nature
SUPPLEMENTARY INFORMATION
https://doi.org/10.1038/s41586-018-0102-6
In the format provided by the authors and unedited.

Vector-based navigation using grid-like representations in artificial agents

Andrea Banino[1,2,3,5]*, Caswell Barry[2,5]*, Benigno Uria[1], Charles Blundell[1], Timothy Lillicrap[1], Piotr Mirowski[1], Alexander Pritzel[1], Martin J. Chadwick[1], Thomas Degris[1], Joseph Modayil[1], Greg Wayne[1], Hubert Soyer[1], Fabio Viola[1], Brian Zhang[1], Ross Goroshin[1], Neil Rabinowitz[1], Razvan Pascanu[1], Charlie Beattie[1], Stig Petersen[1], Amir Sadik[1], Stephen Gaffney[1], Helen King[1], Koray Kavukcuoglu[1], Demis Hassabis[1,4], Raia Hadsell[1] & Dharshan Kumaran[1,3]*

[1]DeepMind, London, UK. [2]Department of Cell and Developmental Biology, University College London, London, UK. [3]Centre for Computation, Mathematics and Physics in the Life Sciences and Experimental Biology (CoMPLEX), University College London, London, UK. [4]Gatsby Computational Neuroscience Unit, University College London, London, UK. [5]These authors contributed equally: Andrea Banino, Caswell Barry. *e-mail: abanino@google.com; caswell.barry@ucl.ac.uk; dkumaran@google.com

Supplemental Information for *Vector-based Navigation using Grid-like Representations in Artificial Agents.*

Andrea Banino[1,2,3*], Caswell Barry[2*], Benigno Uria[1], Charles Blundell[1], Timothy Lillicrap[1], Piotr Mirowski[1], Alexander Pritzel[1], Martin J. Chadwick[1], Thomas Degris[1], Joseph Modayil[1], Greg Wayne[1], Hubert Soyer[1], Fabio Viola[1], Brian Zhang[1], Ross Goroshin[1], Neil Rabinowitz[1], Razvan Pascanu[1], Charlie Beattie[1], Stig Petersen[1], Amir Sadik[1], Stephen Gaffney[1], Helen King[1], Koray Kavukcuoglu[1], Demis Hassabis[1,4], Raia Hadsell[1], Dharshan Kumaran[1,3]

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

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

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

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

[*]equal contribution.

This section contains:

1 - Supplementary Results for *Vector-based Navigation using Grid-like Representations in Artificial Agents.*

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

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 (average score for 100 episodes: grid cell agent = 289 vs place cell agent = 238, effect size = 1.80, 95% CI [1.63, 1.99], Fig. 2h, Extended Data Figure 6d). Performance of the grid cell agent was substantially better than that of a control place cell agent with homogeneous place fields tuned to maximize performance (see Supplemental Methods). Further, to additionally control for differences in the number and area of spatial fields between agents, we also generated two place cell agents – incorporating 256 and 660 heterogeneously sized place fields – that were explicitly matched to the grid cell agent (see Supplemental Methods for details). Again, the performance of the grid cell agent was found to be considerably better than these additional place cell agents (Average score over 100 episodes: grid cell agent = 289 vs. best place agent with 660 heterogeneous fields = 212, effect size = 3.93, 95% CI [3.54, 4.31]; best place agent with 256 heterogeneous fields = 225, effect size = 3.52, 95% CI [3.18, 3.87]).

