Neuron

Article

A unified theory for the computational and mechanistic origins of grid cells

Highlights

- RNNs trained to path integrate with nonnegative firing develop hexagonal grid cells
- Pattern formation theory explains hexagonal lattices and the role of nonnegativity
- Novel analyses show that trained nets generalize published path integrator models
- Trained models fit neural responses significantly better than hand-designed models

Authors

Ben Sorscher, Gabriel C. Mel,
Samuel A. Ocko, Lisa M. Giocomo,
Surya Ganguli

Correspondence

meldefon@stanford.edu

In brief

Sorscher et al. investigate RNNs trained to path integrate and use pattern formation theory to explain when, why and how such networks develop hexagonal grid cells, finding nonnegative firing rates and center-surround outputs are essential. Additional analyses demonstrate trained RNNs fit neural responses better than hand-designed models.

Sorscher et al., 2023, Neuron *111*, 121–137
January 4, 2023 © 2022 Published by Elsevier Inc.
https://doi.org/10.1016/j.neuron.2022.10.003

CellPress

Article

A unified theory for the computational and mechanistic origins of grid cells

Ben Sorscher,[1,4] Gabriel C. Mel,[2,4,5,*] Samuel A. Ocko,[1] Lisa M. Giocomo,[3] and Surya Ganguli[1,3]
[1]Department of Applied Physics, Stanford University, Stanford, CA 94305, USA
[2]Neurosciences PhD Program, Stanford University, Stanford, CA 94305, USA
[3]Department of Neurobiology, Stanford University School of Medicine, Stanford, CA 94305, USA
[4]These authors contributed equally
[5]Lead contact
*Correspondence: meldefon@stanford.edu
https://doi.org/10.1016/j.neuron.2022.10.003

SUMMARY

The discovery of entorhinal grid cells has generated considerable interest in how and why hexagonal firing fields might emerge in a generic manner from neural circuits, and what their computational significance might be. Here, we forge a link between the problem of path integration and the existence of hexagonal grids, by demonstrating that such grids arise in neural networks trained to path integrate under simple biologically plausible constraints. Moreover, we develop a unifying theory for why hexagonal grids are ubiquitous in path-integrator circuits. Such trained networks also yield powerful mechanistic hypotheses, exhibiting realistic levels of biological variability not captured by hand-designed models. We furthermore develop methods to analyze the connectome and activity maps of our networks to elucidate fundamental mechanisms underlying path integration. These methods provide a road map to go from connectomic and physiological measurements to conceptual understanding in a manner that could generalize to other settings.

INTRODUCTION

The discovery of spatially regular hexagonal grid-cell firing patterns in the medial entorhinal cortex (MEC) has been widely observed as a function of spatial position in mice (Fyhn et al., 2008), rats (Hafting et al., 2005), and bats (Yartsev et al., 2011) and as a function of gaze position in monkeys (Killian et al., 2012). In addition, fMRI studies have revealed evidence for grid-like representations in humans (Doeller et al., 2010). The regularity and ubiquity of grid cells raises two related classes of scientific questions. First, what *generic* circuit mechanisms might give rise to grid cells? Second, what computational reasons might explain the pervasive existence of grid cells across many species? In essence, if hexagonal grid cells are evolution's answer to an ethologically relevant computational question, then what is that question?

On the mechanistic side, many works have hand tuned the connectivity of model recurrent neural circuits with a center-surround structure specifically to generate grid-cell firing patterns (Fuhs and Touretzky, 2006; Guanella et al., 2007; Burak and Fiete, 2009; Ocko et al., 2018a), building on prior models of head-direction cells (Skaggs et al., 1994; Blair, 1996; Zhang, 1996; Redish et al., 1996; Hahnloser, 2003; see also Ben-Yishai et al., 1995) and place cells (McNaughton et al., 1996; Samsonovich and McNaughton, 1997; Conklin and Eliasmith, 2005). Such continuous attractor models can robustly integrate velocity to store spatial position via path integration (Burak and Fiete, 2009). More recent attractor networks that incorporate plastic inputs from landmark cells can explain why grid cells deform in irregular environments (Skaggs et al., 1994; Ocko et al., 2018a), and when they either phase shift or remap in altered virtual reality environments (Campbell et al., 2018). However, such hand-tuned models raise two issues. First, they involve many choices about circuit connectivity and dynamics, and it is unclear how generic such choices are. In essence, could there be different classes of neural networks that both path-integrate and generate hexagonal firing patterns? Second, none of these models demonstrate that hexagonal firing patterns naturally arise as the optimal solution to any computational problem, precisely because these patterns are assumed in the first place by hand tuning the connectivity.

In contrast, normative models attempt to shed light on the question of *why* grid firing patterns might be found in many species by demonstrating that these patterns are optimal for a solving a particular task. For example, Dordek et al. (2016) demonstrated that single neurons that receive place cell inputs through plastic synaptic weights undergoing Oja's learning rule develop grid-like receptive fields (RFs) with a square lattice structure. If these synapses are also constrained to be positive, then these same neurons learn hexagonal grid-like RFs. An alternative approach focuses on optimal representations of the statistics of spatial transitions (Stachenfeld et al., 2017; Whittington et al., 2018), finding that square grids are optimal in square environments. Other works have assumed the existence of grid-like representations with different lattice structures, finding hexagonal lattices, outperform other lattice structures in terms of either decoding position under noise (Mathis et al., 2015) or a notion of economy (Wei et al., 2015). However, unlike the hand-tuned attractor network models described above, these normative approaches primarily tackle the issue of spatial representations and do not address the central issue of how neural circuits might actually solve the problem of path integration, which is believed to be a computational function of entorhinal cortex, whether in real space or in abstract spaces (Aronov et al., 2017; Constantinescu et al., 2016).

More recent normative approaches have tackled this central issue by training, rather than hand tuning, neural networks to accurately solve the navigational problem of path integration. Indeed, Cueva and Wei (2018) found that square grid cells spontaneously emerge in square environments in such trained networks. Also, Banino et al. (2018) suggested that hexagonal grid cells emerge even in square environments, though the grid-cell patterns were highly heterogeneous. The process of training neural networks to solve different computational problems can be a powerful method for hypothesis generation in neuroscience, yielding a more unbiased search over the space of circuit solutions than might be possible through imagination alone. Indeed, the representations of trained rather than hand-designed networks have been successfully compared with actual neural representations in the retina (McIntosh et al., 2016; Ocko et al., 2018b; Lindsey et al., 2019; Tanaka et al., 2019), inferotemporal cortex—V4 and V1 (Yamins et al., 2014; Yamins and DiCarlo, 2016)—motor cortex (Sussillo et al., 2015), and prefrontal cortex (Mante et al., 2013). However, in comparisons to the behavior of entorhinal representations, the trained networks in (Cueva and Wei, 2018) and (Banino et al., 2018) exhibit mismatches along two important dimensions. First, as we will show below, the learned lattice structure is inconsistent with data (square versus hexagonal in the case of Cueva and Wei [2018], and smooth and random versus truly hexagonal in the case of Banino et al. [2018]). Second, as we will show below, the methods of training neural networks in Banino et al. (2018) do not yield grid cells whose representations generalize to expanded environments.

Despite these mismatches, these two works exemplify interesting advances in generating, in a more unbiased manner, neural networks that path-integrate with grid-like representations, and they naturally suggest important yet unanswered questions. First, when and why do grid-like representations spontaneously emerge from neural circuits that are trained to solve navigational problems, or other normative problems like efficient spatial encoding (Dordek et al., 2016)? Second, if grid-like representations appear, when and why are they sometimes square, hexagonal, or heterogeneous? Third, focusing on the networks trained to path integrate, what circuit mechanisms yield grid-like responses? Are these circuit mechanisms for both path integration and grid cells in trained models at all related to the circuit mechanisms in prior hand-tuned models? In essence, how can we obtain conceptual insight into how circuit connectivity and dynamics conspire to yield the emergent computational functions of these trained circuits? This latter question is a foundational question for neuroscience in general, especially as we encounter more and more circuit models generated via machine learning-based training methods (Tanaka et al., 2019). Although such models often involve a considerable simplification of the neural microcircuitry (see, e.g., Winterer et al., 2017), they have the advantage of giving us access to the *entire* connectome and the activity patterns of every model neuron. Thus, the process by which we might go from such connectivity and activity data to conceptual understanding (Gao and Ganguli, 2015) could be instructive in teaching us how we might leverage data generated by investments in both large-scale connectomics (Seung, 2009) and brain activity maps (Insel et al., 2013). In the following, we address the above questions through circuit simulations, theory, and comparisons to experiments. We summarize our results in the discussion.

RESULTS

Diverse lattice structures and generalization properties of trained path integrators

Path integration, the process of integrating instantaneous velocity signals to obtain an estimate of current position, is thought to be a major computational function of MEC. Although much theoretical work based on hand-designed attractor models suggests a close connection between entorhinal grid cells and path integration, there is little direct experimental evidence for this connection due to the difficulty of cleanly separating grid cells from the rest of MEC (see e.g., Gil et al., 2018). We sought to work in reverse, by elucidating the conditions under which the computational demand of path integration, in conjunction with simple biologically plausible constraints, might naturally lead to the spontaneous emergence of hexagonal grid cells in a trained neural network.

In particular, we simulated an animal exploring a square environment following a smooth random walk (Figure 1A). As the animal moves, different subsets of simulated place cells become active. At each time step, the network receives the animal's 2-dimensional body velocity $\vec{v}(t)$ as input. Although recent work shows that primarily *head*, rather than *body*, direction is represented within MEC (Raudies et al., 2015; Gerlei et al., 2020), we assumed that a body velocity signal is available in the *input* and were agnostic as to whether this same signal would be found in the model MEC neurons themselves (see discussion). The velocity signal is integrated by the network's recurrently connected units, and the entire network is trained to report its current position by generating an output place cell code (Figure 1B), mirroring the basic grid cell-to-place cell organization observed in MEC and hippocampus. We note that this training procedure, based on backpropagation, is not meant to capture how grid cells might actually develop in the brain. Our goal was simply to discover networks that path integrate in a manner that is less biased than traditional approaches based on hand design.

