difference to minimise the chances of finding random correlations. The Pearson
499 product moment correlation coefficient was calculated between equivalent bins in the two trials
500 and unvisited bins were excluded from the measure.

501 **Quantification of spatial activity** Where possible, we assessed the spatial modulation of units
502 using measures adopted from the neuroscience literature. The hexagonal regularity and scale of
503 grid-like patterns were quantified using the gridness score^{18,20} and grid scale²⁰, measures derived
504 from the spatial autocorelogram²⁰ of each unit's ratemap. Similarly, the degree of directional

modulation exhibited by each unit was assessed using the length of the resultant vector⁴⁶ of the directional activity map. Finally, the propensity of units to fire along the boundaries of the environment was quantified with the border score⁴⁷.

The gridness and border scores exhibited by units in the linear layer were benchmarked against the 95th percentile of null distributions obtained using a permutation procedure (spatial field shuffle⁴⁸) applied to each unit's ratemap. This shuffling procedure aimed to preserve the local topography of fields within each ratemap while distributing the fields themselves at random⁴⁸.

The means, over units, of the thresholds obtained were gridness > 0.37 and border score > 0.50. Units exceeding these thresholds were considered to be grid-like and border-like, respectively. To identify directionally modulated cells we applied Rayleigh tests of directional uniformity to the binned directional activity maps. A unit was considered to be directionally modulated if the null hypothesis of uniform was rejected at the $\alpha = 0.01$ level - corresponding to units with resultant vector length in excess of 0.47 (See Supplementary Methods for further details).

Clustering of scale in grid-like units To determine if grid-like units exhibited a tendency to cluster around specific scales we applied two methods. First, following²², to determine if the scales of grid-like units (gridness > 0.37, 129/512 units) followed a continuous or discrete distribution we calculated the discreteness measure²² of the distribution of their scales (see Supplementary Methods). The discreteness score of the real data was found to exceed that of all of the 500 shuffles. Second, to characterise the number and location of scale clusters, the distribution of scales from grid-like units was fit with Gaussian mixture distributions, 3 components were found to provide the most parsimonious fit, indicating the presence of 3 scale clusters. (See Supplementary Methods

526 for further details.)

527 **Multivariate decoding of representation of metric quantities within LSTM** To test whether the
528 grid agent learns to use the predicted vector based navigation (VBN) metric codes, we recorded the
529 activation from the hidden units of the the Policy LSTM layer while the agent navigated 200 hun-
530 dred episodes in the land maze. We used L2-regularized (ridge) regression to decode Euclidean
531 distance and allocentric direction to the goal (see Supplementary Methods for full decoding de-
532 tails). We specifically focussed on twelve steps (steps 9-21) during the early portion of navigation,
533 but after the agent has had time to accurately self-localize. It is this early period after the agent has
534 reached the goal for the first time where a VBN strategy should be most effective. We conducted
535 the same analysis on the place cell agent control which is not predicted to use vector-based nava-
536 gation as efficiently. The decoding accuracy was measured as the correlation between predicted and
537 actual metric values in held-out data. Decoding accuracy was compared across different agents by
538 assessing the difference in decoding correlations between the agents. A bootstrap method (using
539 10,000 samples) was used to computed a 95% confidence interval on this correlation difference,
540 and these are reported for each comparison. The same approach was used to decode and compare
541 these two metrics in the lesioned agents on the land maze. Finally, to explore VBN metrics in a
542 more complex environment, the same method was applied to the goal-driven task. In this case we
543 also investigated metric decoding in the control A3C agent.

544 **Data availability statement** All reinforcement learning tasks described throughout the paper were
545 built using the publicly available DeepMind Lab platform (<https://github.com/deepmind/lab>). We
546 expect to release this set of tasks through this platform in the near future.

547 **Code availability statement** We will release the code for the supervised learning experiments
548 within the next six months. The codebase for the deep RL agents makes use of proprietary com-
549 ponents, and we are unable to publicly release this code. However, all experiments and agents are
550 described in sufficient detail to allow independent replication.

551

552 31. Raudies, F. & Hasselmo, M. E. Modeling boundary vector cell firing given optic flow as a cue.
553 *PLoS computational biology* **8**, e1002553 (2012).

554 32. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural computation* **9**, 1735–1780
555 (1997).

556 33. Bridle, J. S. Training stochastic model recognition algorithms as networks can lead to max-
557 imum mutual information estimation of parameters. In Touretzky, D. S. (ed.) *Advances in*
558 *Neural Information Processing Systems 2*, 211–217 (Morgan-Kaufmann, 1990).

559 34. Elman, J. L. & McClelland, J. L. Exploiting lawful variability in the speech wave. *Invariance*
560 *and variability in speech processes* **1**, 360–380 (1986).

561 35. Tieleman, T. & Hinton, G. Lecture 6.5—RmsProp: Divide the gradient by a running average
562 of its recent magnitude. COURSERA: Neural Networks for Machine Learning (2012).

563 36. MacKay, D. J. A practical bayesian framework for backpropagation networks. *Neural com-*
564 *putation* **4**, 448–472 (1992).

565 37. Pascanu, R., Mikolov, T. & Bengio, Y. On the difficulty of training recurrent neural networks.
566 *ICML (3)* **28**, 1310–1318 (2013).

- 567 38. Ackley, D. H., Hinton, G. E. & Sejnowski, T. J. A learning algorithm for boltzmann machines.
- 568 *Cognitive science* **9**, 147–169 (1985).
- 569 39. Beattie, C. *et al.* Deepmind lab. *CoRR* **abs/1612.03801** (2016). URL <http://arxiv.org/abs/1612.03801>.
- 570
- 571 40. Mnih, V. *et al.* Asynchronous methods for deep reinforcement learning. In *Proceedings of the*
- 572 *33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA,*
- 573 *June 19-24, 2016*, 1928–1937 (2016).
- 574 41. Touretzky, D. S., Redish, A. D. *et al.* Theory of rodent navigation based on interacting repre-
- 575 sentations of space. *Hippocampus* **6**, 247–270 (1996).
- 576 42. Foster, D., Morris, R., Dayan, P. *et al.* A model of hippocampally dependent navigation, using
- 577 the temporal difference learning rule. *Hippocampus* **10**, 1–16 (2000).
- 578 43. Graves, A. *et al.* Hybrid computing using a neural network with dynamic external memory.
- 579 *Nature* **538**, 471–476 (2016).
- 580 44. Mnih, V. *et al.* Human-level control through deep reinforcement learning. *Nature* **518**, 529–
- 581 533 (2015).
- 582 45. Lin, L.-J. Reinforcement learning for robots using neural networks. Tech. Rep., Carnegie-
- 583 Mellon Univ Pittsburgh PA School of Computer Science (1993).
- 584 46. Knight, R. *et al.* Weighted cue integration in the rodent head direction system. *Philosophical*
- 585 *Transactions of the Royal Society of London B: Biological Sciences* **369**, 20120512 (2014).

586 47. Solstad, T., Boccara, C. N., Kropff, E., Moser, M.-B. & Moser, E. I. Representation of geo-
587 metric borders in the entorhinal cortex. *Science* **322**, 1865–1868 (2008).

588 48. Barry, C. & Burgess, N. To be a grid cell: Shuffling procedures for determining gridness.
589 *BiorXiv* (2017).

590 There is Supplemental Information that contains additional results, discussion and details
591 about the Methods.

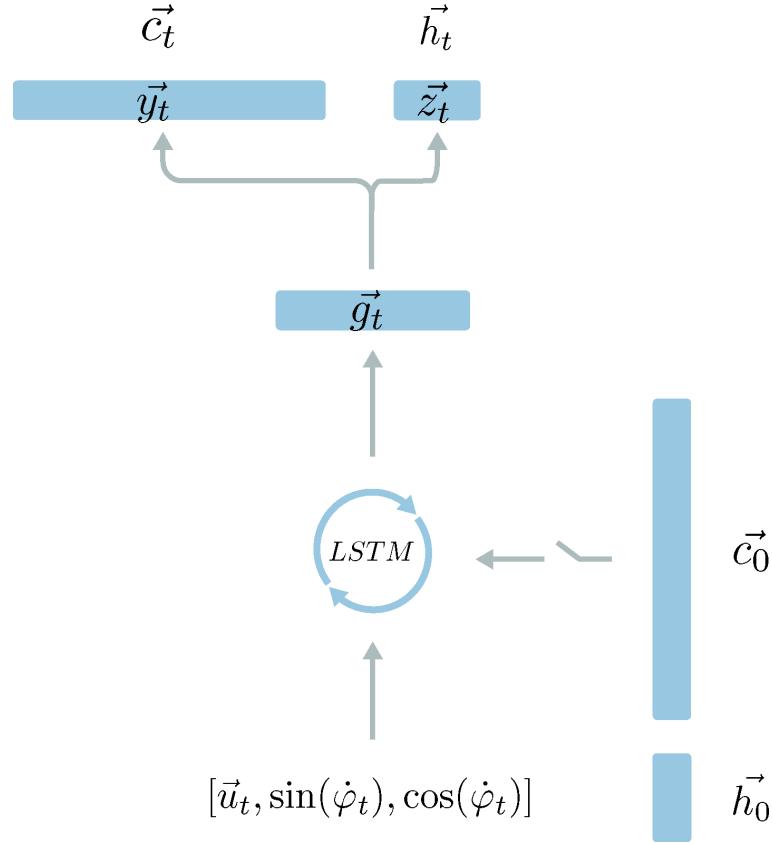
592 **Acknowledgements** We thank Max Jaderberg, Volodymyr Mnih, Adam Santoro, Tom Schaul, Kim
593 Stachenfeld and Jason Yosinski for discussions; Matthew Botvinick, and Jane Wang for comments on an
594 earlier version of the manuscript . C.Ba. funded by Royal Society and Wellcome Trust.