1b - Experimental manipulations to test the Vector-Based navigation hypothesis First, to demonstrate that the goal grid code provided sufficient information to enable the agent to navigate to an arbitrary location, we substituted it with a ”fake” goal grid code sampled randomly from a location in the environment (see Methods). The agent followed a direct path to the newly specified location, circling the absent goal (Fig. 2i) — similar to rodents in probe trials of the Morris water maze (escape platform removed). As a second test, we trained a grid cell agent without providing the goal grid vector to the policy LSTM, effectively ”lesioning” this code. Performance of the grid agent drops to that of the baseline deep RL agent (A3C - a standard deep RL architecture, trained without any grid or place cell input), confirming that the goal grid code is critical for vector based navigation (see Extended Data Fig. 6c). Thirdly, to confirm the presence of a goal-directed vector, we attempted to decode the scalar quantities composing the vector from the policy LSTM. Reasoning that the goal directed vector would be particularly important at the start of a trajectory, we focused on the initial portion of navigation after the agent had reached the goal and was teleported to a new location. We found that the policy LSTM of the grid cell agent contained representations of two key components of vector-based navigation (Euclidean distance, and allocentric goal direction), and that both were more strongly present than in the place cell agent (Euclidean distance difference in r = 0.17; 95% CI [0.11, 0.24]; Goal direction difference in r = 0.22; 95% CI [0.18, 0.26]; Figure 2j&k). Notably, a neural representation of goal distance has recently been reported in mammalian hippocampus[29] (also see [49]). To determine the behavioral relevance of these two metric codes, we examined the goal-homing accuracy in each episode over several steps immediately following the period of metric decoding. We found that variation in both Euclidean distance ($r = 0.22$, 95% CI [-0.32, -0.09]) and allocentric goal direction ($r = 0.22$, 95% CI [-0.38, -0.15]) decoding error correlated with subsequent behavioral accuracy. This suggests that stronger metric codes are indeed important for accurate goal-homing behavior.

Finally, to determine the specific contribution of the grid-like units, we made targeted lesions to the goal grid code and reexamined performance and representation of the goal directed vector. When 25% of the most grid-like units were silenced (see Methods), performance was worse than lesioning 25% at random (average score for 100 episodes: 126.1 vs. 152.5, respectively; effect size = 0.38, 95% CI [0.34, 0.42]). Further, as expected, goal-directed vector codes were more strongly degraded (Euclidean distance: random lesions decoding accuracy $r = 0.45$, top-grid lesions decoding accuracy $r = 0.38$, difference in decoding accuracy = 0.08, 95% CI [0.03, 0.13]). We also performed an additional experiment where the effect of the targeted grid lesion was compared to 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 (A3C[40], 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]and[50], 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 spaces[11]. First, we test these three agents in the "goal-driven" maze (see Methods). The grid-cell agent exhibited high levels of performance, and over the course of 100 episodes, attained an average score of 346.5 (video: https://youtu.be/BWqZwLQfwlM), beating both the place cell agent (average score 258.76; contrast effect size = 1.98, 95% CI [1.79, 2.18]) and the A3C agent (average score 137.00; contrast effect size = 14.31, 95% CI [12.91, 15.71]). The grid cell agent showed markedly superior performance compared to the other agents in the "goal-doors" maze (average score over 100 episodes: grid cell agent = 284.30 vs place cell agent = 90.53, effect size = 7.86, 95% CI [7.09, 8.63]; A3C agent = 48.69, effect size = 7.73, 95% CI [6.97, 8.48]) (video of grid cell agent: https://youtu.be/BWqZwLQfwlM). Interestingly, therefore, the enhanced performance of the grid cell agent was particularly evident when it was necessary to recompute trajectories due to changes in the door configuration, highlighting the flexibility of vector-based navigation in exploiting ad hoc short-cuts (Fig. 3f).

The grid cell agent exhibited stronger performance than a professional human player in both "goal-driven" (average score: grid cell agent = 346.50 vs. professional human player = 261, effect size = 4.00, 95% CI [3.50, 4.52]) and "goal-doors" (average score: grid cell agent = 284.30 vs. professional human player = 240.5, effect size = 2.49, 95% CI [2.18, 2.81]). The human expert received 10 episodes worth of training in each environment before undergoing 20 episodes of testing. This is considerably less training than that experienced by the network. Importantly, however, the mammalian brain has evolved to path integrate and naturally the human expert had a lifetimes worth of relevant navigational experience. Hence, although directly drawing concrete conclusions from relative performance of human and agents is necessarily difficult, providing human-level performance is useful as a broad comparison and represents a commonly used benchmark in similar papers[44].

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

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

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

(Figure 3j&k).