Recent work (Cueva andWei, 2018; Banino et al., 2018) has shown that recurrent neural networks trained on path-integration tasks learn grid-like representations in their hidden units. However, the grids in Cueva and Wei (2018) were square, and the representations in Banino et al. (2018) had two biologically unrealistic characteristics. First, we found that even the highest

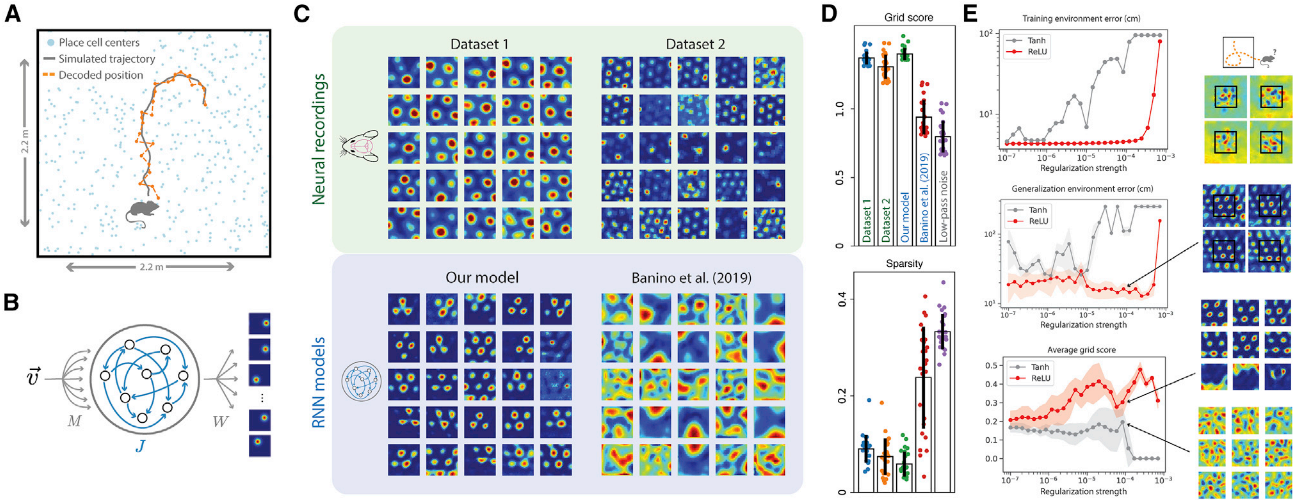

Figure 1. Learned spatial representations of neural networks trained to path integrate

(A) A simulated animal trajectory (gray curve), and the decoded position from the network's output (orange curve). Place cell centers (blue dots) are distributed uniformly and isotropically over a square enclosure.

(B) Our general model architecture includes a velocity input that is fed to a set of recurrently connected hidden units. These hidden units must then generate the desired place cell output representation.

(C) Comparing spatial representations of cells in the brain (top row) with those in trained RNNs (bottom row). Each panel shows the top 25 most grid-like cells. Our model closely matches the regular hexagonal firing fields of grid cells in the brain. Dataset 1 is obtained from Butler et al. (2019), dataset 2 is obtained from Gardner et al. (2022), and the units at bottom right are the top 25 most grid-like cells from Extended Data Figure 2 of Banino et al. (2018).

(D) We quantify the similarity between grid-like units in the trained RNNs and grid cells in the brain using two metrics: grid score (based on 6-fold symmetry of the rate maps; Langston et al., 2010) and firing sparsity, defined as the fraction of time firing between the 40th and 60th percentile of firing rate. Bars show stadard deviation of scores.

(E) We disentangle the roles of regularization strength and non-negativity and study their effects on training error (top), generalization error (middle), and average grid score (bottom). Training error remains low for a wide range of regularization strengths (top) but fails when this strength is above a certain threshold. Generalization performance outside the training arena (middle) improves with increasing regularization strength, up until this threshold. Average grid score (bottom) is significantly higher in ReLU networks than in Tanh networks for all choices of regularization strength.

grid-score units in their network (Figure 1C, bottom right) did not match the regular hexagonal firing of grid cells in the brain (Figure 1C, top, data from Butler et al. 2019 and Gardner et al., 2022), either qualitatively (Figure 1C) or quantitatively (Figure 1D). Indeed, we found that the units in their network did not have higher grid scores than low-pass filtered noise maps (Figure S2). Second, the network was unable to path integrate outside of the training arena (Figure 1E, upper inset). When the animal's simulated walk passed beyond a removed wall of the square environment used during training, the hidden unit activities in the trained network usually froze at their boundary values, rather than continuing to fire in a spatially informative way, as biological grid cells do (Savelli et al., 2008).

We found that we could obtain similar results in a significantly simpler recurrent neural network (RNN) architecture (shown in Figure 1B), without the complexity of the architecture of Banino et al. (2018) (see "rnn training" for details). The simple RNN learned to path integrate comparably well, gave rise to stable rate maps over the environment, and developed qualitatively similar grid-like patterns in its hidden units. The simple RNN architecture has several advantages: (1) the grid cells are recurrently connected, like grid cells in MEC and unlike those in Banino et al. (2018), and (2) this architecture corresponds exactly to traditional path-integrator models of grid cells, except that although the recurrent weights in traditional models are chosen by hand, ours are learned over the course of training.

Next, we found that two simple changes to the training procedure encouraged the network to (1) reliably learn regular hexagonal grids like those in MEC and (2) learn a path-integration mechanism that generalizes outside of the training environment, resolving both the problems identified above and rendering the model more biologically realistic. First, inspired by Dordek et al. (2016), we retrained the network with the additional constraint that hidden unit activities might be nonnegative by simply changing the single-neuron nonlinearity from hyperbolic tangent to a rectified linear unit (ReLU) (Figure 1C, black traces at left). Under these conditions, many units develop strikingly regular hexagonal grid maps similar to those of entorhinal grid cells (Figure 1C, lower), and the distribution of grid scores shifts to larger values (Figure 1E; see also Figure S2 for examples of model grid cells with different grid scores; compare with Extended Data Figure 2 of Banino et al., 2018). Second, we found that training with a small amount of regularization via weight decay on the recurrent weights, leading to small synaptic strengths, encourages a path integration mechanism that continues to operate well beyond the walls of the training environment (Figure 1E, middle). In Figure 1E, we disentangle the roles of nonnegativity and weight decay. We find that models with and without nonnegativity constraints can be trained to achieve

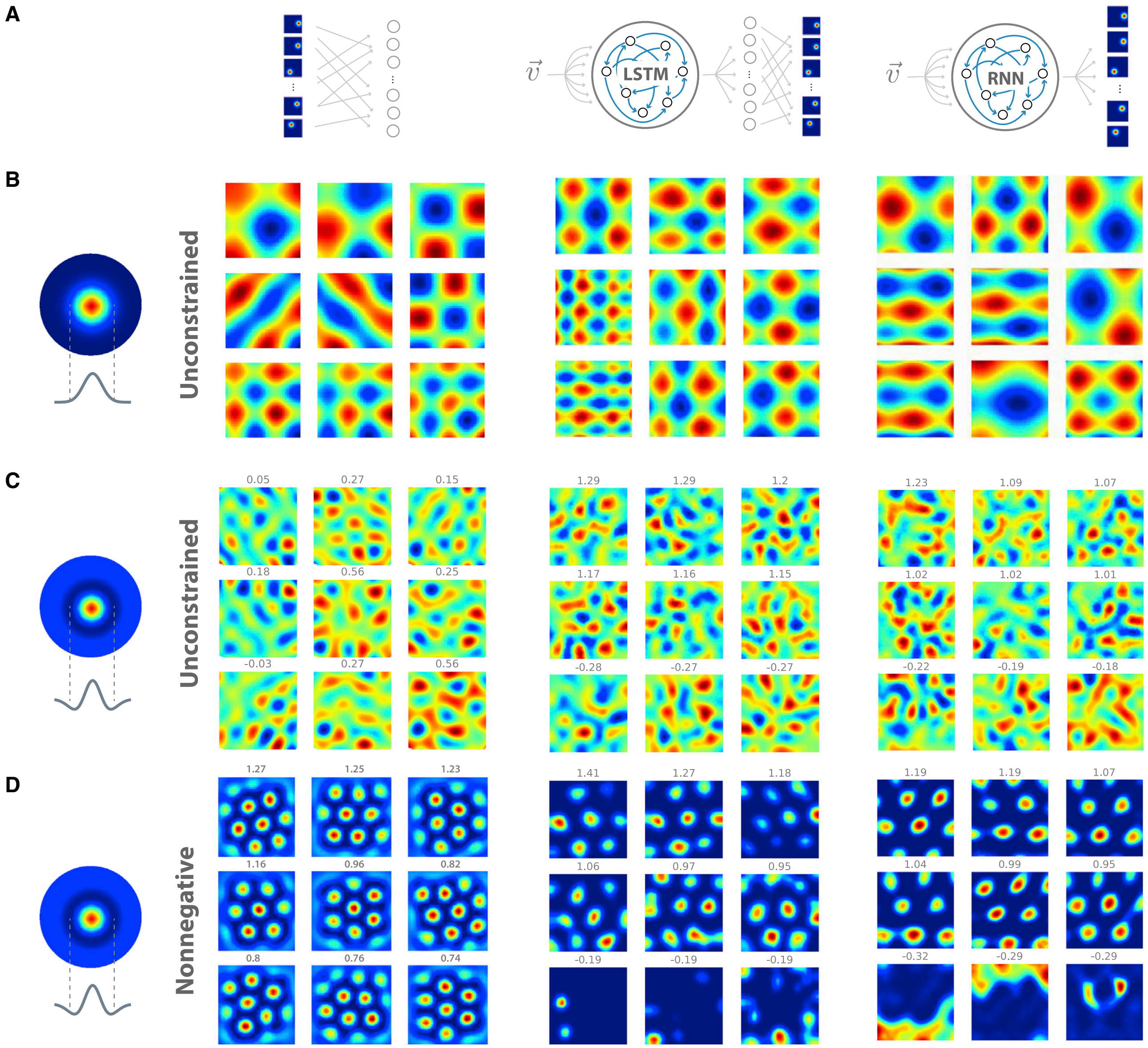

Figure 2. Neural networks trained on normative tasks develop grid-like firing fields

(A) From left to right, we train a single layer neural network, an LSTM, and an RNN on place cell outputs, reproducing the results of Dordek et al. (2016), Cueva and Wei (2018), and Banino et al. (2018).