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

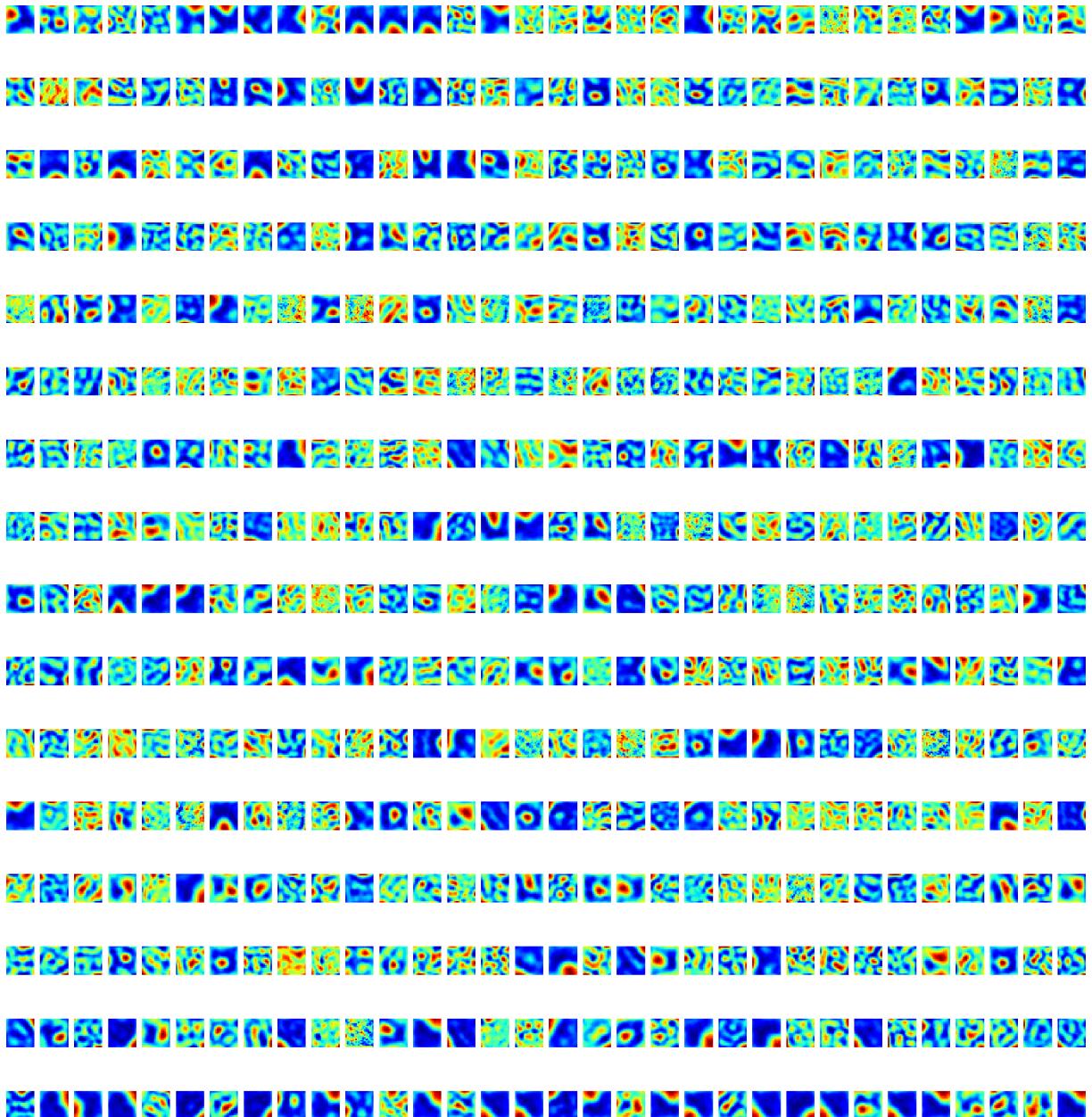
596 **Correspondence** Correspondence and requests for materials should be addressed to Andrea Ban-
597 ino, Caswell Barry, Dharshan Kumaran (email: abanino@google.com, caswell.barry@ucl.ac.uk, dku-
598 maran@google.com).

599 **Author Contributions** Conceived project. A.B., D.K., C.Ba., R.H., P.M., B.U., Contributed ideas to ex-
600 periments. A.B., D.K., C.Ba., B.U., R.H., T.L., C.BI., P.M., A.P., T.D., J.M., K.K., N.R., G.W., R.G., D.H.,
601 R.P. Performed experiments and analysis: A.B., C.Ba., B.U., M.C., T.L., H.S., A.P., B.Z, F.V. Development
602 of testing platform and environments. C.Be., S.P., R.H., T.L., G.W., D.K., A.B., B.U., D.H. Human expert
603 tester. A.S. Managed project. D.K., R.H., A.B., H.K., S.G., D.H. Wrote paper. D.K., A.B., C.Ba., T.L.,
604 C.BI., B.U., M.C., A.P., R.H., N.R., K.K., D.H.

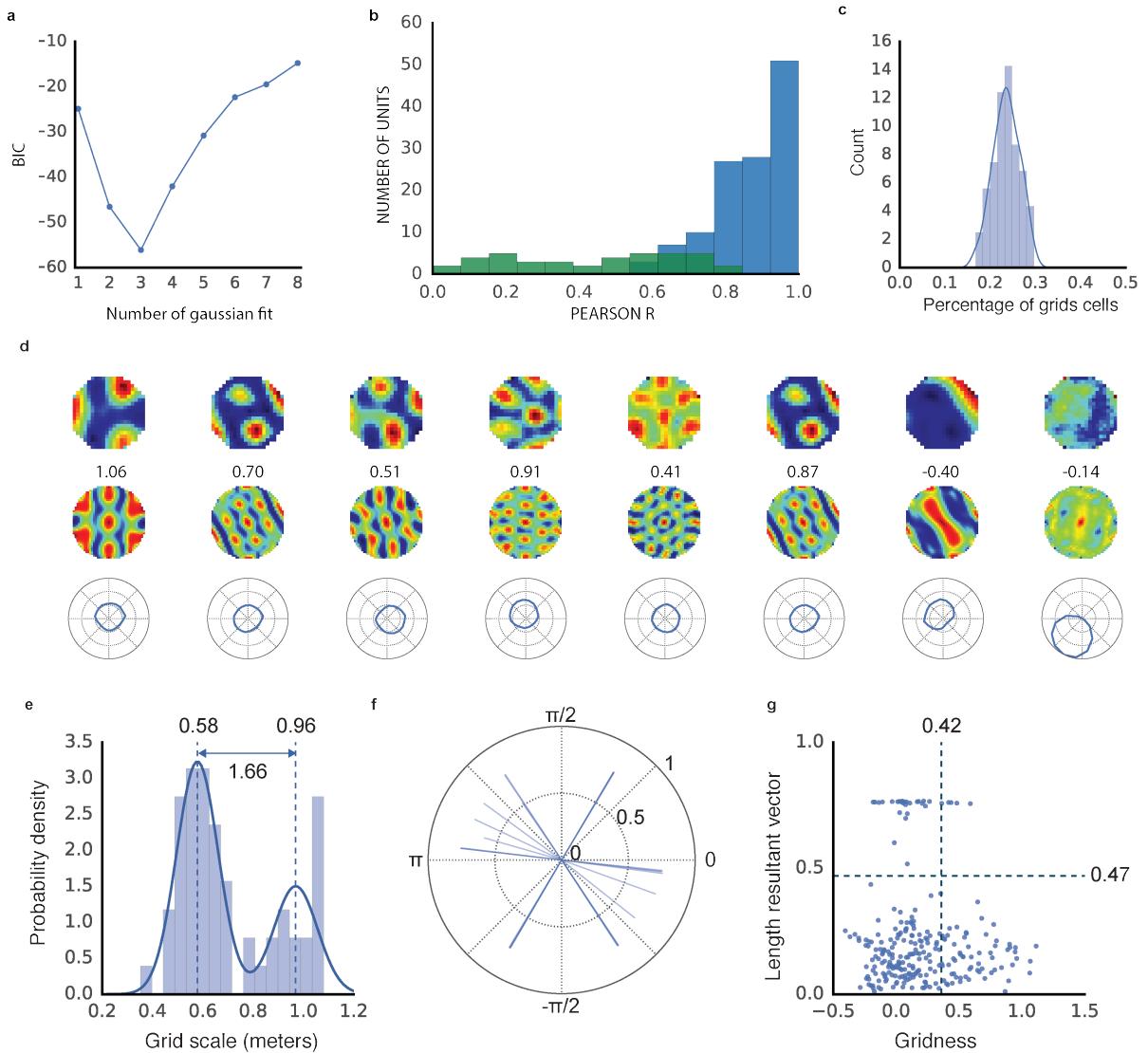
$$L(\vec{y}, \vec{z}, \vec{c}, \vec{h}) = - \sum_{i=1}^N c_i \log(y_i) - \sum_{j=1}^M h_j \log(z_j)$$



Extended Data Figure 1: Network architecture in the supervised learning experiment. The recurrent layer of the grid cell network is an LSTM with 128 hidden units. The recurrent layer receives as input the vector $[\vec{u}_t, \sin(\dot{\varphi}_t), \cos(\dot{\varphi}_t)]$. The initial cell state and hidden state of the LSTM, \vec{l}_0 and \vec{m}_0 respectively, are initialised by computing a linear transformation of the ground truth place \vec{c}_0 and head-direction \vec{h}_0 activity at time 0. The output of the LSTM is followed by a linear layer on which dropout is applied. The output of the linear layer, \vec{g}_t , is linearly transformed and passed to two softmax functions that calculate the predicted head direction cell activity, \vec{z}_t , and place cell activity, \vec{y}_t , respectively. We found evidence of grid-like and head direction-like units in the linear layer activations \vec{g}_t .

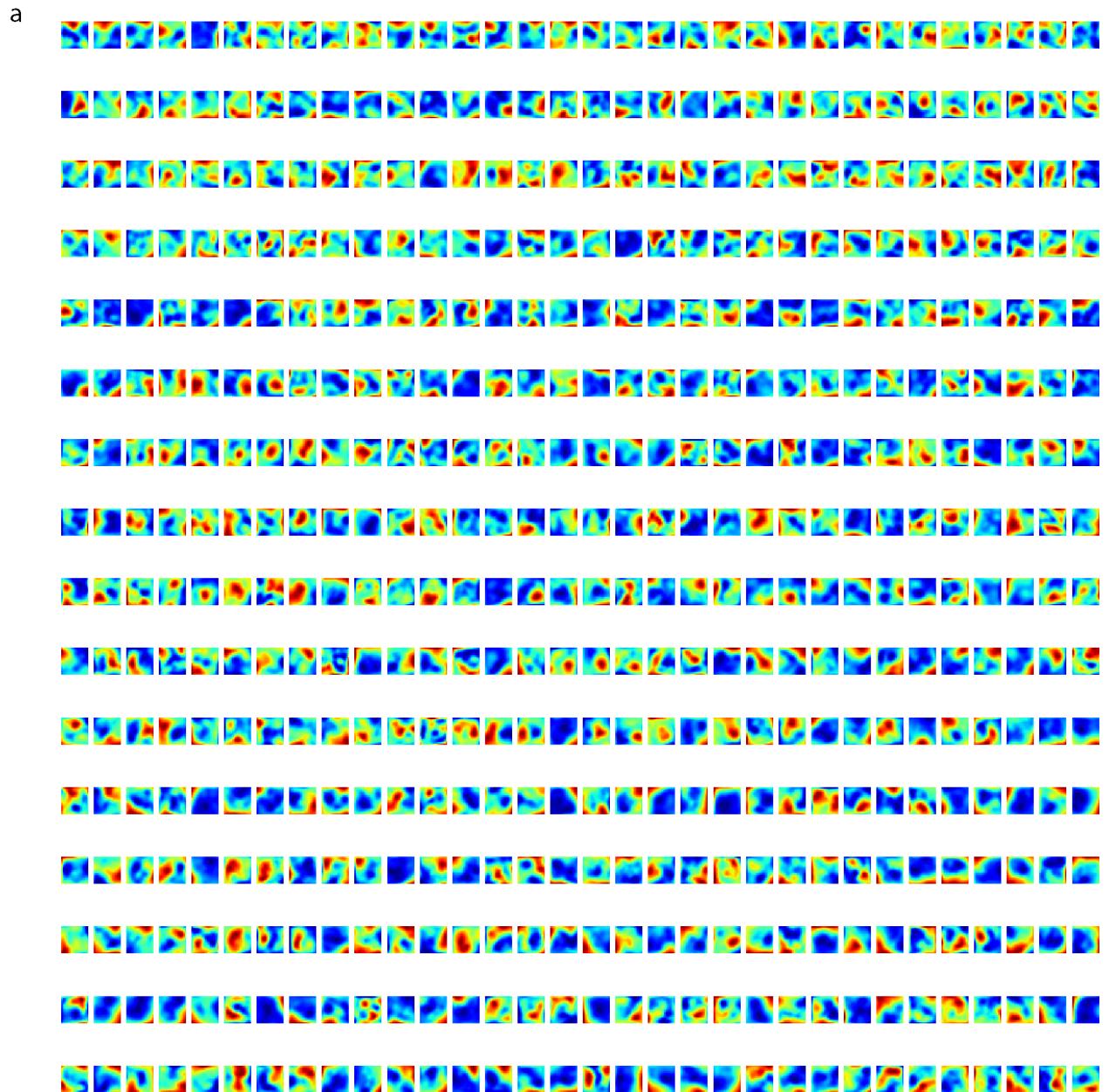


Extended Data Figure 2: **Linear layer spatial activity maps from the supervised learning experiment.** Spatial activity plots for all 512 units in the linear layer \vec{g}_t . Units exhibit spatial activity patterns resembling grid cells, border cells, and place cells — head direction tuning was also present but is not shown.