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

Next, to test the agents' abilities to traverse a previously unvisited section of an environment, we employed the "double-E shortcut" maze (Fig. 4d-f, and Extended Data Figure 10e-l). During training, the corridor presenting the shortest route to the goal was closed at both ends, preventing access or observation of the interior. In this simple configuration the grid and place cell agents performed similarly, exceeding the RL control agent (Extended Data Figure 10i). However, at test, when the doors were opened, the grid cell agent was able to exploit the short-cut corridor, whereas the control agents continued to follow the longer route they had previously learnt (Extended Data Figure 10j-l). In the "double-E shortcut" maze performance does not significantly differ between the grid and place cell agents, but both are significantly better than the A3C control (grid cell agent vs. place cell agent, effect size = 0.27, 95% CI [0.24, 0.29]; grid cell agent vs. A3C agent, effect size = 2.99, 95% CI [2.69, 3.29]; place cell agent vs. A3C agent, effect size = 2.92, 95% CI [2.63, 3.21]). When shortcuts become available in the test phase, the grid cell agent performs significantly better than the place agent (grid cell agent vs. place cell agent, effect size = 1.89, 95% CI [1.69, 2.09]; grid cell agent vs. A3C agent, effect size = 12.77, 95% CI [11.48, 14.07]; place cell agent vs. A3C agent, effect size = 14.87, 95% CI [13.35, 16.38]).

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

2a - Backpropagation through time (BPTT) Whilst backpropagation provides a powerful mechanism for adjusting the weights within hierarchical networks analogous to those found in the brain (e.g. the ventral visual stream), it has long been thought to be biologically implausible for several reasons: for example, it requires access to information that is non-local to a synapse (i.e. information about errors many layers downstream). However, recent research in several directions have provided fresh new insights into how a process akin to backpropagation may be implemented in the brain [51]. Whilst less research has been conducted into how BPTT could be implemented in the brain, recent work points to potentially promising avenues that deserve further exploration [52].

2b - Relationship to previous models of grid cells Our work contrasts with previous approaches where grid cells have been hard-wired[53–56] and[57], derived through eigendecomposition of place fields[58,59], or arisen through self organization in the absence of an objective function[60]. It is worth noting that our experiments were not designed to provide insights into the development of grid cells in the brain — due to the limitations of the training algorithm used (i.e. backpropagation) in terms of biological plausibiliy (although see [61]). More generally, however, our findings accord with the perspective that the internal representations of individual brain regions such as the entorhinal cortex arise as a consequence of optimizing for specific ethologically important objective functions (e.g. path integration) — providing a parallel to the optimization process in neural networks[62].

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

3a - Navigation through Deep RL

Probe mazes to test for shortcut behavior The first maze used to test shortcut behaviour was a linearized version of Tolman's sunburst maze[63] (Fig. 4a). The maze was used to determined if the agent was able to follow an accurate heading towards the goal when a path became available. The maze was size 13×13 and contained 5 evenly spaced corridors, each of which had a door at the end closest to the start position of the agent. The agent always started on one side of the corridors with the same heading orientation (North; see Fig 4a) and the goal was always placed in the same location on the other side of the corridors. Until the agent reached the goal the first time only one door was open (door 5, Fig. 4a), but after that all the doors were opened for the remainder of the episode. After reaching the goal, the agent was teleported to the original position with the same heading orientation. This maze was used to test the shortcut capabilities of agents that had been previously trained in the ”goal doors” environment. All the agents were tested in the maze for 100 episodes, each one lasting for a fixed duration of $5,400$ environment steps (90 seconds).

The second maze, termed double E-maze, was designed to test the agents abilities to traverse a previously unvisited section of an environment. The maze was size 12×13 and was formed of 2 symmetric sides each one with 3 corridors. The goal location was always on the bottom right or left, and the location was randomized over episodes. During training, the left and right 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.

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

Place cell agent with heterogeneously sized place fields: to control for differences in the number and area of spatial fields between agents, we also generated two further place cell agents that were explicitly matched to the grid cell agent. Specifically, we used a watershedding algorithm[64] to detect 660 individual grid fields in the grid-like units of the grid cell agent. The distribution of the areas of these fields were found to exhibit 3 peaks — based on a Gaussian fitting procedure — having means equivalent to 2D Gaussians with standard deviations of 8.2cm, 15.0cm, and 21.7cm. Hence we generated a further control agent having 395 place cells of size 8.2cm, 198 of size 15.0cm, and 67 of 21.7cm — 660 place cells in total, the relative numbers reflecting the magnitudes of the Gaussians fit to the distribution. A final control agent was also generated having 256 place cell units in total — the same number of linear layer units as the grid agent — distributed across the same three scales in a similar ratio. Additionally, we note that from a machine learning perspective, the place cell and grid cell agents with the same number of linear layer units are in principle well matched since they are provided with the same input information and have an identical number of parameters.