(B) When the place cell RF field (left) is broad, all networks learn square grids.

(C) When the place cell RF exhibits center-surround structure, all networks learn amorphous, quasi-periodic patterns.

(D) In addition to place cell center-surround structure, when a nonnegativity constraint is imposed on hidden unit activations, all networks now learn regular hexagonal grids.

low path integration error in the training environment, provided the weight-decay regularization is not too strong. Increasing regularization improves generalization performance up until the point where training fails. Finally, including a nonnegativity constraint promotes hexagonal grid cells with substantially higher average grid score for all choices of regularization strength.

After demonstrating that training a simple RNN with nonnegative firing rates leads to the spontaneous emergence of hexagonal grid maps (Figure 1C, bottom left), we tested whether this effect generalizes to different network architectures and further explored the effects of the target place cell output. We focused on three cases. First, we explored the feedforward architecture of Dordek et al. (2016) that does not path integrate but rather takes place cell *inputs* and learns grid-cell outputs (Figure 2A, left). Second, we considered the complex path integrator of Banino et al. (2018) that generates grid responses in a *disconnected* layer of neurons presynaptic to the desired place cell code and

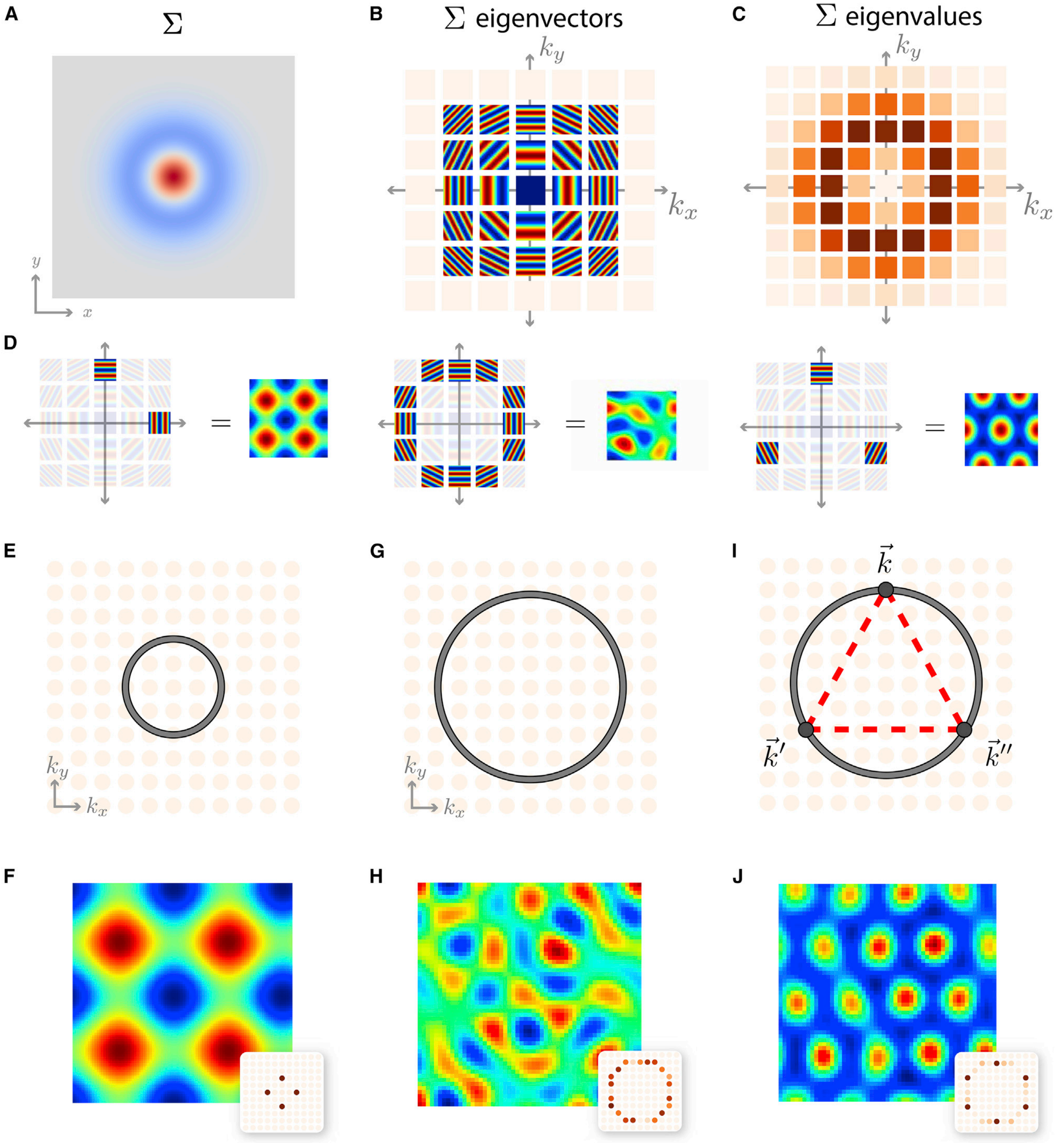

Figure 3. A theory for predicting the structure of learned spatial maps

(A) A central row of the place cell correlation matrix Σ, indicating similarity in the target place cell code as a function of spatial displacement, illustrating center-surround structure as in Figure 2CD. Red (blue) indicates positive (negative) similarity.

(B) Eigenvectors of Σ are well approximated by Fourier plane waves and are shown arranged on a discrete lattice of integers (k_x, k_y) corresponding to the frequencies of each plane wave along the two cardinal axes.

(C) The eigenvalues of Σ corresponding to the eigenvectors in (B). Darker red indicates larger eigenvalues.

(D) Particular linear combinations of plane waves yield grid patterns. A square grid can be generated by combining waves that oscillate along the cardinal axes (left). A heterogeneous grid can be generated by arbitrary combinations of waves with frequencies near an annulus of fixed radius (middle). A hexagonal grid pattern arises when waves whose frequencies form an equilateral triangle in the (k_x, k_y) plane are combined (right).

(legend continued on next page)

postsynaptic to a hidden recurrently connected network (Figure 2A, middle). This network also possesses complex nonlinear single-neuron properties corresponding to long-short term memory (LSTM) cells (Dordek et al., 2016). Third, we considered our simple architecture that generates grid-like responses in a single hidden layer of recurrently connected neurons with simple sigmoidal or ReLU nonlinearities (Figure 2A, right) (see "rnn training" for details of all architectures and training).

These three diverse architectures exhibited fairly universal properties of the learned hidden representations as a function of both the assumed place cell code and the constraints on synapses in the feedforward model or firing rates in the recurrent models. First, for wide place cell outputs without any surround inhibition, and with no constraints on synapses or firing rates, all three models learned *square* grid-like responses (Figure 2B), similar to those found in Cueva and Wei (2018). If place cells have a surround inhibition, then all three models learn highly heterogeneous grid-like responses (Figure 2C). In the feedforward model, this surround inhibition corresponds to place cell input firing rates suppressed below spontaneous rates, as in Dordek et al. (2016). In the recurrent models, this surround corresponds to the layer of grid-like cells exciting the place cells with spatial RFs closest to the current position and inhibiting the neighboring place cells. If in addition to the surround inhibition, we further constrain either synapses or firing rates to be nonnegative, then all three models learn hexagonal firing fields (Figure 2D).

A theory for the emergence of diverse grid structures in trained neural circuits

The collection of training experiments in Figure 2 raises an intriguing question: why do these diverse neural architectures, across multiple tasks, all converge to grid-like solutions, and what governs the lattice structure of this solution? We address this question by noting that all of the models described above contain within them a common position encoding sub-problem that involves selecting an optimal hidden representation that can generate place cell activity patterns through one layer of synaptic transformation with minimum neural activity cost. We develop our mathematical theory of this common sub-problem in full detail in "pattern formation theory predicts structure of learned representations" and summarize its salient points here at a conceptual level. Overall, our theory allows us to understand the nature and structure of the resultant grid-like solutions in Figure 2, and their dependence on various modeling choices. Readers who are primarily interested in how hexagonal grid-cell responses are mechanistically generated from the connectivity and dynamics of the learned circuit, as well as comparisons to neural data, can safely skip this section.

To define the position encoding problem, we begin with a minimal subcircuit found in all of the above networks, schematized as $\mathbf{g}\overset{w_i}{\rightarrow}\mathbf{p}_i$. Here, $\mathbf{g}$ denotes an n_x dimensional firing rate vector of a position encoding cell across n_x spatial bins; when the animal is at spatial bin x, this cell's firing rate is given by $g(x)$. Similarly, $\mathbf{p}_i$ for $i = 1, \ldots, n_p$ denotes the firing rate vectors of n_p place cells with similarly defined rate maps $p_i(x)$. Finally, w_i denotes the strength of a feedforward synaptic connection from the position encoding cell to place cell i. We collect the place cell firing rate vectors as the n_p columns of the n_x by n_p matrix P. Now, the goal of position encoding problem is to choose an optimal position encoding rate map $g(x)$ satisfying two criteria: accurately generate the desired place cell activities $p_i(x)$ and minimize a negativity cost $\sigma(g)$ that is large (small), if g is negative (positive). After optimizing over the synaptic weights w_i, the position encoding problem can be expressed as:

$$\min_g \| P - \mathbf{g}\mathbf{g}^T P\|^2 + \sigma(\mathbf{g}) \quad \text{subject to } \mathbf{g}^T\mathbf{g} = 1. \quad \text{(Equation 1)}$$

According to our theory, the solutions to (1) can be roughly understood by considering the two terms separately. Minimizing the first term requires picking an encoding map $\mathbf{g}$ that accurately reconstructs place cell rate maps and corresponds exactly to extracting the top principal component (PC) of the place cell activities. As we explain below, this activity has multiple top PCs with equal variance, giving rise to multiple encoding rate maps with equal accuracy. The second term breaks the tie: among equally accurate encoding maps, $\sigma(\mathbf{g})$ favors the one with the lowest negativity cost.