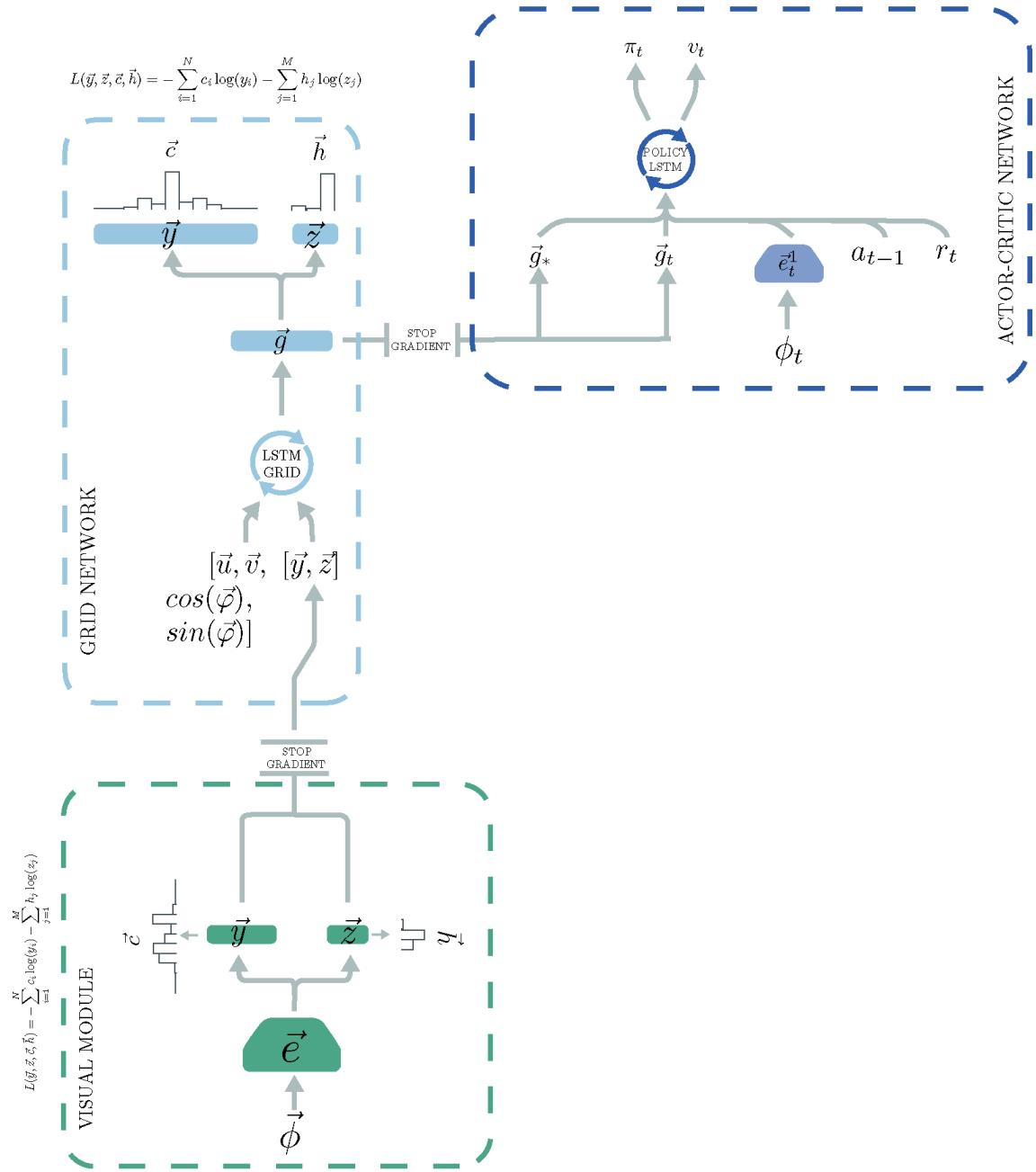
Extended Data Figure 3: **Characterization of grid-like units in Square environment and Circular environment.**

a) The scale (assessed from the spatial autocorrelogram of the ratemaps) of grid-like units exhibited a tendency to cluster at specific values. The number of distinct scale clusters was assessed by sequentially fitting Gaussian mixture models with 1 to 8 components. In each case, the efficiency of the fit (likelihood vs. number of parameters) was assessed using Bayesian information criterion (BIC). BIC was minimized with three Gaussian components indicating the presence of three distinct scale clusters. b) Spatial stability of units in the linear layer of the supervised network was assessed using spatial correlations — bin-wise Pearson product moment correlation between spatial activity maps (32 spatial bins in each map) generated at 2 different points in training, $t = 2e5$ and $t' = 3e5$ training steps. That is, $\frac{2}{3}$ of the way through training and the end of training, respectively. This separation was imposed to minimise the effect of temporal correlations and to provide a conservative test of stability. Grid-like units (gridness > 0.37) blue, directionally modulated units (resultant vector length > 0.47) green. Grid-like units exhibit high spatial stability, while directionally modulated units do not. c) Robustness of the grid representation to starting conditions. The network was retrained 100 times with the same hyperparameters but different random seeds controlling the initialisation of network weights, \vec{c} and \vec{h} . Populations of grid-like units (gridness > 0.37) were found to appear in all cases, the average proportion of grid-like units being 23% (SD of 2.8%). d) **circular environment:** the supervised network was also trained in a circular environment (diameter = 2.2m). As before, units in the linear layer exhibited spatially tuned responses resembling grid, border, and head direction cells. Eight units are shown. Top, ratemap displaying activity binned over location. Middle, spatial autocorrelogram of the ratemap, gridness²⁰ is indicated above. Bottom, polar plot of activity binned over head direction. e) Spatial scale of grid-like units ($n = 56$ (21.9%)) is clustered. Distribution is best fit by a mixture of 2 Gaussians (centres 0.58 & 0.96m, ratio = 1.66). f) Distribution of directional tuning for 31 most directionally active units, single line for each unit indicates length and orientation of resultant vector⁴⁶ g) Distribution of gridness and directional tuning. Dashed lines indicate 95% confidence interval derived from shuffling procedure (500 permutations), 5 grid units (9%) exhibit significant directional modulation.

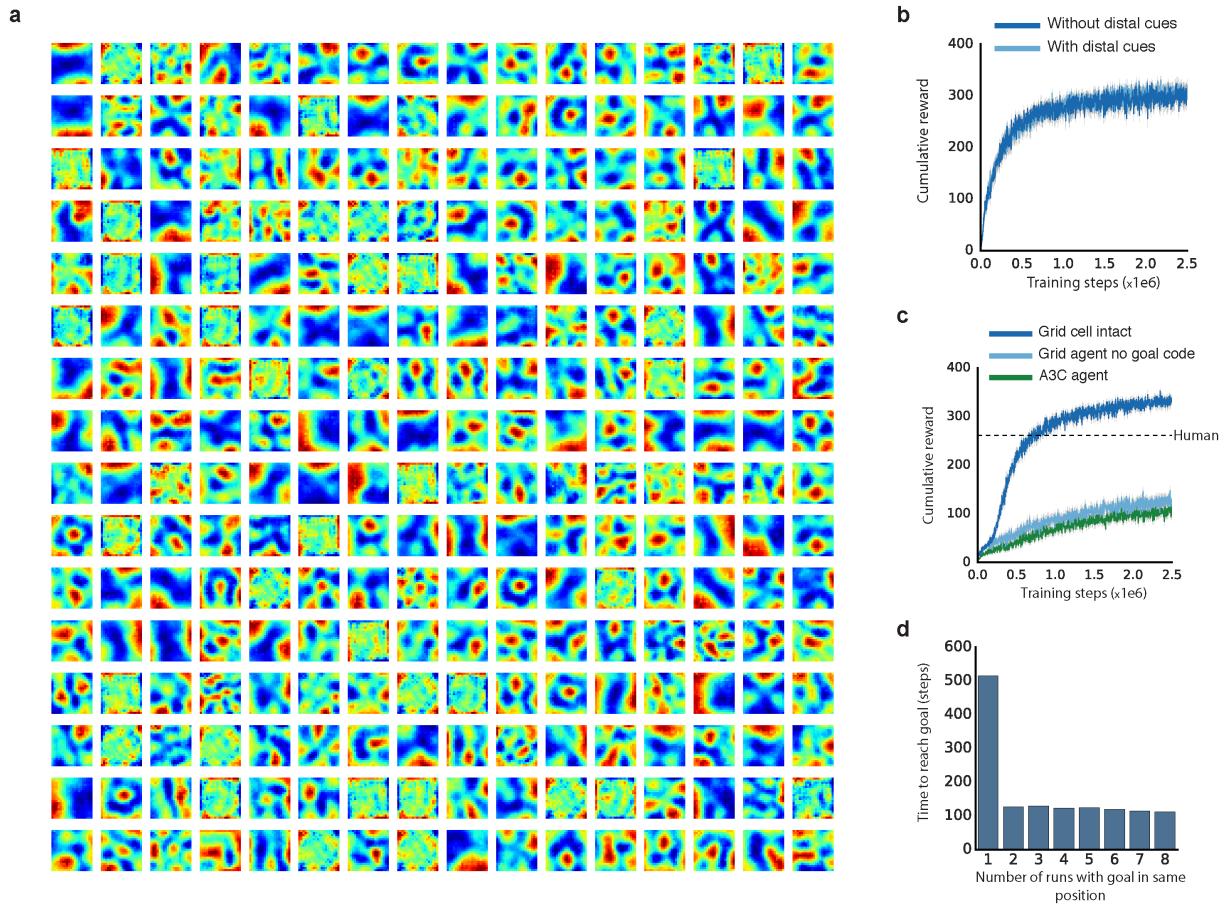


Extended Data Figure 4: **Grid-like units did not emerge in the linear layer when dropout was not applied.** Linear layer spatial activity maps ($n=512$) generated from a supervised network trained without dropout. The maps do not exhibit the regular periodic structure diagnostic of grid cells.