Place cell prediction agent. The architecture of the place cell prediction agent (Extended Data Figure 9a) is similar to the grid cell agent described in the Methods : the key difference is the nature of the input provided to the policy LSTM as described below. Specifically, 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 current predicted place cell activity vector, $\vec{y}_t$, and the current predicted head direction cell activity vector $\vec{h}_t$ — and the goal predicted place cell activity vector , $\vec{y}_*$, and goal head direction activity vector, $\vec{h}_*$, observed the last time the agent had reached the goal — or zeros if the agent had not yet reached the goal within the episode. The convolutional network had the same architecture described for the grid cell agent.

3c - Training algorithms

We assume the standard reinforcement learning setting where an agent interacts with an environment over a number of discrete time steps. As previously defined the at time t the agent receives an observation o_t along with a reward r_t and produces an action a_t. The agent's state s_t is a function of its experience up until time t, $s_t = f(o_1, r_1, a_1, ..., o_t, r_t)$ (The specifics of o_t are defined in the architecture section). The n-step return $R_{t:t+n}$ at time t is defined as the discounted sum of rewards, $\hat{R}_t = \sum_{i=0...n-1} \gamma^i r_{t+i} + \gamma^n V(s_{t+n}, \theta)$. The value function is the expected return from 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 methods for the details of the loss functions.

3d - Neuroscience-based analyses of units

Gridness score and grid scale calculation Following [20] and [18] spatial autocorrelograms of ratemaps were used to assess the gridness and grid scale of linear layer units. First, for each unit, the spatial autocorrelogram was calculated as defined in [20]. To calculate gridness[20], a measure of hexagonal periodicity, we followed the 'expanding gridness' method introduced by [18]. Briefly, a circular annulus centred on the origin of the autocorrelogram was defined, having radius of 8 bins and with the central peak excluded. The annulus was rotated in 30° increments and, at each increment, the Pearson product moment correlation coefficient with the unrotated version of itself found. An interim gridness value was then defined as the highest correlation obtained from rotations of 30, 90 and 150° subtracted from the lowest at 0, 60 and 120°. This process was then repeated, each time expanding the annuls by 2, up to a maximum of 20. Finally, the gridness value was taken as the highest interim score.

Grid scale[20], a simple measure of the wavelength of spatial periodicity, was defined from the autocorrelogram as follows. The six local maxima closest to but excluding the central peak were identified. Grid scale was then calculated as the median distance of these peaks from the origin.

Directional measures Following[46] the degree of directional modulation exhibited by each unit was assessed using the length of the resultant vector of the directional activity map. Vectors corresponding to each bin of the activity map were created:

$$r_i = \begin{bmatrix} \beta_i \cos \alpha_i \\ \beta_i \sin \alpha_i \end{bmatrix}, \tag{6}$$

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:

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

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 score[47]. 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:

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

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 procedures[65,48].

For the gridness measure and border score, null distributions were constructed using a 'field shuffle' procedure equivalent to that specified by[48]. Briefly, a watershedding algorithm[64] 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 activity.

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

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

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, following [22], 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 measure'[22] of the distribution of their scales. Specifically, scales were binned into a histogram with 13 bins distributed evenly across a range corresponding to scales 10 to 36 spatial bins. 'Discreteness' was defined as the standard deviation of the counts in each bin. Again following[22], statistical significance for this value was obtained by comparing it to a null distribution generated from a 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 algorithm[64] 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 the watershedding procedure, and these units were silenced. The results were not qualitatively different from the former experiment (data not shown).