We now perform a detailed analysis of each term. As noted above, the first term requires us to extract the top principal component of the place cell activity. To do so, we construct the n_x by n_x spatial correlation matrix Σ of the place cell activities. Its matrix elements are $\Sigma_{xx'} = (PP^T)_{xx'} = \sum_i p_i(x)p_i(x')$. Here, a positive (negative) matrix element $\Sigma_{xx'}$ quantifies how similar (dissimilar) the place cell population code is at two points x and x' in space. Figure 3A shows a single row of the spatial correlation matrix where x is the center of a 2D enclosure and x' varies over the enclosure, in the case where the place cell maps $p_i(x)$ have a center-surround structure as in Figures 2C and 2D. The similarity is high and positive for points x' near the center, low and negative for points further away, and close to zero for points even further away.

Next, the top principal component is obtained by extracting the eigenvectors and eigenvalues of $\Sigma_{xx'}$. Indeed, such eigenvectors of inputs and outputs often determines the nature of learned representations in neural networks, from the development of ocular dominance columns (Miller et al., 1989) to the

(E) If the similarity structure in (A) is wide, the annulus of Σ eigenvalues will be small, shown in gray, and will only intersect a few plane modes oscillating primarily along the cardinal axes.
(F) Simulations confirm that neural circuits with unconstrained firing rates will learn combinations of precisely these cardinal modes, generating square grids.
(G) For narrow similarity structure, the large eigenvalues of Σ lie near an annulus of large radius, shown in gray.
(H) Simulations confirm that neural circuits with unconstrained firing rates learn arbitrary linear combinations of plane waves with oscillations frequencies near this annulus, generating heterogeneous grids.
(I) The nonnegativity constraint creates a cooperative interaction among frequency triples that sum to 0 as vectors in the (k_x, k_y) lattice. Since these frequencies must lie on the annulus of large eigenvalues, they must form an equilateral triangle (dashed red).
(J) Simulations confirm that neural circuits with nonnegative firing rates learn hexagonal grids, as predicted in (D) (right) and (I). Panels (F), (H), and (J) show the learned grid-cell representation and Fourier power spectrum (inset).

development of semantic categories (Saxe et al., 2014, 2018). In our context, the eigenvectors are rate maps, i.e., spatial functions of 2D position and are well approximated by Fourier plane waves that oscillate in different frequencies and directions (Figure 3B; see "pattern formation theory predicts structure of learned representations"). The eigenvectors are indexed by two integers, k_x and k_y, indicating the spatial frequency of oscillation in each of the two cardinal spatial directions. Each such eigenvector has an associated nonnegative eigenvalue. These eigenvalues are shown in Figure 3C at the integer (k_x, k_y) lattice points associated with the corresponding eigenvectors in Figure 3B. The strength of these eigenvalues can be obtained by computing the power in each Fourier mode at spatial frequency (k_x, k_y) of the similarity function displayed Figure 3A (see "pattern formation theory predicts structure of learned representations"). Because this similarity function has a center-surround structure, the maximal Fourier power occurs near an annulus in the (k_x, k_y) lattice, and the narrower the similarity function in Figure 3A, the larger the radius of this annulus in Figure 3C.

Crucially, as Figure 3C shows, there are multiple maximal eigenvalues distributed over a ring centered on the origin, corresponding to multiple equal-variance top principal components. Consequently, there is an entire family of equally accurate grid maps consisting of arbitrary linear combinations of eigenvectors in Figure 3B with maximal associated eigenvalue (see "pattern formation theory predicts structure of learned representations"). Figure 3D indicates how, for example, square, heterogeneous, or hexagonal grid patterns can be constructed from appropriate combinations of these top principal components. The key issue then is how does the structure of the place cell code conspire with constraints on hidden representations of the form $\sigma(g)$ to generate these three types of grid codes?

Our theory answers this question by elucidating three general scenarios that lead to these three qualitatively distinct types of codes. In the first case, if the place cell similarity structure in Figure 3A is wide relative to the size of the enclosure, then the maximal eigenvalues will occur near an annulus of small radius, as in Figure 3E. This annulus will intersect a small number of lattice points in the (k_x, k_y) plane corresponding to low frequency eigenmode oscillations aligned along the cardinal axes of the enclosure, and linear combinations of these oscillations along these cardinal directions would predict square grid cells as in Figure 3D, left. This prediction is confirmed in simulations of our position encoding problem in Figure 3F (see "pattern formation theory predicts structure of learned representations"). Indeed, square grid cells were previously found in trained path integrators (Cueva and Wei, 2018). On the other hand, if the similarity structure in Figure 3A is narrow, the maximal eigenvalues will occur near an annulus of large radius, which intersects many lattice points in the (k_x, k_y) plane, as in Figure 3G. As described above, without further constraints $\sigma(\mathbf{g})$, the hidden representation of neural circuits will learn arbitrary linear combinations of eigenmodes associated with the many lattice points on the large annulus, yielding relatively heterogeneous patterns, as predicted in Figure 3D, middle. This prediction is confirmed in simulations in Figure 3H. Indeed, Banino et al. (2018) found highly heterogeneous grid-like representations, with a few cells having a high grid score, but the entire distribution of grid scores was indistinguishable from that obtained by grid patterns obtained by low-pass filtering random noise, as demonstrated in Figure 1E

We now turn to the effect of the second term, $\sigma(\mathbf{g})$. As noted above, $\sigma(\mathbf{g})$'s role is to single out maps with minimum negativity penalty among the equally accurate maps found on the ring of maximal eigenvalue in Figures 3B and 3C. We prove (see "pattern formation theory predicts structure of learned representations") that the effect of this penalty is to favor 3-fold combinations of eigenvectors whose spatial frequencies (k_x, k_y) form an equilateral triangle centered at the origin (Figure 3I). This combination of eigenvectors predicts a hexagonal grid representation as in Figure 3D, right. This prediction is confirmed in simulations of our position encoding problem in Figure 3J. Thus, although non-hexagonal maps are also possible, they are not optimal in the sense of solving the problem with both minimum activity and nonnegativity.

Taken together, our theory provides a unifying conceptual explanation for when and why square, heterogeneous, or hexagonal grids spontaneously emerge across three diverse architectures depicted in Figure 2A and elucidates why prior work (Cueva and Wei, 2018; Banino et al., 2018) on training path-integrator neural networks did not find truly hexagonal grid structure. We have shown that under a wide range of assumptions, any network trained to solve the position-encoding problem of efficiently generating relatively narrow center-surround place cell maps with minimum total grid-cell activity consisting of only nonnegative firing rates will have a tendency to develop hexagonal grid maps. We now move beyond the computational origin of grid cells to understanding the mechanistic origins of these cells in trained neural networks.

Two-dimensional attractor dynamics underlies path integration in trained networks

Any neural circuit that is required to maintain a memory trace of position while the animal is standing still, with no velocity or other sensory inputs, must maintain a set of stable attractor patterns of neural activity. Many traditional hand-designed neural network models for grid cells build in, *by design*, a 2D attractor manifold of stable activity patterns that has the structure of a torus (Conklin and Eliasmith, 2005; Fuhs and Touretzky, 2006; Guanella et al., 2007; Burak and Fiete, 2009; Ocko et al., 2018a). This raises the question of whether our network, which does not build in any attractor structure *a priori*, naturally develops a similar toroidal manifold. We took advantage of having complete access to both the connectome and dynamics of our trained networks to search for and characterize neural activity patterns that are stable for long periods of time, using the methods of Sussillo and Barak (2013). We found a large number of such attractor patterns that when projected into a high-variance 6 dimensional subspace could be arranged continuously along a *2D* manifold with the shape of a torus (Figure 4A; see "attractor manifold analysis" for details). Moreover, the attractor patterns were in correspondence with both positions in physical space (Figure 4A, red dashed lines) and the network's decoded position (Figure 4A, blue-green colormap). Furthermore, we found that as the animal moves from one end of the enclosure to another, at least within this 6-dimensional subspace, the neural trajectory wraps