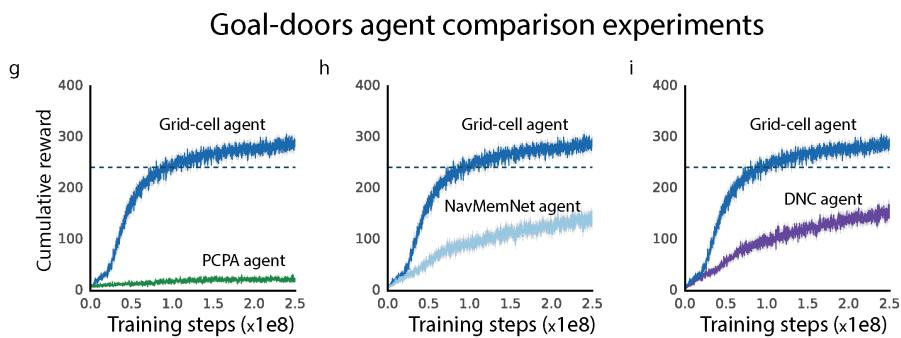
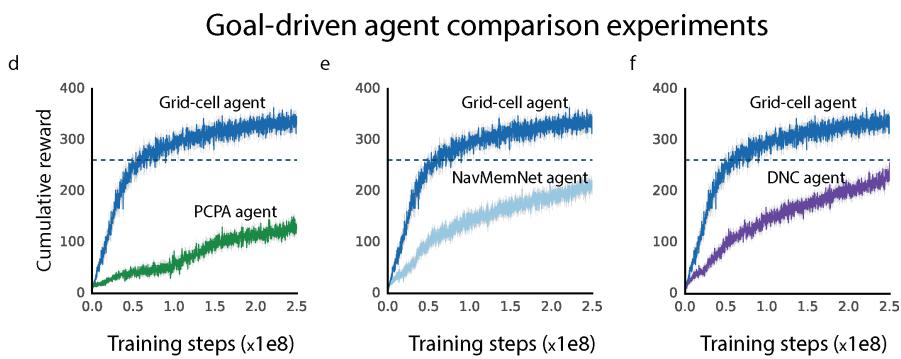
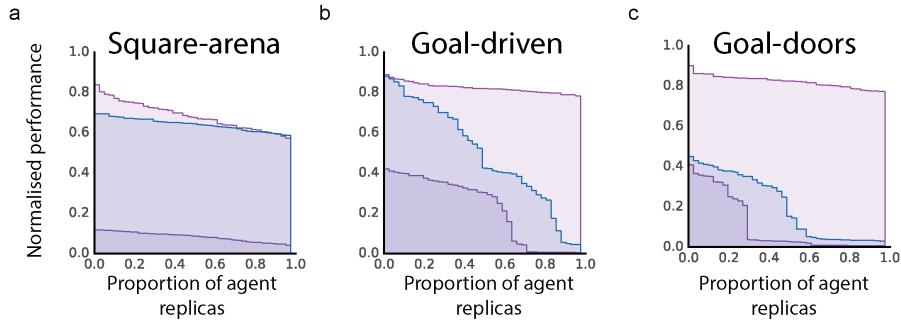
$$\mathcal{L}_{\text{A3C}} \approx \mathcal{L}_{\text{VR}} + \mathcal{L}_{\pi} - \mathbb{E}_{\theta \sim \pi} [\alpha H(\pi(s, \cdot, \theta))], \text{ where } \mathcal{L}_{\text{VR}} = \mathbb{E}_{\theta \sim \pi} [(R_{t, t+n} + \gamma^n V(s_{t+n+1}, \theta^-) - V(s_t, \theta))^2].$$



Extended Data Figure 5: **Architecture of the grid cell agent.** The architecture of the supervised network (grid network, light blue dashed) was incorporated into a larger deep RL network, including a visual module (green dashed) and an actor critic learner (based on A3C⁴⁰; dark blue dashed). In this case the supervised learner does not receive the ground truth \vec{c}_0 and \vec{h}_0 to signal its initial position, but uses input from the visual module to self-localize after placement at a random position within the environment. Visual module: since experimental evidence suggests that place cell input to grid cells functions to correct for drift and anchor grids to environmental cues^{21,27}, visual input was processed by a convolutional network to produce place cell (and head direction cell) activity patterns which were used as input to the grid network. The output of the vision module was only provided 5% of the time to the grid network; see Methods for implementational details), akin to occasional observations made by behaving animals of salient environmental cues²⁷. The output of the vision module was concatenated with \vec{u} , \vec{v} , $\sin\dot{\varphi}$, $\cos\dot{\varphi}$ to form the input to the GRID LSTM, which is the same network as in the supervised case (see Methods and Extended Data Figure 1). The actor critic learner (light blue dashed) receives as input the concatenation of \vec{e}_t^1 produced by a convolutional network with the reward r_t , the previous action $a_t - 1$, the linear layer activations of the grid cell network \vec{g}_t (“current grid-code”), and the linear layer activations observed last time the goal was reached, \vec{g}_* . \vec{g}_* (“goal grid-code”), which is set to zeros if the goal has not been reached in the episode. The fully connected layer was followed by an LSTM with 256 units. The LSTM has 2 different outputs. The first output, the actor, is a linear layer with 6 units followed by a softmax activation function, that represents a categorical distribution over the agent’s next action $\vec{\pi}_t$. The second output, the critic, is a single linear unit that estimates the value function v_t .



Extended Data Figure 6: **Characterisation of grid-like representations and robustness of performance for the grid cell agent in the square “land maze” environment.** a) Spatial activity plots for the 256 linear layer units in the agent exhibit spatial patterns similar to grid, border, and place cells. b) Cumulative reward indexing goal visits per episode (goal = 10 points) when distal cues are removed (dark blue) and when distal cues are present (light blue) — performance is unaffected, hence dark blue largely obscures light blue. Average of 50% best agent replicas (n=32) plotted (see Methods). The gray band displays the 68% confidence interval based on 5000 bootstrapped samples. c) Cumulative reward per episode when no goal code was provided (light blue) and when goal code was provided (dark blue). When no goal code was provided the agent performance fell to that of the baseline deep RL agent (A3C) (100 episodes average score ”no goal code” = 123.22 vs. A3C = 112.06 ,effect size = 0.21, 95% CI [0.18, 0.28]). Average of 50% best agent replicas (n=32) plotted (see Methods). The gray band displays the 68% confidence interval based on 5000 bootstrapped samples. d) After locating the goal for the first time during an episode the agent typically returned directly to it from each new starting position, showing decreased latencies for subsequent visits, paralleling the behaviour exhibited by rodents.

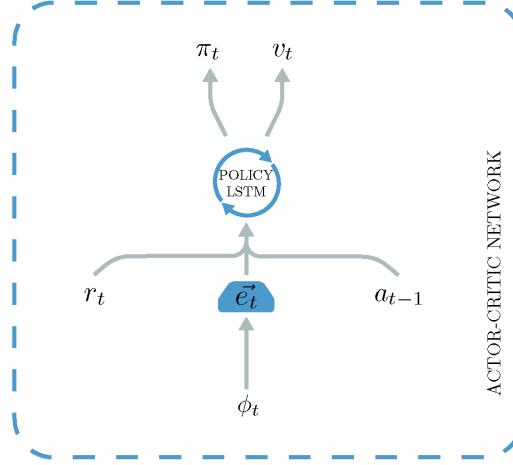


Extended Data Figure 7: **Robustness of grid cell agent and performance of other agents.** a-c) AUC performance gives the robustness to hyperparameters (i.e. learning rate, baseline cost, entropy cost - see Table 2 in Supplementary Methods for details of the range) and seeds (see Methods). For each environment we run 60 agent replicas (see Methods). Light purple is the grid agent, blue is the place cell agent and dark purple is A3C. a) Square arena b) Goal-driven c) Goal Doors. In all cases the grid cell agent shows higher robustness to variations in hyperparameters and seeds. d-i) Performance of place cell prediction/NavMemNet/DNC agents (see Methods) against grid cell agent. Dark blue is the grid cell agent (Extended Data Figure 5), green is the place cell prediction agent (Extended Data Figure 9a), purple is the DNC agent, light blue is the NavMemNet agent (Extended Data Figure 9b). The gray band displays the 68% confidence interval based on 5000 bootstrapped samples. d-f) Performance in goal-driven. g-i) Performance in goal-doors. Note that the performance of the place cell agent (Extended Data Figure 8b, lower panel) is shown in Figure 3.

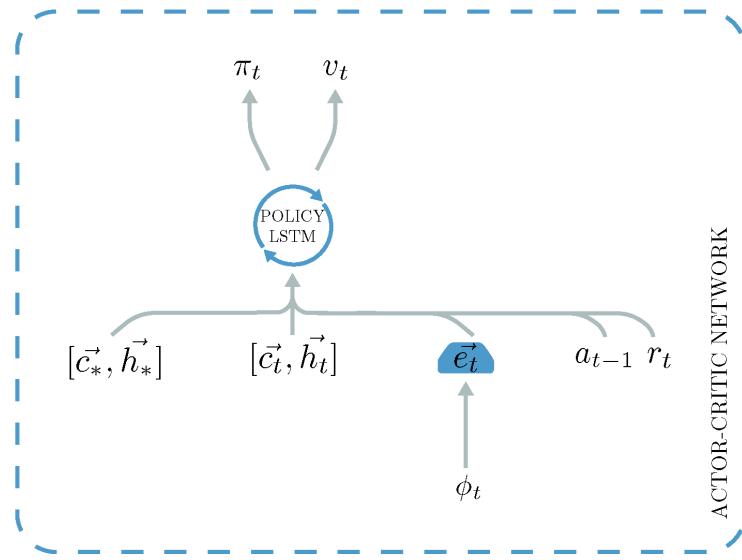
611

a

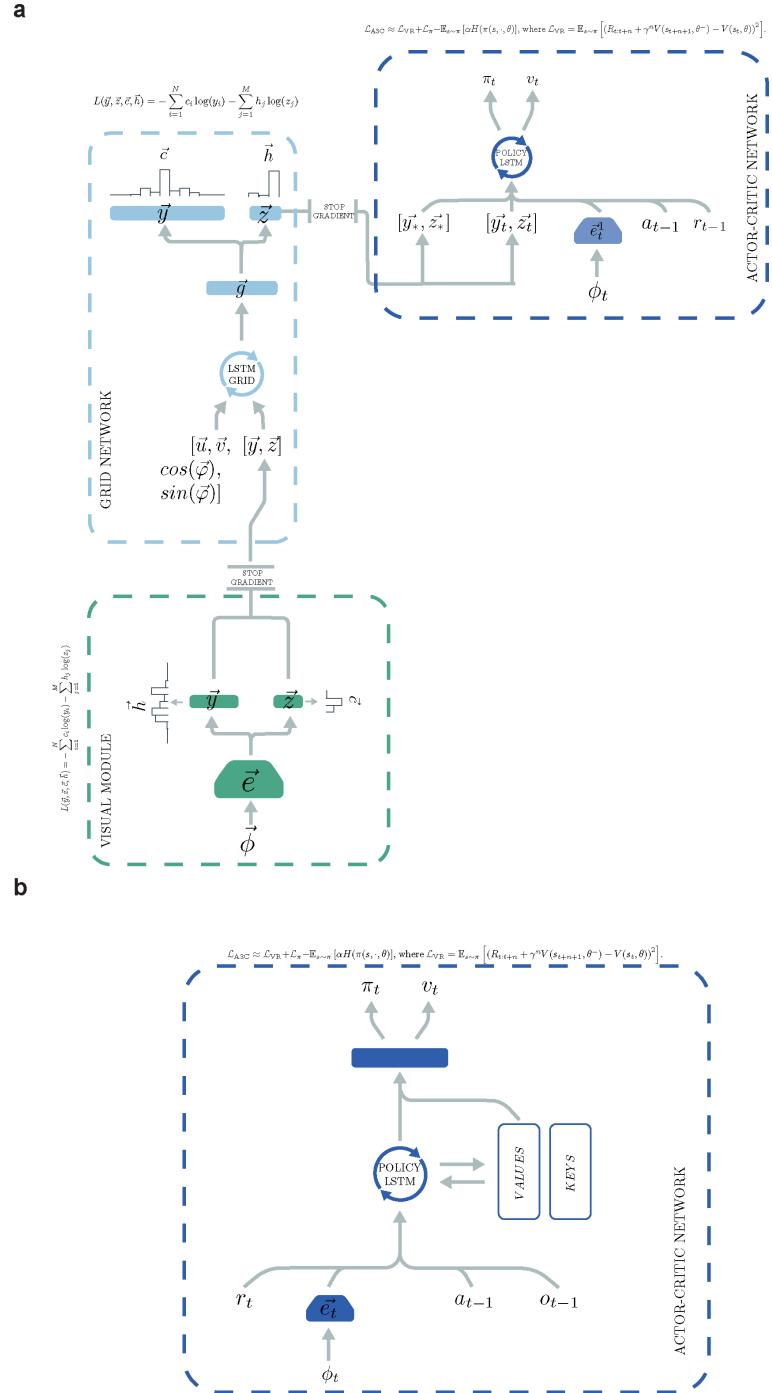
$$\mathcal{L}_{\text{A3C}} \approx \mathcal{L}_{\text{VR}} + \mathcal{L}_{\pi} - \mathbb{E}_{s \sim \pi} [\alpha H(\pi(s, \cdot, \theta))], \text{ where } \mathcal{L}_{\text{VR}} = \mathbb{E}_{s \sim \pi} [(R_{t:t+n} + \gamma^n V(s_{t+n+1}, \theta^-) - V(s_t, \theta))^2].$$