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

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

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

3f - Statistical reporting

We followed the guidelines outlined by[67]. Specifically reporting effect sizes and confidence intervals. Unless otherwise stated, the effect sizes are calculated using the following formula:

$$effect\ size = \frac{\mu_{group1} - \mu_{group2}}{\sigma_{pooled}}, \tag{9}$$

and the σ_{pooled} was calculated accordingly to[68] using:

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

The confidence interval for the effect size was calculated accordingly to[69] using:

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

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_{R_H}	0.25	Velocity reduction factor when located in the perimeter
Δ_{R_H}	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 bu 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

49. Chadwick, M. J., Jolly, A. E., Amos, D. P., Hassabis, D. & Spiers, H. J. A goal direction signal in the human entorhinal/subicular region. *Current Biology* **25**, 87–92 (2015).

50. Kubie, J. L. & Fenton, A. A. Linear look-ahead in conjunctive cells: an entorhinal mechanism for vector-based navigation. *Frontiers in neural circuits* **6**, 20 (2012).

51. Scellier, B. & Bengio, Y. Towards a biologically plausible backprop. *arXiv preprint arXiv:1602.05179* **914** (2016).

52. Ke, N. R. *et al.* Sparse attentive backtracking: Long-range credit assignment in recurrent networks. *arXiv preprint arXiv:1711.02326* (2017).

53. Burgess, N., Barry, C. & O'keefe, J. An oscillatory interference model of grid cell firing. *Hippocampus* **17**, 801–812 (2007).

54. Hasselmo, M. E., Giocomo, L. M. & Zilli, E. A. Grid cell firing may arise from interference of theta frequency membrane potential oscillations in single neurons. *Hippocampus* **17**, 1252–1271 (2007).

55. Burak, Y. & Fiete, I. R. Accurate path integration in continuous attractor network models of grid cells. *PLoS Comput Biol* **5**, e1000291 (2009).

56. Fuhs, M. C. & Touretzky, D. S. A spin glass model of path integration in rat medial entorhinal cortex. *Journal of Neuroscience* **26**, 4266–4276 (2006).

57. Gustafson, N. J. & Daw, N. D. Grid cells, place cells, and geodesic generalization for spatial reinforcement learning. *PLoS Comput Biol* **7**, e1002235 (2011).

58. Stachenfeld, K. L., Botvinick, M. & Gershman, S. J. Design principles of the hippocampal cognitive map. In *Advances in neural information processing systems*, 2528–2536 (2014).

59. Dordek, Y., Soudry, D., Meir, R. & Derdikman, D. Extracting grid cell characteristics from place cell inputs using non-negative principal component analysis. *eLife* **5**, e10094 (2016).

60. Widloski, J. & Fiete, I. How does the brain solve the computational problems of spatial navigation? In *Space, Time and Memory in the Hippocampal Formation*, 373–407 (Springer, 2014).

61. Bengio, Y., Lee, D.-H., Bornschein, J., Mesnard, T. & Lin, Z. Towards biologically plausible deep learning. *arXiv preprint arXiv:1502.04156* (2015).

62. Marblestone, A. H., Wayne, G. & Kording, K. P. Toward an integration of deep learning and neuroscience. *Frontiers in Computational Neuroscience* **10** (2016).

63. Tolman, E. C. *et al.* Cognitive maps in rats and men (1948).

64. Beucher, S. Use of watersheds in contour detection. In *Proceedings of the International Workshop on Image Processing* (CCETT, 1979).

65. Yartsev, M. M., Witter, M. P. & Ulanovsky, N. Grid cells without theta oscillations in the entorhinal cortex of bats. *Nature* **479**, 103–107 (2011).

66. Schwarz, G. Estimating the dimension of a model. *The Annals of Statistics* **6**, 461–464 (1978).

67. Halsey, L. G., Curran-Everett, D., Vowler, S. L. & Drummond, G. B. The fickle p value generates irreproducible results. *Nature methods* **12**, 179 (2015).

68. Olejnik, S. & Algina, J. Measures of effect size for comparative studies: Applications, interpretations, and limitations. *Contemporary educational psychology* **25**, 241–286 (2000).

69. Hedges, L. & Olkin, I. *Statistical Methods for Meta-analysis* (Academic Press, 1985). URL https://books.google.co.uk/books?id=brNpAAAAMAAJ.