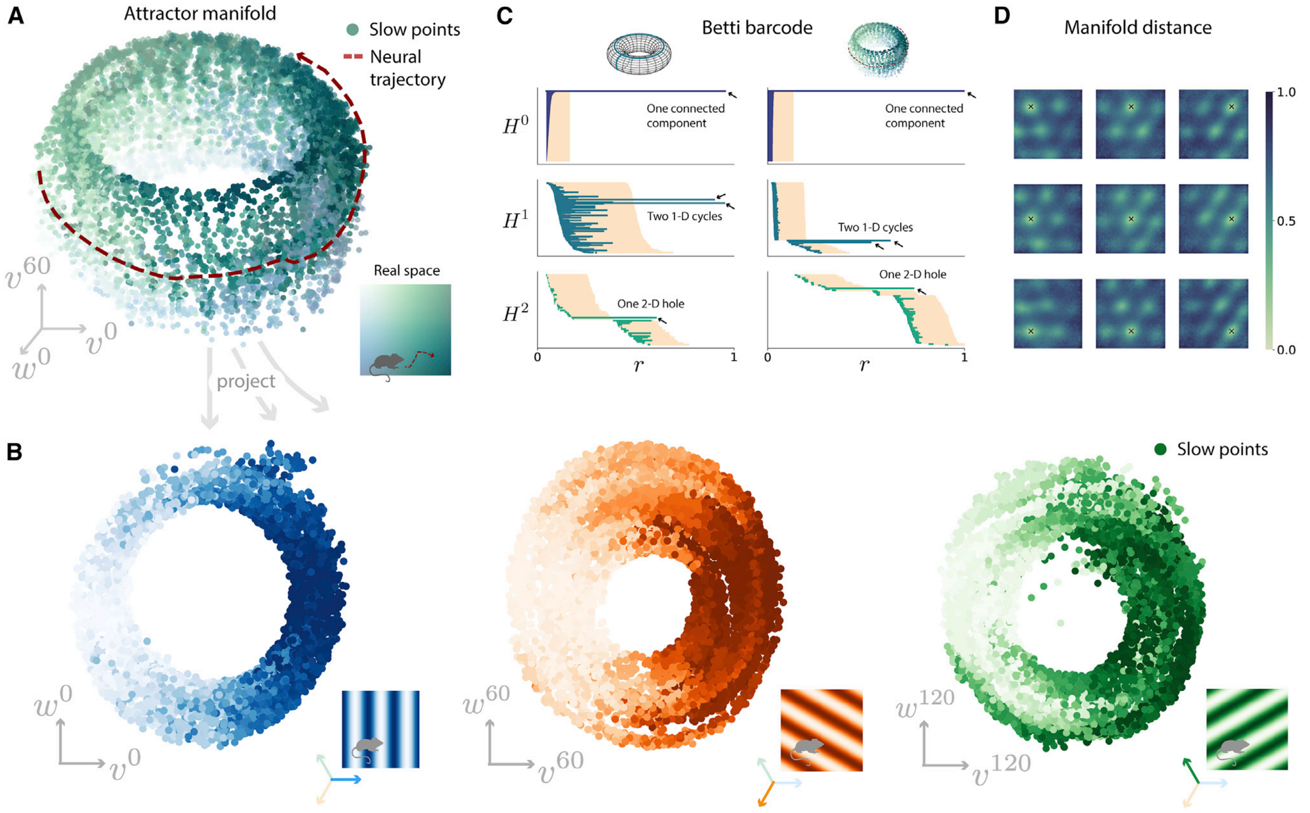

Figure 4. Emergent two-dimensional attractor neural dynamics in trained path integrators

(A) A toroidal manifold of stable attractor patterns in neural activity space, visualized via dimensionality reduction, with each attractor pattern colored by the decoded position within the 2d environment in the inset. The neural trajectory corresponding to the animal trajectory in the inset is shown as a red dashed line.

(B) Projecting attractor patterns onto three suitably chosen pairs of axes (see "attractor manifold analysis") reveals three rings, representing position along the 0°, 60°, and 120° vectors in real space. Colors correspond to decoded position in the 2d environment (insets).

(C and D) Model-free analyses for identifying toroidal structure.

(C) Persistent homology indicates that the attractor manifold has the topology of a torus. Left: Betti barcode for a synthetic torus with additive gaussian noise. Prominent bars (indicated by arrows) reflect the presence of one connected component (H^0), two 1-D holes (H^1), and one 2-D hole (H^2), which persist across scales. Right: the attractor manifold of our trained model has the same barcode, indicating toroidal structure. The four prominent bars are substantially longer than a null barcode (shaded region), defined as the longest lifetimes obtained over many shuffles of the data.

(D) Distance between attractor patterns at different spatial locations. One point is varied over the environment and the other is fixed at the black "x."

multiple times around the 2D attractor manifold. Figure 4B illustrates three different 2D projections, which show how the neural trajectory wraps multiple times around the torus as the animal moves either 0°, 60°, or 120° relative to the horizontal axis of the enclosure.

As a simple control, we repeated the analyses shown in Figures 4B and 4D on random patterns obtained by low-pass filtering spatial noise as in Figure 1D. The results, shown in Figures 1A and 1B, indicate a clear qualitative departure from a toroidal manifold structure, indicating that our novel toroidal subspace finding method only detects statistically robust tori and will not spuriously generate them when they do not exist, even in high dimensional datasets.

Next, we investigated the structure of the attractor manifold in the *full* activity space, rather than just the high-variance 6-D space, using two model-free approaches. First, we performed a persistent homology analysis (Zomorodian and Carlsson, 2005), which identifies topological holes of different dimensions in the data and produces a barcode characterizing its topology (see "persistent homology"). A perfect torus (Figure 4C, left) is characterized by a barcode with one connected component, two 1-D cycles (representing the two orthogonal ways of encircling the torus), and one 2-D hole (representing the 3-D volume enclosed by the torus). In Figure 4C, right, we compute the barcode for the attractor manifold in our trained model and find that it matches the barcode of a torus. This reproduces a similar topological analysis done on actual MEC grid cells (Gardner et al., 2022). Second, we note that a key property of the toroidal structure in 6-D space is that as the animal moves along the 0°, 60°, and 120° directions in physical space, motion along the attractor manifold curls back to itself to a very good approximation. We can similarly compute distances between different pairs of attractor patterns in the full space of all $N = 4096$ neurons, without relying on dimensionality reduction (Figure 4D). Each heatmap shows the neural activity distance when one environmental location is fixed at the black "x," and the other is varied over all environmental locations. These distances indicate that as the animal moves in physical space along the 0°, 60°, and 120° directions, the attractor manifold does indeed approximately, but not fully, return to itself multiple times in the full space of all neurons. Repeating this analysis on cells with a high grid score, the manifold returns much more closely to itself (Figure S1C), indicating that the partial departure from toroidal structure is due to heterogeneous neural activity patterns that coexist with more regular hexagonal grid patterns.

Thus, overall, trained neural networks, although containing some of the structure of the perfect toroidal attractor manifold that underlies hand-designed models, nevertheless, also contain within them a more general and varied 2*D* manifold structure that goes beyond the Platonic torus. This larger space of network solutions includes many neurons with highly regular grid patterns that *simultaneously* coexist with many neurons with more heterogeneous patterns (Figure S2), as is consistent with a recent statistical analysis of MEC firing patterns (Hardcastle et al., 2017). Interestingly, path-integrator networks with a simultaneous coexistence of neurons with both highly structured and highly heterogeneous firing patterns have been difficult to hand design. Thus, neural network training, which naturally finds such solutions, can generate circuit-level hypotheses for neural function with more biologically realistic levels of heterogeneity.

Mechanisms underlying path integration

We next turn our attention to how circuit connectivity and dynamics conspire to both generate the 2D attractor manifold and update position along this manifold as velocity signals enter the network. Below we review how traditional hand-designed models solve these two core problems. We find that at the level of individual neurons and synapses, the mechanisms of trained networks are much less readily apparent. We then develop and apply several analyses that reveal emergent, population-level mechanisms generalizing traditional mechanisms. In the supplement, we give an account of the simpler 1-dimensional version of this problem, where the results are easier to visualize and understand (see "analysis of 1D path integrator network mechanisms"; Methods S1 and S2).

Stable storage of positional information in two dimensions

We first study the problem of storing position in two dimensions. Traditional models accomplish this by arranging neurons on a two-dimensional neural sheet, with a center-surround connectivity over this sheet (Figure 5A), yielding a set of stable hexagonal firing patterns on the neural sheet (Figure 5B; simulation details in "idealized path integrator models"). To better understand how this tailored connectivity gives rise to periodic bump patterns, we first plotted the average connectivity as a function of displacement on the neural sheet, which clearly exhibits the center-surround structure built into the model (Figure 5C, left). To obtain an approximate, low-dimensional characterization of the connectivity, we computed its top eigenmodes, corresponding to activity patterns that are strongly amplified. The 6 top eigenmodes are plotted as heatmaps over the neural sheet (Figure 5D) and correspond simply to quadrature pairs of Fourier waves at 0°, 60°, and 120° along the neural sheet. Next, to characterize the full, high-dimensional behavior of the connectivity, we measured how a complete set of wave inputs of various frequencies and orientations are transformed over time (this approach is equivalent to Fourier analysis of the connectivity matrix; see "fourier analysis of recurrent weights"). The connectivity in hand-designed models transforms a single-wave input into a single-wave output of the *same* frequency and orientation, with a given amplification factor. These amplification factors are shown Figure 5C, right, demonstrating that only a small number of low frequency periodic patterns are amplified by the connectivity. In essence, a 2D family of stable patterns is maintained fundamentally through the self-excitation of low frequency Fourier modes on the 2D neural sheet (Figures 5A–5D).

A key impediment to applying the same analyses to the trained network lies in the absence of any organizing principle to sort the neurons onto a putative neural sheet and to understand the connectivity as a function of the relative position of pairs of neurons on the neural sheet. For a one-dimensional head-direction network, we show how to easily sort neurons by their preferred head direction to obtain a neural ring (Figure S4C). This simple strategy does not generalize to two dimensions because single-neuron firing rate maps are now multi-modal, and also, many are significantly heterogeneous. To address these issues, we developed a new, noise-tolerant sorting procedure tailored to hexagonal maps (Figure 5I). Briefly, for each rate map, we measured 3 spatial phases designed to locate the neuron's hexagonal grid along the 0°, 60°, and 120° axes of physical space. We then arranged the neurons onto a 2D neural sheet such that neurons with similar spatial phases were physically close (Figure 5E; see "sorting RNN units onto a neural line and a neural sheet" for full details).

After sorting the neurons, when we plot stable activity patterns of the trained network as patterns over the sorted 2D neural sheet, we obtain, remarkably, hexagonal firing patterns over the emergent neural sheet (Figure 5F). We note that these hexagonal firing patterns are distinct from, but related to, the firing fields of individual grid cells across physical space, as shown for example in Figure 2D. The latter involve the average firing rate of *single* cells across *all* of physical space, whereas the former involve the activity of *all* neurons across a neural sheet, although the animal is at a *single* point in physical space, analogous to a single frame in a Calcium imaging experiment (Gu et al., 2018).