**b**

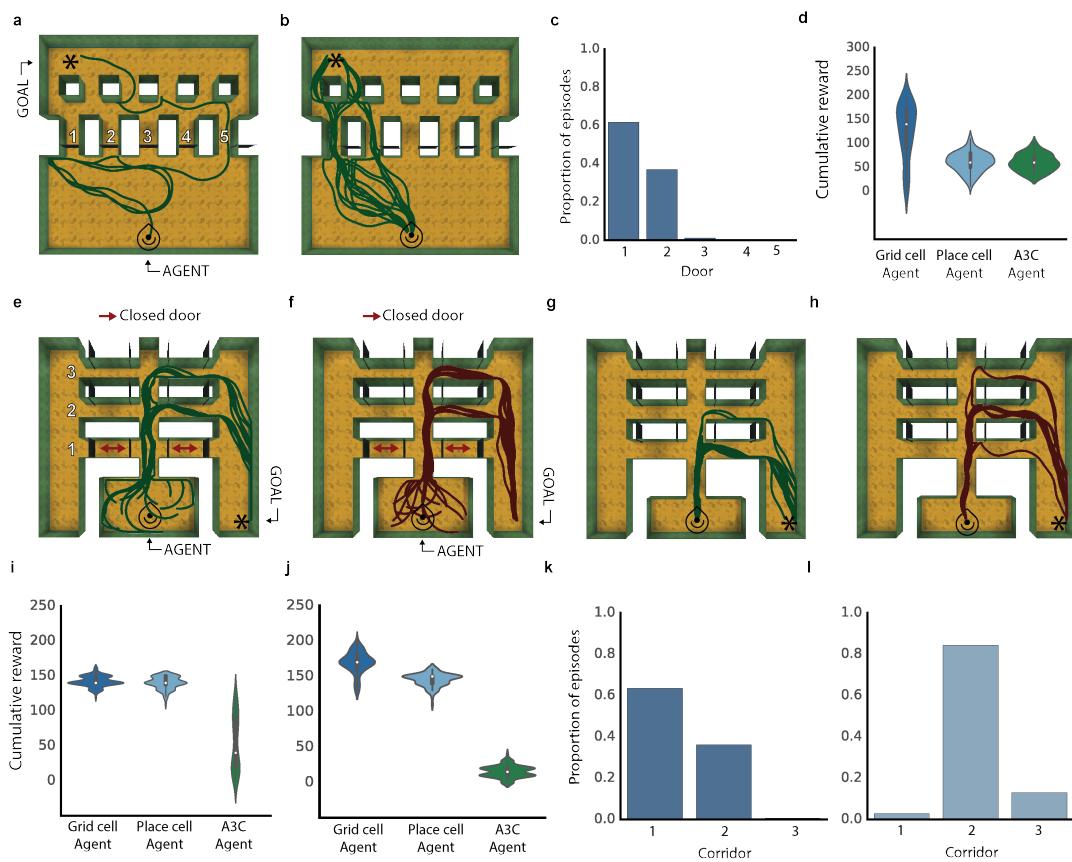
$$\mathcal{L}_{\text{A3C}} \approx \mathcal{L}_{\text{VR}} + \mathcal{L}_{\pi} - \mathbb{E}_{s \sim \pi} [\alpha H(\pi(s, \cdot, \theta))], \text{ where } \mathcal{L}_{\text{VR}} = \mathbb{E}_{s \sim \pi} [(R_{t:t+n} + \gamma^n V(s_{t+n+1}, \theta^-) - V(s_t, \theta))^2].$$



Extended Data Figure 8: **Architecture of the A3C and place cell agent.** a) The A3C implementation is as described in⁴⁰. b) The place cell agent was provided with the ground-truth place, \vec{c}_t , and head-direction, \vec{h}_t , cell activations (as described above) at each time step. The output of the fully connected layer of the convolutional network \vec{e}_t was concatenated with the reward r_t , the previous action $a_t - 1$, the ground-truth current place code, \vec{c}_t , and current head-direction code, \vec{h}_t — together with the ground truth goal place code, \vec{c}_* , and ground truth head direction code, \vec{h}_* ,
612 observed the last time the agent reached the goal (see Methods).



Extended Data Figure 9: **Architecture of the place cell prediction agent and of the NavMem-Net agent.** a) The architecture of the place cell prediction agent is similar to the grid cell agent — having a grid cell network with the same parameters as that of the grid cell agent. The key difference is the nature of the input provided to the policy LSTM. Instead of using grid codes from the linear layer of the grid network \vec{g} , we used the predicted place cell population activity vector \vec{y} and the predicted head direction population activity vector \vec{z} (i.e. the activations present on the output place and head direction unit layers of the grid cell network, corresponding to the current and goal position) as input for the policy LSTM. As in the grid cell agent, the output of the fully connected layer of the convolutional network, \vec{e}_t , the reward r_t , and the previous action a_{t-1} , were also input to the policy LSTM. The convolutional network had the same architecture described for the grid cell agent. b) NavMemNet agent. The architecture implemented is the one described in³, specifically FRMQN but the Asynchronous Advantage Actor-Critic (A3C) algorithm was used in place of Q-learning. The convolutional network had the same architecture described for the grid cell agent and the memory was formed of 2 banks (keys and values), each one composed of 1350 slots.



Extended Data Figure 10: **Flexible use of short-cuts** a) Overhead view of the linear sunburst maze in initial configuration, with only door 5 open. Example trajectory from grid cell agent during training (green line, icon indicates start location). b) Test configuration with all doors open: grid cell agent uses the newly available shortcuts (multiple episodes shown). c) Histogram showing proportion of times the agent uses each of the doors during 100 test episodes. The agent shows a clear preference for the shortest paths. d) Performance of grid cell agent and comparison agents during test episodes. e) Example grid cell agent and f) example place cell agent trajectory during training in the double E-maze (corridor 1 doors closed). g-h) in the test phase, with all doors open, the grid cell agent exploits the available shortcut (g), while the place cell agent does not (h). i-j) Performance of agents during training (i) and test (j). k-l, The proportion of times the grid (k) and place (l) cell agents use the doors on the 1st to 3rd corridor during test. The grid cell agent shows a clear preference for available shortcuts, while the place cell agent does not.

615 **Supplemental Information for *Vector-based Navigation using Grid-like Representations in***

616 ***Artificial Agents.***

617 Andrea Banino^{1*}, Caswell Barry^{2*}, Benigno Uria¹, Charles Blundell¹, Timothy Lillicrap¹, Piotr

618 Mirowski¹, Alexander Pritzel¹, Martin Chadwick¹, Thomas Degrif¹, Joseph Modayil¹, Greg

619 Wayne¹, Hubert Soyer¹, Fabio Viola¹, Brian Zhang¹, Ross Goroshin¹, Neil Rabinowitz¹, Razvan

620 Pascanu¹, Charlie Beattie¹, Stig Petersen¹, Amir Sadik¹, Stephen Gaffney¹, Helen King¹, Koray

621 Kavukcuoglu¹, Demis Hassabis¹, Raia Hadsell¹, Dharshan Kumaran¹

622 ¹DeepMind, 5 New Street Square, London EC4A 3TW, UK.

623 ²Department of Cell and Developmental Biology, University College London, London, UK

624 *equal contribution.

625 This section contains:

626 1. Supplementary Results

627 (a) Assessing path integration and goal-finding in a square arena

628 (b) Experimental manipulations to test the Vector-Based navigation hypothesis

629 (c) Comparison of grid cell agent with other agents in complex, procedurally-generated
630 multi-room environments

631 (d) Probe mazes assessing ability to take novel shortcuts

632 2. Supplementary Discussion

633 (a) Backpropagation through time (BPTT)

634 (b) Relationship to previous models of grid cells

635 3. Supplementary Methods

636 (a) Navigation through Deep RL

637 (b) Additional information about Agent Architectures

638 (c) Training algorithms

639 (d) Neuroscience-based analyses of units

640 (e) Multivariate decoding of representation of metric quantities within LSTM

641 (f) Statistical reporting

642 **1 - Supplementary Results for Vector-based Navigation using Grid-like Representations in Ar-**
643 ***tificial Agents.***

644 **1a - Assessing path integration and goal-finding in a square arena** To better understand the
645 advantage conveyed by a grid-like representation, we trained the agent to navigate to an unmarked
646 goal in a simple setting inspired by the classic Morris water maze (Fig. 2b&c; 2.5m×2.5m square
647 arena; see Methods). The agent was trained in episodes to ensure it was able to generalize to
648 arbitrary open field enclosures, each episode consisted of 5,400 steps — corresponding to ap-
649 proximately 90 s in total — after which the goal location, floor texture, and cue location were
650 randomized. An episode started with the agent in a random location, requiring it to first explore
651 in order to find an unmarked goal. Upon reaching the goal the agent was teleported to another
652 random location and continued to navigate with the aim of maximising the number of times it
653 reached the goal before the episode ended. In this setting self-localisation was more challenging.
654 Previously, in experiment described above, information about the ground truth initial location was
655 provided to initialise the LSTM, here the grid network learned to use visual information to de-
656 termine the agent's starting location and to correct for drift resulting from noise introduced to the
657 velocity inputs (see Methods). Despite these differences the grid network continued to self-localize
658 accurately, outputting place cell predictions consistent with the agent's location (Fig. 2e).

659 After locating the goal for the first time during an episode, the agent typically returned directly to it
660 from each new starting position, showing decreased latencies for subsequent visits (average score
661 for 100 episodes: grid cell agent = 289 vs place cell agent = 238, effect size = 1.80, 95% CI [1.63,
662 1.99], Fig. 2h, Extended Data Figure 6d). Performance of the grid cell agent was substantially

663 better than that of a control place cell agent with homogeneous place fields tuned to maximize
664 performance (see Supplemental Methods). Further, to additionally control for differences in the
665 number and area of spatial fields between agents, we also generated two place cell agents – incor-
666 porating 256 and 660 heterogeneously sized place fields – that were explicitly matched to the grid
667 cell agent (see Supplemental Methods for details). Again, the performance of the grid cell agent
668 was found to be considerably better than these additional place cell agents (Average score over 100
669 episodes: grid cell agent = 289 vs. best place agent with 660 heterogeneous fields = 212, effect
670 size = 3.93, 95% CI [3.54, 4.31]; best place agent with 256 heterogeneous fields = 225, effect size
671 = 3.52, 95% CI [3.18, 3.87]).