After using activity to organize the neurons on a 2D neural sheet, we examined the neural connectivity. The insets in Figure 5G, showing three examples of outgoing connectivity of a single neuron as a function of displacement on the neural sheet, exhibit very little structure. However, when outgoing connectivity is averaged over all neurons, clear center-surround structure emerges (Figure 5G, left). We next extracted the top eigenvectors of the connectivity matrix. Although the eigenmodes did not form clear Fourier waves over the neural sheet, we found linear combinations of the top 10 eigenmodes (see "extracting Fourier modes from top connectivity eigenvectors" for details) that resemble the simple structure of the traditional model (Figures 5D and 5H). Moreover, the eigenvalues of the connectivity, shown in Figure 5J, reveal a small number of strong eigenmodes in the connectivity. Finally, replicating the wave analysis of Figure 5C, right, we found that the connectivity primarily amplifies a small number of wave inputs while preserving frequency and orientation (Figure 5G, right; see "fourier analysis of

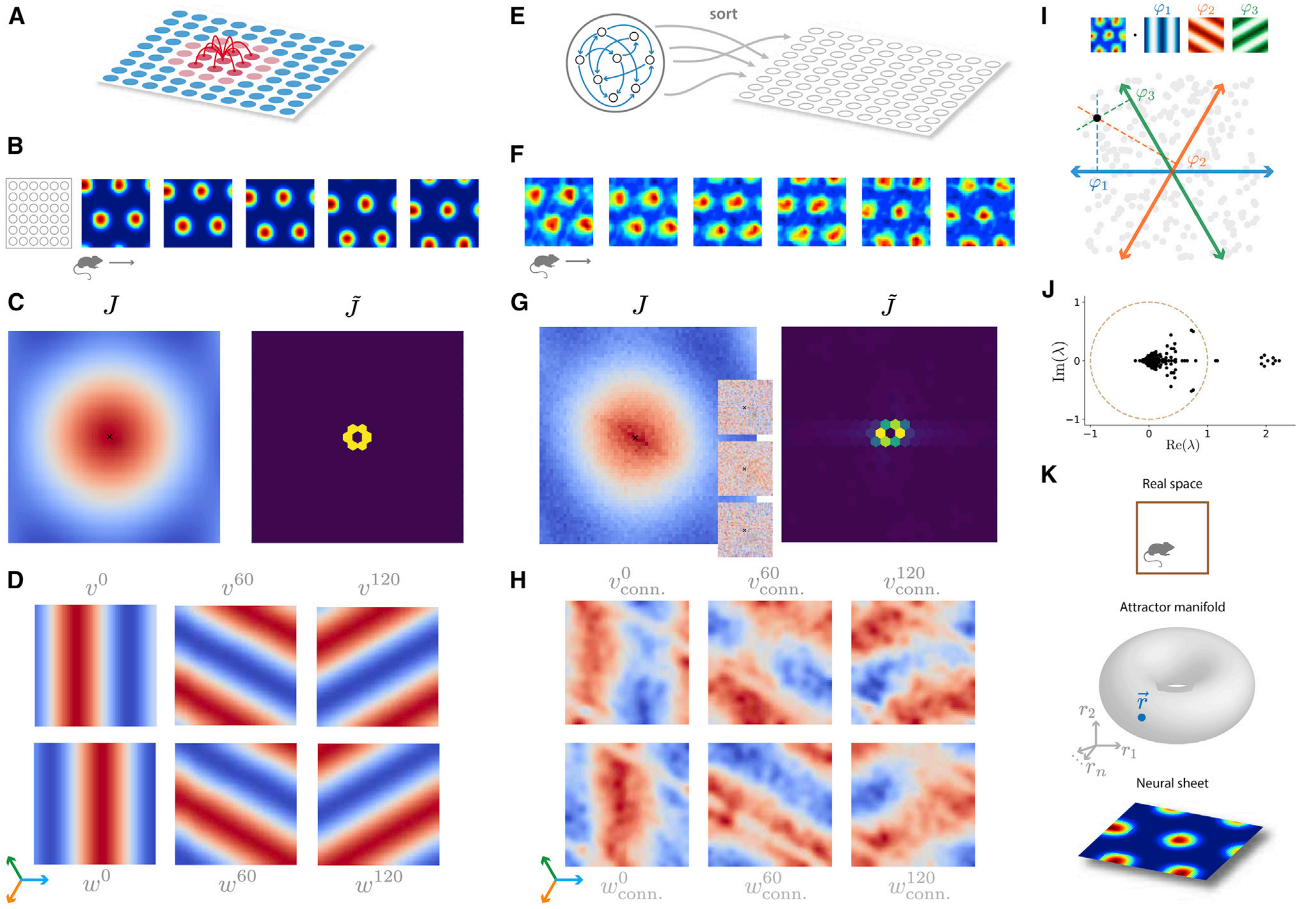

Figure 5. Mechanisms of information storage in two dimensions in hand-designed and trained neural networks

(A) Hand-designed networks employ a 2D sheet of neurons, with local excitation and long-range inhibition to yield stable activity patterns (simulation details in "idealized path integrator models").

(B) Stable activity patterns on the neural sheet when the animal is at 5 successive positions in physical space.

(C) The average outgoing connectivity profile (left; red (blue) indicates excitation (inhibition)), and the degree of self-excitation of Fourier modes over the 2D neural sheet (right). Fourier modes on the neural sheet are indexed by 2 discrete frequency variables, just as the Fourier modes in physical space Figures 3B and 3C. A small number of low frequency Fourier modes excite themselves.

(D) The top 6 eigenmodes of the connectivity correspond to low frequency Fourier modes on the 2D neural sheet.

(E) Schematic of our method to order neurons in a trained recurrent network along a 2D neural sheet.

(F–H) Replication of (B)–(D), now for the trained network using the extracted 2D neural sheet (see "sorting RNN units onto a neural line and a neural sheet" for sort details). The additional 3 insets in (G, left) show outgoing connectivity profiles for three neurons, revealing no discernible structure. Structure only appears after averaging outgoing connectivity profiles across neurons (G, left) or in the basis of Fourier modes (G, right), both as functions on our extracted 2D neural sheet.

(I) Schematic of method to position neurons on a 2D neural sheet. Each cell has a Fourier phase along 3 different axes. For hexagonal patterns, these phases obey a linear relation enabling us to explain them using only two variables, which then become coordinates on a neural sheet. For more heterogeneous cells, we find the best coordinates we can to explain the 3 phases, thereby placing all neurons (gray dots) at some point on a 2D neural sheet, revealing the emergent structure of trained networks in (F–H).

(J) Eigenvalues of the trained connectivity, indicating positive feedback for a small number of eigenmodes.

(K) Visualizing the storage of position in three different spaces. (Upper) The animal at a location in physical space; (middle) The current neural activity pattern corresponds to a point on a 2-D toroidal manifold in neural activity space. (Lower) The current neural activity as a pattern of hexagonal bumps on a 2D neural sheet.

recurrent weights"). Geometrically, the simple Fourier waves over the neural sheet sustained by the connectivity (Figures 5F and 5K, bottom) form a toroidal attractor manifold in activity space (middle), where each point represents a specific spatial location in the arena (top).

In summary, remarkably, the trained neural network finds a solution to the problem of storage of 2D position in a manner that is quite similar, at a collective level, to the hand-designed neural network, as seen by the qualitative similarity of Figures 5A–5H. This conclusion was not *a priori* obvious and required analysis methods that: (1) employ activity to organize neurons on a neural sheet and (2) find combinations of connectivity eigenmodes that behave simply as functions on the neural sheet. Notably, individual synaptic strengths have a much less clear meaning

(Figure 5G, insets). Instead, our ability to understand the essential principles underlying memory storage in this network required the combined analysis of both activity patterns and the connectome.

Velocity-based updating of positional information in two dimensions

We next asked how the trained RNN updates its stored location as it receives velocity inputs. See Figure S5 for an analysis of how this is done for a one-dimensional head direction system. In a two-dimensional path-integration circuit, hand-designed models accomplish velocity updating through multiple subpopulations of neurons with offset outgoing connections, biased in different directions on the neural sheet (Figure 6A). To update position correctly, each neuron's preferred velocity matches its biased pattern of outgoing connectivity. When the animal receives a northward velocity signal, for example, this leads to a sequence of steps. First, the velocity inputs activate cells that are selective for northward movement. These cells then transmit increased activation through their northward biased recurrent connectivity. Thus, cells on the north edge of the activity bump are excited, causing the activity bump to shift in this direction. This structure yields a situation in which animal motion in any direction in physical space moves the bump on the neural sheet in a corresponding direction. The precise combined effect of these steps can be understood by linearizing the dynamics, whereas the network is at a stable bump pattern and a velocity input is given to the network. This yields a quantitative expression for the activity pattern change $\Delta_i = \sum_j J_{ij}\sigma'_j(M_{jx}v_x + M_{jy}v_y)$ that combines the velocity v_x, v_y, velocity input weights M_{jx}, M_{jy}, neuron gain σ'_j, and the recurrent connectivity J_{ij} (see full derivation in "linearized RNN dynamics"). Consistent with the intuition given above, these update patterns Δ_i, shown in Figure 6B, excite (inhibit) the leading (trailing) edge of the activity bump and thus cause the activity pattern to shift in the appropriate direction. Geometrically, the dynamics of these hand-designed networks can be thought of as motion along a toroidal manifold of stable activity patterns that is pushed along this manifold in a velocity-dependent manner (Figure 6C), resulting in translation of a hexagonal pattern on the 2D neural sheet (Figure 6D).

We next asked whether trained networks learn the same structure in their feedforward and recurrent connectivity. Leveraging our ability to sort the neurons onto a 2D neural sheet using activity patterns alone (Figures 5E and 5I), we examined the histogram of biases in the outgoing connectivity of each neuron over the neural sheet. For a hand-designed network, this histogram would yield four populations of biases in outgoing projections (Figure 6E, left). However, in trained networks, we found that the histogram of biases was random and unimodal, centered around zero bias (Figure 6E, right; details in "connectivity bias"). Nevertheless, after sorting the units onto a neural sheet (Figure 6F), when we used the same linearization technique to compute the velocity-induced pattern of excitation and inhibition delivered across the neural sheet, given by $\Delta_i = \sum_j J_{ij}\sigma'_j(M_{jx}v_x + M_{jy}v_y)$, we found this pattern, obtainable only by summing over all neurons j in the network, behaved as in the hand-designed network, providing the correct excitation (inhibition) pattern to neurons at the leading (lagging) edge of the current activity bump along the correct axis on the neural sheet (Figure 6G). This yields a geometric picture in which the stable activity patterns form a toroidal manifold in the same 6 dimensional subspace of Figures 4B and 4D, and the pattern of excitation/inhibition Δ_i, when reduced to this space, pushes activity tangent to the manifold along a direction determined by the velocity in physical space (Figure 6H). This push moves the hexagonal pattern on the neural sheet in the correct direction (Figure 6I), and indeed, the cosine of the angle between the push is given by Δ_i, and the tangent direction of correct motion along the manifold stays roughly constant along the entire manifold of stable activity patterns (Figure 6J).

Thus, relative to a hand-designed model, the updating mechanism of the trained network remains murky at the level of individual synapses and neurons (compare Figure 6E, left and right), and it is only at a collective level that similar bump-pushing dynamics emerge (compare Figures 6B and 6G).

Investigating the role of heterogeneous cells in trained networks

Beyond the regular hexagonal grids predicted by our theory, our trained networks contain a number of units that exhibit more heterogeneous firing patterns, not unlike the diverse heterogeneous cells found in MEC (Hardcastle et al., 2017). These heterogeneous units develop late in training, after a large population of regular hexagonal grid cells have already emerged. We therefore sought to investigate whether the heterogeneous units play a key functional role in navigation and whether the heterogeneity in our trained models quantitatively matches the heterogeneity in MEC.

To address the first question, we ablated either the highest or lowest scoring grid cells in our model (see "analysis of heterogeneous cells"). We then evaluated the path integration performance of the ablated networks by measuring the distance between the location of the maximally activated place cell and the animal's true position at each time step. This mimics an experiment in an animal in which either grid cells or non-grid cells are selectivity knocked out, but the downstream circuitry from the remaining entorhinal cells to other cells mediating either behavior, or the animal's internal estimate of position, remain intact without modification. We found that performance degraded significantly when high grid score cells were ablated (Figure 7A, left), but not when low grid-score cells were ablated (Figure 7A, right), indicating that grid cells, but not heterogeneous cells, play a central role in path integration in our models.

What then is the role of heterogeneous cells? And why do they consistently emerge in trained networks? We hypothesized that although grid cells form the backbone of the path integrator, heterogeneous cells help solve the readout problem of transforming spatial information present in the path integrator to a place cell code. To test this hypothesis, we examined the ability of our model to generate the population code of place cells as we included increasing numbers of heterogeneous cells (Figure 7B, pink). We compared this with the readout performance of a population of pristine grid cells like those in a continuous attractor model (hexagonal grids with scale matched to RNN maps; Figure 7B, blue). We found that performance is similar across the two populations when only the most grid-like cells are included, but performance continues to improve as heterogeneous cells from our model are added, whereas performance quickly

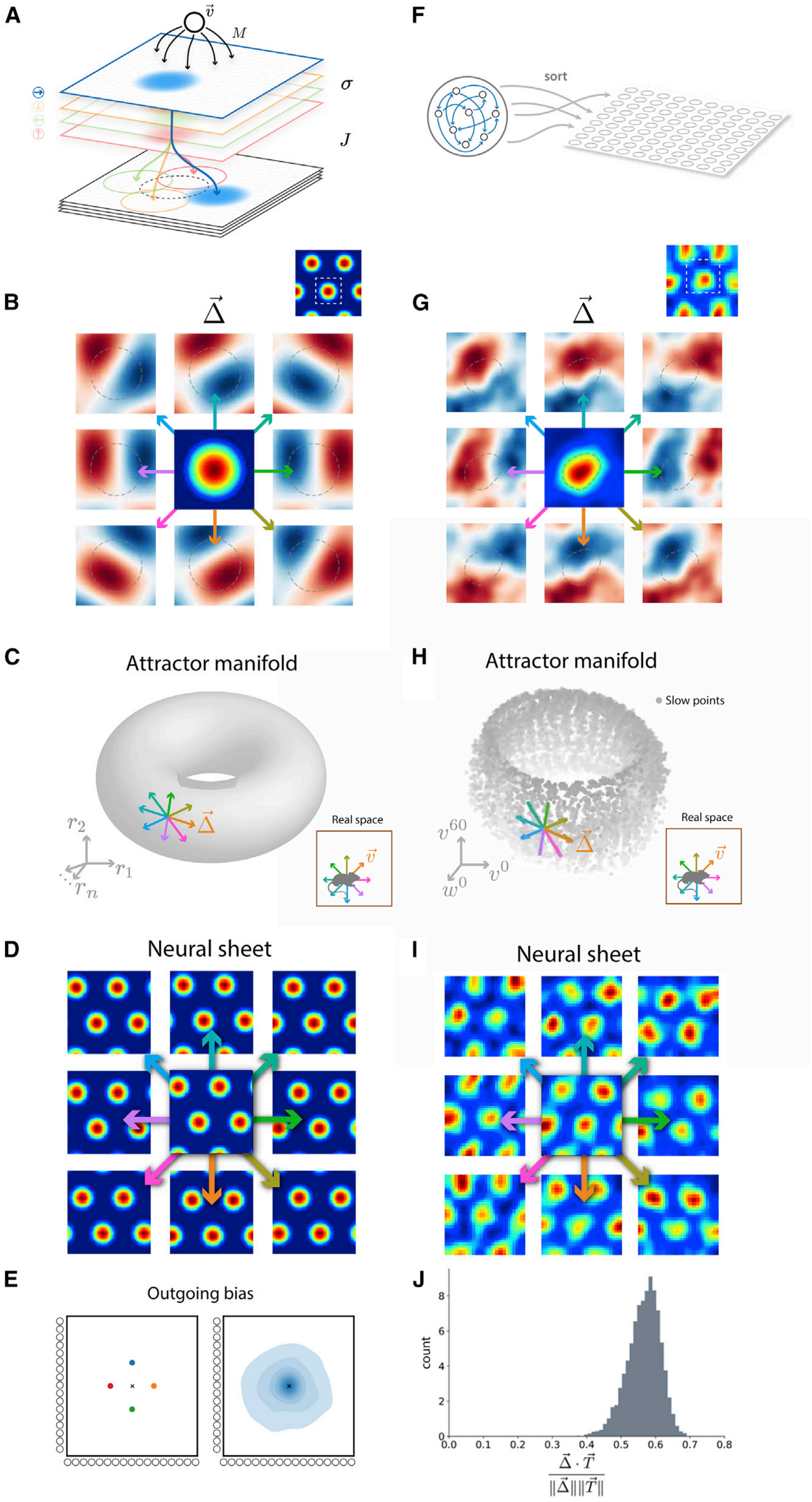

Figure 6. Mechanisms of 2-D path integration in hand-designed and trained neural networks

(A) Schematic of connectivity rule for updating position in many hand-designed models (full details in “idealized path integrator models”). Incoming velocity inputs $\vec{v}$ (black cell) with feedforward synapses M_j (black arrows) excite a population of eastward-projecting neurons (blue). The active eastward-projecting neurons (blue circle, top) excite neurons (blue circle, bottom) to the east of the current bump of activity (dashed-gray circle, bottom), through eastward-offset recurrent connectivity (blue arrow, J), shifting the bump of activity to the east. Analogous dynamics capture southward, westward, or northward motion (colored planes).

(B) The pattern of excitation (red) and inhibition (blue) Δ_i over the neural sheet when the animal moves in eight directions in physical space. The central pattern is a small section of the neural sheet before the motion (white-dashed box in inset).

(C) Motion of the animal in the environment (bottom right) corresponds to excitation/inhibition pattern Δ_i that are tangent along the 2D toroidal manifold of stable activity patterns.

(D) Motion of the hexagonal pattern on the 2D neural sheet for motions of the animal in physical space.

(E) Histogram of outgoing connectivity biases as a function of the relative displacement on the neural sheet, for hand-designed (left) and trained (right) networks (see “connectivity bias”).

(F) Schematic of sorting procedure (see Figure 5I and “sorting RNN units onto a neural line and a neural sheet”).

(G–I) Same as (B)–(D) for 2D neural sheet extracted from the trained network.

(J) Histogram of the cosine angle between the velocity-generated pattern of excitation inhibition Δ_i, and the correct direction of motion along the attractor manifold, across all points on the manifold. A roughly constant value (even if this value is not 1) is sufficient for accurate path integration.

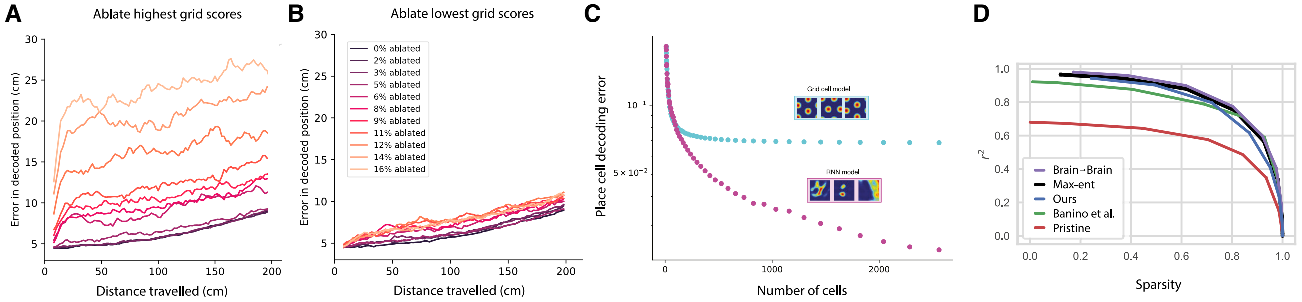

Figure 7. Investigating the role of heterogeneous cells in trained models

(A and B) (A) Ablating the highest-scoring grid cells during navigation significantly degrades spatial information within the network, whereas (B) ablating the heterogeneous cells has comparatively little effect, suggesting that grid cells play a more important role in path integration.

(C) On the other hand, heterogeneous cells significantly improve the network's ability to read out spatial information from the recurrent units into an output place-cell code. Cyan dots represent pristine grid maps with scale matched to the RNN; magenta dots represent trained RNN maps (ordered by grid score). For small numbers of regular grid cells, both sets perform similarly; as more heterogeneous cells are added to the RNN set, it significantly outperforms the pristine model.