672 **1b - Experimental manipulations to test the Vector-Based navigation hypothesis** First, to
673 demonstrate that the goal grid code provided sufficient information to enable the agent to navigate
674 to an arbitrary location, we substituted it with a "fake" goal grid code sampled randomly from a
675 location in the environment (see Methods). The agent followed a direct path to the newly specified
676 location, circling the absent goal (Fig. 2i) — similar to rodents in probe trials of the Morris water
677 maze (escape platform removed). As a second test, we trained a grid cell agent without providing
678 the goal grid vector to the policy LSTM, effectively "lesioning" this code. Performance of the grid
679 agent drops to that of the baseline deep RL agent (A3C - a standard deep RL architecture, trained
680 without any grid or place cell input), confirming that the goal grid code is critical for vector based
681 navigation (see Extended Data Fig. 6c). Thirdly, to confirm the presence of a goal-directed vector,
682 we attempted to decode the scalar quantities composing the vector from the policy LSTM. Rea-
683 soning that the goal directed vector would be particularly important at the start of a trajectory, we

684 focused on the initial portion of navigation after the agent had reached the goal and was teleported
685 to a new location. We found that the policy LSTM of the grid cell agent contained representations
686 of two key components of vector-based navigation (Euclidean distance, and allocentric goal direc-
687 tion), and that both were more strongly present than in the place cell agent (Euclidean distance
688 difference in $r = 0.17$; 95% CI [0.11, 0.24]; Goal direction difference in $r = 0.22$; 95% CI [0.18,
689 0.26]; Figure 2j&k). Notably, a neural representation of goal distance has recently been reported
690 in mammalian hippocampus²⁹ (also see ⁴⁹). To determine the behavioral relevance of these two
691 metric codes, we examined the goal-homing accuracy in each episode over several steps immedi-
692 ately following the period of metric decoding. We found that variation in both Euclidean distance
693 ($r = 0.22$, 95% CI [-0.32, -0.09]) and allocentric goal direction ($r = 0.22$, 95% CI [-0.38, -0.15])
694 decoding error correlated with subsequent behavioral accuracy. This suggests that stronger metric
695 codes are indeed important for accurate goal-homing behavior.

696 Finally, to determine the specific contribution of the grid-like units, we made targeted lesions to the
697 goal grid code and reexamined performance and representation of the goal directed vector. When
698 25% of the most grid-like units were silenced (see Methods), performance was worse than lesion-
699 ing 25% at random (average score for 100 episodes: 126.1 vs. 152.5, respectively; effect size =
700 0.38, 95% CI [0.34, 0.42]). Further, as expected, goal-directed vector codes were more strongly
701 degraded (Euclidean distance: random lesions decoding accuracy $r = 0.45$, top-grid lesions de-
702 coding accuracy $r = 0.38$, difference in decoding accuracy = 0.08, 95% CI [0.03, 0.13]). We also
703 performed an additional experiment where the effect of the targeted grid lesion was compared to
704 that of lesioning non-grid units with patchy firing (see Supplemental Methods - section 3d for the

details of the procedure). Our results show that the targeted grid cell lesion had a greater effect than the patchy non-grid cell lesion (average score for 100 episodes: 126.1 vs. 151.7, respectively; effect size = 0.38, 95% CI [0.34, 0.42]). These results support a role for the grid-like units in vector-based navigation, with the relatively mild impact on performance potentially accounted for by the difference in lesioning networks as compared to animals. Specifically, the procedure for lesioning networks differs in important respects from experimental lesions in animals — which bears upon the results observed. Briefly, networks have to be trained in the presence of an incomplete goal grid code and thus have the opportunity to develop a degree of robustness to the lesioning procedure – which would otherwise likely result in a catastrophic performance drop (see Methods). This opportunity is not typically afforded to experimental animals. This, therefore, may explain the significant but relatively small performance deficit observed in lesioned networks.

1c - Comparison of grid cell agent with other agents in challenging, procedurally-generated multi-room environments Our comparison agents for the grid cell agent included an agent specifically designed to use a different representational scheme for space (i.e. place cell agent, see Extended Data Figure 8b and see Methods), and a baseline deep RL agent (A3C⁴⁰, see Extended Data Figure 8a). The place cell agent relates to theoretical models of goal-directed navigation from the neuroscience literature (e.g.^{41,42}). A key difference between grid and place cell based models is that the former are proposed to enable the computation of goal-directed vectors across large-scale spaces^{7,10,11} and⁵⁰, whereas place cell based models are inherently limited in terms of navigational range (i.e. to the largest place field) and do not support route planning across unexplored spaces¹¹. First, we test these three agents in the “goal-driven” maze (see Methods). The grid-cell agent ex-

726 hibited high levels of performance, and over the course of 100 episodes, attained an average score
727 of 346.5 (video: <https://youtu.be/BWqZwLQfwIM>), beating both the place cell agent (average
728 score 258.76; contrast effect size = 1.98, 95% CI [1.79, 2.18]) and the A3C agent (average score
729 137.00; contrast effect size = 14.31, 95% CI [12.91, 15.71]). The grid cell agent showed markedly
730 superior performance compared to the other agents in the “goal-doors” maze (average score over
731 100 episodes: grid cell agent = 284.30 vs place cell agent = 90.53, effect size = 7.86, 95% CI
732 [7.09, 8.63]; A3C agent = 48.69, effect size = 7.73, 95% CI [6.97, 8.48]) (video of grid cell agent:
733 <https://youtu.be/BWqZwLQfwIM>). Interestingly, therefore, the enhanced performance of the grid
734 cell agent was particularly evident when it was necessary to recompute trajectories due to changes
735 in the door configuration, highlighting the flexibility of vector-based navigation in exploiting ad
736 hoc short-cuts (Fig. 3f).

737 The grid cell agent exhibited stronger performance than a professional human player in both “goal-
738 driven” (average score: grid cell agent = 346.50 vs. professional human player = 261, effect size
739 = 4.00, 95% CI [3.50, 4.52]) and “goal-doors” (average score: grid cell agent = 284.30 vs. profes-
740 sional human player = 240.5, effect size = 2.49, 95% CI [2.18, 2.81]). The human expert received
741 10 episodes worth of training in each environment before undergoing 20 episodes of testing. This
742 is considerably less training than that experienced by the network. Importantly, however, the mam-
743 malian brain has evolved to path integrate and naturally the human expert had a lifetimes worth of
744 relevant navigational experience. Hence, although directly drawing concrete conclusions from rel-
745 ative performance of human and agents is necessarily difficult, providing human-level performance
746 is useful as a broad comparison and represents a commonly used benchmark in similar papers⁴⁴.

⁷⁴⁷ We also tested the ability of agents trained on the standard environment (11×11) to generalise to
⁷⁴⁸ larger environments (11×17 , corresponding to 2.7×4.25 meters) (see Methods). The grid cell
⁷⁴⁹ agent exhibited strong generalisation performance compared to the control agents (average score
⁷⁵⁰ over 100 episodes grid cell agent = 366.5 vs place cell agent = 175.7, effect size = 4.60, 95% CI
⁷⁵¹ [4.16, 5.06]; A3C agent = 219.4, effect size = 3.78, 95% CI [3.41, 4.15]).

⁷⁵² We assessed the performance of two deep RL agents with external memory^{3, 43} (see Extended Data
⁷⁵³ Figure 9b). Whilst these agents were trained purely using RL — that is, they did not utilize super-
⁷⁵⁴ vised learning implemented by the grid cell agent — their relatively poor performance illustrates
⁷⁵⁵ the challenge posed by the environments used (i.e. goal-driven and goal-doors) and shows that is
⁷⁵⁶ not readily solved by the use of external memory alone. Importantly, this also serves to highlight
⁷⁵⁷ the substantial advantage afforded to agents that can exploit vector-based mechanisms grounded
⁷⁵⁸ in a grid-cell based Euclidean framework of space — and the potential for future work to examine
⁷⁵⁹ the combination of such navigational strategies with more memory-intensive approaches. We also
⁷⁶⁰ compare the grid cell agent with a variation of the place cell agent which used the predicted place
⁷⁶¹ cell and head direction cell as input to the Policy LSTM instead of the ground truth information
⁷⁶² (see Extended Data Figure 9a and Supplementary Methods). This agent exhibited substantially
⁷⁶³ poorer performance than the grid agent.

⁷⁶⁴ Further, decoding accuracy was substantially and significantly higher in the grid cell agent than
⁷⁶⁵ both the place cell (Euclidean distance difference in $r = 0.44$; 95% CI [0.37, 0.51]; Goal direction
⁷⁶⁶ difference in $r = 0.52$; 95% CI [0.49, 0.56]) and deep RL (Euclidean distance difference in $r =$
⁷⁶⁷ 0.57; 95% CI [0.5, 0.63]; Goal direction difference in $r = 0.66$; 95% CI [0.62, 0.70]) control agents

768 (Figure 3j&k).

769 **1d - Probe mazes assessing ability to take novel shortcuts** A core feature of mammalian spatial
770 behaviour is the ability to exploit novel shortcuts and traverse unvisited portions of space⁹, a capac-
771 ity thought to depend on vector-based navigation^{9,11}. To assess this, we examined the ability of the
772 grid cell agent and comparison agents to use novel shortcuts when they became available in specif-
773 ically configured probe mazes (see Methods for details). First, agents trained in the goal-doors
774 environment were exposed to a linearized version of Tolman’s sunburst maze. The grid cell agent,
775 but not comparison agents, was reliably able to exploit shortcuts, preferentially passing through
776 the doorways that offered a direct route towards the goal (Fig. 4a-c, and Extended Data Figure 10).
777 The average testing score of the grid cell agent was higher than that of the place agent (124.1 vs
778 60.9, effect size = 1.46, 95% CI [1.32, 1.61]) and of the A3C agent (124.1 vs. 59.7, effect size =
779 1.51, 95% CI [1.36, 1.66]).

780 Next, to test the agents’ abilities to traverse a previously unvisited section of an environment,
781 we employed the “double-E shortcut” maze (Fig. 4d-f, and Extended Data Figure 10e-l). During
782 training, the corridor presenting the shortest route to the goal was closed at both ends, preventing
783 access or observation of the interior. In this simple configuration the grid and place cell agents
784 performed similarly, exceeding the RL control agent (Extended Data Figure 10i). However, at test,
785 when the doors were opened, the grid cell agent was able to exploit the short-cut corridor, whereas
786 the control agents continued to follow the longer route they had previously learnt (Extended Data
787 Figure 10j-l). In the “double-E shortcut” maze performance does not significantly differ between
788 the grid and place cell agents, but both are significantly better than the A3C control (grid cell

789 agent vs. place cell agent, effect size = 0.27, 95% CI [0.24, 0.29]; grid cell agent vs. A3C agent,
790 effect size = 2.99, 95% CI [2.69, 3.29]; place cell agent vs. A3C agent, effect size = 2.92, 95%
791 CI [2.63, 3.21]). When shortcuts become available in the test phase, the grid cell agent performs
792 significantly better than the place agent (grid cell agent vs. place cell agent, effect size = 1.89, 95%
793 CI [1.69, 2.09]; grid cell agent vs. A3C agent, effect size = 12.77, 95% CI [11.48, 14.07]; place
794 cell agent vs. A3C agent, effect size = 14.87, 95% CI [13.35, 16.38]).

795 **2 - Supplementary Discussion for *Vector-based Navigation using Grid-like Representations in***
796 ***Artificial Agents.***

797 **2a - Backpropagation through time (BPTT)** Whilst backpropagation provides a powerful mech-
798 anism for adjusting the weights within hierarchical networks analogous to those found in the brain
799 (e.g. the ventral visual stream), it has long been thought to be biologically implausible for several
800 reasons: for example, it requires access to information that is non-local to a synapse (i.e. informa-
801 tion about errors many layers downstream). However, recent research in several directions have
802 provided fresh new insights into how a process akin to backpropagation may be implemented in
803 the brain⁵¹. Whilst less research has been conducted into how BPTT could be implemented in the
804 brain, recent work points to potentially promising avenues that deserve further exploration⁵².

805 **2b - Relationship to previous models of grid cells** Our work contrasts with previous approaches
806 where grid cells have been hard-wired⁵³⁻⁵⁶ and⁵⁷, derived through eigendecomposition of place
807 fields^{58,59}, or arisen through self organization in the absence of an objective function⁶⁰. It is worth
808 noting that our experiments were not designed to provide insights into the development of grid

809 cells in the brain — due to the limitations of the training algorithm used (i.e. backpropagation) in
810 terms of biological plausibiliy (although see ⁶¹). More generally, however, our findings accord with
811 the perspective that the internal representations of individual brain regions such as the entorhinal
812 cortex arise as a consequence of optimizing for specific ethologically important objective functions
813 (e.g. path integration) — providing a parallel to the optimization process in neural networks⁶².

814 **3 - Supplementary Methods for Vector-based Navigation using Grid-like Representations in**
815 *Artificial Agents.*

816 **3a - Navigation through Deep RL**

817 **Probe mazes to test for shortcut behavior** The first maze used to test shortcut behaviour was a
818 linearized version of Tolman's sunburst maze⁶³ (Fig. 4a). The maze was used to determine if the
819 agent was able to follow an accurate heading towards the goal when a path became available. The
820 maze was size 13×13 and contained 5 evenly spaced corridors, each of which had a door at the
821 end closest to the start position of the agent. The agent always started on one side of the corridors
822 with the same heading orientation (North; see Fig 4a) and the goal was always placed in the same
823 location on the other side of the corridors. Until the agent reached the goal the first time only one
824 door was open (door 5, Fig. 4a), but after that all the doors were opened for the remainder of the
825 episode. After reaching the goal, the agent was teleported to the original position with the same
826 heading orientation. This maze was used to test the shortcut capabilities of agents that had been
827 previously trained in the "goal doors" environment. All the agents were tested in the maze for 100
828 episodes, each one lasting for a fixed duration of 5,400 environment steps (90 seconds).

829 The second maze, termed double E-maze, was designed to test the agents abilities to traverse
830 a previously unvisited section of an environment. The maze was size 12×13 and was formed
831 of 2 symmetric sides each one with 3 corridors. The goal location was always on the bottom
832 right or left, and the location was randomized over episodes. During training, the left and right
833 corridors closest to the bottom (i.e. those providing the shortest paths to the goals) were always

closed from both sides to avoid any exploration down these corridors (see Extended Data Figure 10e&f). This ensured any subsequent shortcut behavior had to traverse unexplored space. Of the remaining corridors, at any time, on each side only one was accessible (top or middle, randomly determined). Each time the agent reached the goal, the doors were randomly configured again (with the same constraints). The agent always started in a random location in the central room with a random orientation. At test time, after the agent reached the goal for the first time, all corridors were opened, allowing potential shortcut behavior (see Extended Data Figure 10g&h). During the test phase, the agent always started in the center of the central room facing north. Each agent was trained for $1e9$ environment step divided into episodes of 5,400 steps (90 seconds), and subsequently tested for 100 episodes, each one lasting for a fixed duration of 5,400 environment steps (90 seconds).

3b - Additional information about Agent Architectures

Details of vision module in the grid cell agent The convolutional neural network had four convolutional layers. The first convolutional layer had 16 filters of size 5×5 with stride 2 and padding 2. The second convolutional layer had 32 filters of size 5×5 with stride 2 and padding 2. The third convolutional layer had 64 filters of size 5×5 with stride 2 and padding 2. Finally, the fourth convolutional layer with 128 filters of size 5×5 with stride 2 and padding 2. All convolutional hidden layers were followed by a rectifier nonlinearity. The last convolution was followed by a fully connected layer with 256 hidden units. The same convolutional neural network was used for the actor-critic learner. The weights of the two network were not shared.

854 **Further details about the place cell agent** Place cell agent with homogeneously sized place
855 fields: we tested agents with fields — modelled as regular 2D Gaussians — having standard devi-
856 ations of 7.5cm, 25cm, and 75cm bins. The agent with fields of size 7.5cm was found to perform
857 best (highest cumulative reward on the Morris water maze task; see Supplemental Results) and
858 hence was chosen as the primary place cell control agent (see main text for score comparisons).

859 Place cell agent with heterogeneously sized place fields: to control for differences in the num-
860 ber and area of spatial fields between agents, we also generated two further place cell agents that
861 were explicitly matched to the grid cell agent. Specifically, we used a watershedding algorithm⁶⁴
862 to detect 660 individual grid fields in the grid-like units of the grid cell agent. The distribution
863 of the areas of these fields were found to exhibit 3 peaks — based on a Gaussian fitting proce-
864 dure — having means equivalent to 2D Gaussians with standard deviations of 8.2cm, 15.0cm, and
865 21.7cm. Hence we generated a further control agent having 395 place cells of size 8.2cm, 198
866 of size 15.0cm, and 67 of 21.7cm — 660 place cells in total, the relative numbers reflecting the
867 magnitudes of the Gaussians fit to the distribution. A final control agent was also generated having
868 256 place cell units in total — the same number of linear layer units as the grid agent — distributed
869 across the same three scales in a similar ratio. Additionally, we note that from a machine learn-
870 ing perspective, the place cell and grid cell agents with the same number of linear layer units are
871 in principle well matched since they are provided with the same input information and have an
872 identical number of parameters.

873 **Place cell prediction agent.** The architecture of the place cell prediction agent (Extended Data
874 Figure 9a) is similar to the grid cell agent described in the Methods : the key difference is the

875 nature of the input provided to the policy LSTM as described below. Specifically, the output of the
876 fully connected layer of the convolutional network, \vec{e}_t , was concatenated with the reward r_t , the
877 previous action $a_t - 1$, the current predicted place cell activity vector, \vec{y}_t , and the current predicted
878 head direction cell activity vector \vec{h}_t — and the goal predicted place cell activity vector, \vec{y}_* , and
879 goal head direction activity vector, \vec{h}_* , observed the last time the agent had reached the goal — or
880 zeros if the agent had not yet reached the goal within the episode. The convolutional network had
881 the same architecture described for the grid cell agent.

882 **3c - Training algorithms**

883 We assume the standard reinforcement learning setting where an agent interacts with an environ-
884 ment over a number of discrete time steps. As previously defined the at time t the agent receives
885 an observation o_t along with a reward r_t and produces an action a_t . The agent's state s_t is a func-
886 tion of its experience up until time t , $s_t = f(o_1, r_1, a_1, \dots, o_t, r_t)$ (The specifics of o_t are defined
887 in the architecture section). The n -step return $R_{t:t+n}$ at time t is defined as the discounted sum of
888 rewards, $\hat{R}_t = \sum_{i=0 \dots n-1} \gamma^i r_{t+i} + \gamma^n V(s_{t+n}, \theta)$. The value function is the expected return from
889 state s , $V^\pi(s) = \mathbb{E}[R_{t:\infty} | s_t = s, \pi]$, under actions selected accorded to a policy $\pi(a|s)$. See main
890 methods for the details of the loss functions.

891 **3d - Neuroscience-based analyses of units**

892 **Gridness score and grid scale calculation** Following ²⁰ and ¹⁸ spatial autocorrelograms of
893 ratemaps were used to assess the gridness and grid scale of linear layer units. First, for each unit,
894 the spatial autocorrelogram was calculated as defined in ²⁰. To calculate gridness²⁰, a measure

895 of hexagonal periodicity, we followed the 'expanding gridness' method introduced by ¹⁸. Briefly,
 896 a circular annulus centred on the origin of the autocorrelogram was defined, having radius of 8
 897 bins and with the central peak excluded. The annulus was rotated in 30° increments and, at each
 898 increment, the Pearson product moment correlation coefficient with the unrotated version of itself
 899 found. An interim gridness value was then defined as the highest correlation obtained from ro-
 900 tations of 30, 90 and 150° subtracted from the lowest at 0, 60 and 120°. This process was then
 901 repeated, each time expanding the annulus by 2, up to a maximum of 20. Finally, the gridness value
 902 was taken as the highest interim score.

903 Grid scale²⁰, a simple measure of the wavelength of spatial periodicity, was defined from the
 904 autocorrelogram as follows. The six local maxima closest to but excluding the central peak were
 905 identified. Grid scale was then calculated as the median distance of these peaks from the origin.

906 **Directional measures** Following⁴⁶ the degree of directional modulation exhibited by each unit
 907 was assessed using the length of the resultant vector of the directional activity map. Vectors corre-
 908 sponding to each bin of the activity map were created:

$$r_i = \begin{bmatrix} \beta_i \cos \alpha_i \\ \beta_i \sin \alpha_i \end{bmatrix}, \quad (6)$$

where α and β are, respectively, the centre and intensity of angular bin i in the activity map. These vectors were averaged to generate a mean resultant vector:

$$\vec{r} = \frac{\sum_{n=1}^N r_i}{\sum_{n=1}^N \beta_i}, \quad (7)$$

909 and the length of the resultant vector calculated as the magnitude of \vec{r} . We used 20 angular bins.

Border score To identify units that were preferentially active adjacent to the edges of the enclosure we adopted a modified version of the border score⁴⁷. For each of the four walls in the square enclosure, the average activation for that wall, b_i , was compared to the average centre activity c obtaining a border score for that wall, and the maximum was used as the border-score for the unit:

$$b_s = \max_{i \in \{1,2,3,4\}} \frac{b_i - c}{b_i + c} \quad (8)$$

where b_i is the mean activation for bins within d_b distance from the i -th wall and c the average activity for bins further than d_b bins from any wall. In all our experiments 20 by 20 bins where used and d_b took value 3.

Threshold setting for gridness, border score, and directional measures The hexagonality of the spatial activity map (gridness), directional modulation (length of resultant vector), and propensity to be active against environmental boundaries (border scale) exhibited by units in the linear layer were benchmarked against null distributions obtained using permutation procedures⁶⁵⁴⁸.

For the gridness measure and border score, null distributions were constructed using a 'field shuffle' procedure equivalent to that specified by⁴⁸. Briefly, a watershedding algorithm⁶⁴ was applied to the ratemap to segment spatial fields. The peak bin of each field was found and allocated to a random position within the ratemap. Bins around each peak were then incrementally replaced, retaining as far as possible their proximity to the peak bin. This procedure was repeated 100 times for each of the units present in the linear layer and the gridness and border score of the shuffled ratemaps assessed as before. In each case the 95th percentile of the resulting null distribution was found and used as a threshold to determine if that unit exhibited significant grid or border-like

925 activity.

926 To validate the thresholds obtained using shuffling procedures we calculated alternative null
927 distributions by analysing the grid and border responses of linear units from 500 untrained net-
928 works. Again, in each case, a grid score and border score for each unit was calculated, these were
929 pooled, and the 95th percentile found. In all cases the thresholds obtained by the first method were
930 found to be most stringent and these were used for all subsequent analyses

931 To establish a significance threshold for directional modulation we calculated the length of
932 the resultant vector that would demonstrate statistically significance under a Rayleigh test of direc-
933 tional uniformity at $\alpha = 0.01$. The resultant vector was calculated by first calculating the average
934 activation for 20 directional bins. A threshold length of 0.47 for the resultant vector was obtained.
935 The most stringent of these two thresholds was used.

936 **Clustering of scale in grid-like units** To determine if grid-like units exhibited a tendency to
937 cluster around specific scales we applied two methods.

938 First, following²², to determine if the scales of grid-like units ($\text{gridness} > 0.37$, 129/512
939 units) followed a continuous or discrete distribution we calculated the 'discreteness measure'²²
940 of the distribution of their scales. Specifically, scales were binned into a histogram with 13 bins
941 distributed evenly across a range corresponding to scales 10 to 36 spatial bins. 'Discreteness'
942 was defined as the standard deviation of the counts in each bin. Again following²², statistical
943 significance for this value was obtained by comparing it to a null distribution generated from a
944 shuffled version of the same data. Specifically, shuffles were generated as follows: For each unit, a

random number was drawn from a flat distribution between -1/2 and +1/2 of the smallest grid scale in this case between -7 and +7 spatial bins. The random number was added to the grid scales, the population was binned as before, and the discreteness score calculated. This procedure was completed 500 times. The discreteness score of the real data was found to exceed that of all the 500 shuffles ($p < 0.002$).

Second, to characterise the number and location of scale clusters, the distribution of scales from grid-like units was fit with Gaussian mixture distributions containing 1 to 8 components. Fits were made using an Expectation-Maximization approach implemented with fitgmdist (Matlab 2016b, Mathworks, MA). The efficiency of fits made with different numbers of components was compared using Bayesian Information Criterion (BIC)⁶⁶ the model (3 components) with the lowest BIC score was selected as the most efficient.

Lesioning experiment: comparison of targeted grid unit lesion vs lesion of patchy non-grid units We lesioned a random subset of patchy multi-field spatial cells that were non-grid units (i.e. grid score lower than 0.37 threshold). The units chosen had a head-direction score lower than 0.47 and the number of spatial fields was in the same range as grid-like units (3 to 13). The number of fields in each ratemap was calculated by applying a watershedding algorithm⁶⁴ to their ratemap – ignoring fields with area smaller than 4 bins. This procedure identified 174 units as multi-field patchy spatial cells (out of 256 units in the linear layer). We then selected 64 random units from these 174 and we ran 100 episodes in which these units were silenced (see Supplemental Results section 1b). We also ran another variant of the experiment where we ran 100 episodes and in each episode we selected a different subset of 64 random units from the 174 identified by

966 the watershedding procedure, and these units were silenced. The results were not qualitatively
967 different from the former experiment (data not shown).

968 **3e - Multivariate decoding of representation of metric quantities within LSTM**

969 A key prediction of the vector-based navigation hypothesis is that grid codes should allow down-
970 stream regions to compute a set of metric codes relevant to accurate goal-directed navigation.
971 Specifically, Euclidean distance and allocentric direction to the goal should both be computed by
972 an agent using vector-based navigation (see Fig. 2j&k also 3i-k). To test whether the same rep-
973 resentations can be found in the grid cell agent, and thereby provide additional evidence that it is
974 indeed using a vector-based navigation strategy, we recorded the activity in the policy LSTM of
975 the grid cell agent while it navigated in the land-maze and goal-driven environments. For each en-
976 vironment and agent, we collected data from 200 separate episodes. In each episode, we recorded
977 data from the time period following the first time the agent reached the goal and was teleported to
978 a new location in the maze. After an initial period to allow self-localization (8 steps), we exam-
979 ined the representation of the metric quantities over the next 12 steps, where the LSTM activity
980 was sampled at 4 even points over those steps. We focussed on this time period because the agent
981 potentially has knowledge of the goal location, but has not yet been able to learn the optimal path
982 to the goal. Thus it is this initial period of time where the computation of the vector-based navi-
983 gation metrics should be most useful, as this allows accurate navigation right from the start of being
984 teleported to a new location. In the land maze task, we additionally collected the same data from a
985 place cell agent control, and the two lesioned grid cell agents. In the goal driven task, we collected
986 data from the place cell agent and A3C. For each agent, we applied a decoding analysis to the

987 LSTM dictating the policy and value function. We ran two separate decoding analyses, looking
988 for evidence of each of the two metric codes (i.e. Euclidean distance, allocentric goal direction).
989 For each decoding analysis we trained a L2-regularized (ridge) regression model on all data apart
990 from the first 21 time-steps of each episode. The model was then tested on the four early sampling
991 steps of interest, where accuracy was assessed as the Pearson correlation between the predicted
992 and actual values over the 200 episodes. The penalization parameter was selected by randomly
993 splitting the training data into internal training and validation sets (90% and 10% of the episodes
994 respectively). The optimal parameter was selected from 30 values, evenly spaced on a log scale
995 between 0.001 and 1000, based on the best performance on the validation set. This parameter was
996 then used to train the model on the full training set, and evaluated on the fully independent test set.
997 As the allocentric direction metric is circular, we decomposed the vector into two target variables:
998 the cosine and sine of the polar angle. All reported allocentric decoding results are the average
999 of the cosine and sine results. For the purpose of comparing decoding accuracy across agents,
1000 we report the difference in accuracy, along with a 95% bootstrapped confidence interval on this
1001 difference, based on 10,000 samples.

1002 **3f - Statistical reporting**

We followed the guidelines outlined by⁶⁷. Specifically reporting effect sizes and confidence intervals. Unless otherwise stated, the effect sizes are calculated using the following formula:

$$\text{effect size} = \frac{\mu_{group1} - \mu_{group2}}{\sigma_{pooled}}, \quad (9)$$

and the σ_{pooled} was calculated accordingly to⁶⁸ using:

$$\sigma_{pooled} = \sqrt{\frac{(N_{group1} - 1) \times \sigma_{group1}^2 + (N_{group2} - 1) \times \sigma_{group2}^2}{N_{group1} + N_{group2} - 2}} \quad (10)$$

The confidence interval for the effect size was calculated accordingly to⁶⁹ using:

$$ci_{effectsize} = \sqrt{\frac{N_{group1} + N_{group2}}{N_{group1} \times N_{group2}} + \frac{effect\ size^2}{2 \times (N_{group1} + N_{group2})}}. \quad (11)$$

Parameter name	Value	Description
T	15	Duration of simulated trajectories (seconds)
L	2.2	Width and height of environment, or diameter for circular environment (meters)
d	0.03	Perimeter region distance to walls (meters)
$\sigma^{(v)}$	0.13	Forward velocity Rayleigh distribution scale (m/sec)
$\mu^{(\phi)}$	0	Rotation velocity Gaussian distribution mean (deg/sec)
$\sigma^{(\phi)}$	330	Rotation velocity Gaussian distribution standard deviation (deg/sec)
ρ_{RH}	0.25	Velocity reduction factor when located in the perimeter
Δ_{RH}	90	Change in angle when located in the perimeter (deg)
Δt	0.02	Simulation-step time increment (seconds)
N	256	Number of place cells
$\sigma^{(c)}$	0.01	Place cell standard deviation parameter (meters)
M	12	Number of target head direction cells
$\kappa^{(h)}$	20	Head direction concentration parameter
g_c	10^{-5}	Gradient clipping threshold
minibatch size	10	Number of trajectories used in the calculation of a stochastic gradient
trajectory length	100	Number of time steps in the trajectories used for the supervised learning task
learning rate	10^{-5}	Step size multiplier in the RMSProp algorithm
momentum	0.9	Momentum parameter of the RMSProp algorithm
L2 regularisation	10^{-5}	Regularisation parameter for linear layer
parameter updates	300000	Total number of gradient descent steps taken

Table 1: Supervised learning hyperparameters.

Parameter name	Value	Description
Learning rate	[0.000001, 0.0002]	Step size multiplier in the shared RMSProp algorithm of the actor-critic learner with a break
Gradient momentum	0.99	Momentum parameter of the shared RMSProp algorithm
Baseline cost [α]	[0.48, 0.52]	Cost applied on the gradient of v
Entropy regularisation [β]	[0.00006, 0.0001]	Entropy regularization term with respect to the policy parameters
Discount	0.99	Discount factor gamma used in the value function estimation
Back-propagation step in the actor-critic learner	100	Number of backpropagation step used to unroll the LSTM
Action repeat	4	Repeat each action selected by the agent this many times
Learning rate grid network	0.001	Step size multiplier in the RMSProp algorithm of the supervised learner
$\sigma^{(c)}$	40	Place cell scale
M	12	Number of target head direction cells
$\kappa^{(h)}$	20	Head direction concentration parameter
Back-propagation step in the supervised learner	100	Number of time steps in the trajectories used for the supervised learning task
L2 regularization	0.0001	Regularization parameter for linear layers in bottleneck
Gradient momentum	0.9	Momentum parameter of the RMSProp algorithm in the supervised learner

Table 2: Hyperparameters of all the agents presented. Values in square bracket are sampled from a categorial distribution in that range

1003

- 1004 49. Chadwick, M. J., Jolly, A. E., Amos, D. P., Hassabis, D. & Spiers, H. J. A goal direction signal
1005 in the human entorhinal/subiculum region. *Current Biology* **25**, 87–92 (2015).
- 1006 50. Kubie, J. L. & Fenton, A. A. Linear look-ahead in conjunctive cells: an entorhinal mechanism
1007 for vector-based navigation. *Frontiers in neural circuits* **6**, 20 (2012).
- 1008 51. Scellier, B. & Bengio, Y. Towards a biologically plausible backprop. *arXiv preprint*
1009 *arXiv:1602.05179* **914** (2016).
- 1010 52. Ke, N. R. *et al.* Sparse attentive backtracking: Long-range credit assignment in recurrent
1011 networks. *arXiv preprint arXiv:1711.02326* (2017).
- 1012 53. Burgess, N., Barry, C. & O'keefe, J. An oscillatory interference model of grid cell firing.
1013 *Hippocampus* **17**, 801–812 (2007).
- 1014 54. Hasselmo, M. E., Giocomo, L. M. & Zilli, E. A. Grid cell firing may arise from interference
1015 of theta frequency membrane potential oscillations in single neurons. *Hippocampus* **17**, 1252–
1016 1271 (2007).
- 1017 55. Burak, Y. & Fiete, I. R. Accurate path integration in continuous attractor network models of
1018 grid cells. *PLoS Comput Biol* **5**, e1000291 (2009).
- 1019 56. Fuhs, M. C. & Touretzky, D. S. A spin glass model of path integration in rat medial entorhinal
1020 cortex. *Journal of Neuroscience* **26**, 4266–4276 (2006).

- 1021 57. Gustafson, N. J. & Daw, N. D. Grid cells, place cells, and geodesic generalization for spatial
1022 reinforcement learning. *PLoS Comput Biol* **7**, e1002235 (2011).
- 1023 58. Stachenfeld, K. L., Botvinick, M. & Gershman, S. J. Design principles of the hippocampal
1024 cognitive map. In *Advances in neural information processing systems*, 2528–2536 (2014).
- 1025 59. Dordek, Y., Soudry, D., Meir, R. & Derdikman, D. Extracting grid cell characteristics from
1026 place cell inputs using non-negative principal component analysis. *eLife* **5**, e10094 (2016).
- 1027 60. Widłoski, J. & Fiete, I. How does the brain solve the computational problems of spatial
1028 navigation? In *Space, Time and Memory in the Hippocampal Formation*, 373–407 (Springer,
1029 2014).
- 1030 61. Bengio, Y., Lee, D.-H., Bornschein, J., Mesnard, T. & Lin, Z. Towards biologically plausible
1031 deep learning. *arXiv preprint arXiv:1502.04156* (2015).
- 1032 62. Marblestone, A. H., Wayne, G. & Kording, K. P. Toward an integration of deep learning and
1033 neuroscience. *Frontiers in Computational Neuroscience* **10** (2016).
- 1034 63. Tolman, E. C. *et al.* Cognitive maps in rats and men (1948).
- 1035 64. Beucher, S. Use of watersheds in contour detection. In *Proceedings of the International
1036 Workshop on Image Processing* (CCETT, 1979).
- 1037 65. Yartsev, M. M., Witter, M. P. & Ulanovsky, N. Grid cells without theta oscillations in the
1038 entorhinal cortex of bats. *Nature* **479**, 103–107 (2011).
- 1039 66. Schwarz, G. Estimating the dimension of a model. *The Annals of Statistics* **6**, 461–464 (1978).

- 1040 67. Halsey, L. G., Curran-Everett, D., Vowler, S. L. & Drummond, G. B. The fickle p value
1041 generates irreproducible results. *Nature methods* **12**, 179 (2015).
- 1042 68. Olejnik, S. & Algina, J. Measures of effect size for comparative studies: Applications, inter-
1043 pretations, and limitations. *Contemporary educational psychology* **25**, 241–286 (2000).
- 1044 69. Hedges, L. & Olkin, I. *Statistical Methods for Meta-analysis* (Academic Press, 1985). URL
1045 <https://books.google.co.uk/books?id=brNpAAAAMAAJ>.