(D) We compare the ability of our model (blue) and that of Banino et al. (2018) (green) to predict ratemaps of MEC neurons (from Butler et al., 2019), along with several controls: (1) a ceiling obtained by fitting one subset of the recorded data with another (purple), (2) a maximum entropy model with mean and covariance matched to MEC activity (black), and (3) a set of pristine grid maps as in continuous attractor models (Burak and Fiete, 2009) with several modules (red). L1 regularized fit performance is measured by cross-validated Pearson correlation (r^2) between actual and predicted ratemaps. Performance is plotted as the l1 regularization coefficient is varied (the x axis represents fraction of regression coefficients equal to zero). All heterogeneous models, including the maximum entropy model, roughly saturate the prediction ceiling set by brain-brain fitting, suggesting maps may be well characterized by second-order statistics alone.

saturates as increasing numbers of pristine grid cells are added (see "analysis of heterogeneous cells" for details). We obtained similar results when RNN maps/pristine grid maps were used to read out random value functions, rather than place cells (Figure S7; see "analysis of heterogeneous cells"). Together, Figures 7A and 7B suggest that grid cells form the core path integrator, whereas heterogeneous cells provide a richer repertoire of maps with which to readout downstream population codes such as place cells or arbitrary value functions.

Motivated by these observations, we next asked whether the structure of heterogeneity in our trained model might explain that of population codes in MEC. Using electrophysiology recordings from 778 MEC neurons in awake, behaving rats performing a free-foraging task (Butler et al., 2019), we evaluated the ability of our model neurons to predict the spatial rate maps of MEC neurons, including the heterogeneous non-grid cells, through linear regression under a sparsity penalty, using varying numbers of model rate maps as regressors (see "analysis of heterogeneous cells").

As an upper bound on the ability of any model to predict MEC activity, we first examined the ability of neurons in one animal to predict the activity of neurons in another animal (Figure 7C, purple). As a lower bound, we examined the ability of a population of pristine, regular hexagonal grids, constructed by hand across several spatial scales, to predict population activity in MEC (Figure 7C, red; see "analysis of heterogeneous cells"). We observed a significant gap between the upper and lower bounds, indicating that the heterogeneity in MEC contains structure, beyond that explained by regular hexagonal grids, and that this structure is preserved across animals. We next asked whether this structure is captured by the heterogeneity in our trained model. We found that its prediction performance nearly saturates the upper bound (Figure 7C, blue), indicating that the structure learned in our model captures nearly all of the heterogeneity in MEC. We additionally examined the performance of the model in Banino et al. (2018) and found that it is similar to ours, suggesting a match between MEC and model heterogeneity (despite the mismatch for high-scoring grid cells; Figure 1).

To better understand the heterogeneity that is preserved across animals, we also examined how well a maximum entropy model can predict MEC firing patterns. We constructed this maximum entropy model by creating a new population of model cells with random gaussian spatial firing patterns subject to the same first- and second-order statistics of actual MEC firing patterns (see "analysis of heterogeneous cells"). Thus, this population of cells has no statistical structure in its firing patterns above and beyond these MEC statistics and is otherwise essentially unstructured noise. Surprisingly, the maximum entropy model can predict actual MEC firing patterns in an animal (Figure 7C, black) essentially as well as MEC firing patterns in another animal can (Figure 7C, purple). Finally, in our trained model, we replicated a published analysis of heterogeneous cells in MEC (Hardcastle et al., 2017) and demonstrated that heterogeneous cells in our model, just like in data, lack any clear organizing structure along their top principal components (Figure S3).

Thus, overall, these results yield several insights into the nature of heterogeneous firing patterns in MEC: (1) at least by this prediction assay, they are unlikely to contain significant structure above and beyond their first- and second-order spatial statistics, (2) these spatial statistics are preserved across animals but further structure beyond that is unlikely to be preserved, and (3) our trained model, but not hand-designed continuous attractor networks, can account for almost all of this statistical structure in heterogeneous MEC firing patterns.

DISCUSSION

In summary, we addressed the questions raised in the introduction concerning the computational and mechanistic origins of grid cells across normative representational models and hand-tuned as well as trained path integrators. First, we obtain more robust hexagonal grid-like representations in neural networks trained to path integrate by introducing two simple biological constraints: nonnegative firing rates (inspired by Dordek et al., 2016) and a center-surround structure of inputs to place cells. Second, we demonstrate that our models not only path integrate with hexagonal grid cells in a square environment but also generalize this hexagonal pattern outside the original square, as this environment is expanded. Our networks thus go beyond prior trained networks, which neither yield robust hexagonal grids nor generalize in expanded environments. Third, we develop a general unified theory for *why* grid cells can spontaneously emerge in diverse normative as well as mechanistic models, including in recurrent networks trained to path integrate (Cueva and Wei, 2018; Banino et al., 2018) as well as feedforward networks trained to efficiently encode place cell inputs (Dordek et al., 2016). Fourth, across all these works, our general theory explains when and why different grid lattice structures (i.e., square, hexagonal, and heterogeneous) spontaneously emerge. Fifth, we develop novel algorithmic methods to extract, from the seemingly highly unstructured connectomes of such trained networks, a conceptual understanding of how they both path integrate as well as mechanistically generate hexagonal grid-cell responses. Sixth, we relate our conceptual understanding of the circuit mechanisms underlying trained grid-cell path integrators to those of hand-tuned path integrators, showing that the former obeys similar computational principles as the latter, but these principles only emerge at a collective level of analysis and are hard to discern from the properties of single neurons and synapses. Seventh, we uncovered a functional dichotomy between grid cells and heterogeneous cells in our model, with the former primarily contributing to path integration and the latter primarily contributing to the construction of diverse spatial functions of positions. Finally, we quantitatively matched the heterogeneous firing patterns in our model to those of actual MEC neurons, finding that our model could predict the structure of this heterogeneity almost as well as neurons in another animal could. These results address and raise a host of interesting issues.

Limitations and future directions

In this work, we have worked backward from function to structure by showing that neural networks trained to path integrate, under two additional simple constraints, yield the structure of grid cells. (Of course, the converse, namely that the mere existence of grid cells by themselves implies the task of path integration, is not true: indeed, the model of Dordek et al. (2016) provides a counterexample). It would be interesting to study how additional normative criteria could yield more detailed predictions about MEC recurrent connectivity, such as its purely inhibitory nature (Couey et al., 2013), as the attractor framework alone cannot predict synapse polarity since both polarities can yield similarly functioning networks (Fuhs and Touretzky, 2006; Burak and Fiete, 2009). These extensions could help shed light on the precise relation between biological grid cells and path integration, which remains a challenging open question, due to the experimental difficulty of specifically perturbing grid cells (Gil et al., 2018).

Another interesting extension would be to incorporate landmarks and/or other movement or direction signals such as head direction, in addition to the body velocity inputs we consider here, as additional sources of information about position. The fusion of landmark and velocity inputs has been previously studied in hand-designed models, which have successfully accounted for the deformation of grid cells in irregular environments (Ocko et al., 2018a) and the remapping of grid cells in virtual reality environments (Campbell et al., 2018). However, such models, with their crystalline grid-cell structure, cannot make predictions for what heterogeneous cells would do under the same experimental manipulations. Intriguingly, non-grid spatial cells have been shown to remap more readily to environmental manipulations than grid cells (Diehl et al., 2017).

Also, recent work has shown that changes in rewarded location can affect grid-cell firing properties (Butler et al., 2019; Boccara et al., 2019). Training neural networks to forage in response to changed rewarded locations could yield new, general hypotheses about interactions between reward, spatial location, and grid-cell and heterogeneous-cell firing patterns. We note that our model is trained using a place cell-like code as a teacher signal (although our theory reveals that *any* set of stable spatial responses with the same spatial correlational structure as our place cell code will produce grid cells; see "pattern formation theory predicts structure of learned representations"). This teaching process is simply a method for exploring the space of path-integrator models and is not meant to model the actual biological development of grid cells. A future direction would be to investigate whether a more general framework can promote the simultaneous emergence of grid and place cells within a single model.

Perhaps more ambitiously, it would be exciting to explore the functional role of MEC grid cells in more general settings beyond that of path integration, for example, in general episodic memory, abstract reasoning, planning, and imagination (Moser and Moser, 2013; Constantinescu et al., 2016; Bellmund et al., 2016; Whittington et al., 2020). Indeed, ancient mechanisms for integrating velocity to arrive at position, in path integration, may have been co-opted by evolution to integrate individual deductive steps to arrive at final inferences, in general processes of reasoning and imagination. To obtain mechanistic hypotheses as to how neural circuits can accomplish such remarkable feats, the proximal path may lie in training them on complex tasks, developing analytic tools to understand their function and extracting predictions that can be tested at emergent levels of circuit organization—perhaps beyond individual neurons and synapses—in large-scale connectomes and brain activity maps. We hope our work along these lines in the simple setting of path integration inspires similar ways forward in more complex settings.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.neuron.2022.10.003.

ACKNOWLEDGMENTS

We would like to thank Shaul Druckmann for helpful discussions. We also thank the following for financial support: the Stanford Graduate Fellowship for B.S.; the Stanford Neurosciences Graduate program for G.C.M.; an Urbanek fellowship for S.A.O.; the Office of Naval Research (N00141812690), Simons Foundation (SCGB 542987SPI), National Institute of Mental Health (NIMH MH106475), and the James S. McDonnell Foundation for L.M.G.; and the Simons and James S. McDonnell foundations and an NSF CAREER award for S.G.

AUTHOR CONTRIBUTIONS

All authors participated in the design of experiments and analyses. B.S. and G.C.M. performed all network experiments. B.S., G.C.M., and S.A.O. performed mathematical analyses. All authors contributed to writing the paper.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: November 25, 2020
Revised: May 5, 2022
Accepted: October 3, 2022
Published: October 27, 2022

REFERENCES

Aronov, D., Nevers, R., and Tank, D.W. (2017). Mapping of a non-spatial dimension by the hippocampal-entorhinal circuit. Nature *543*, 719–722. https://doi.org/10.1038/nature21692.

Banino, A., Barry, C., Uria, B., Blundell, C., Lillicrap, T., Mirowski, P., Pritzel, A., Chadwick, M.J., Degris, T., Modayil, J., et al. (2018). Vector-based navigation using grid-like representations in artificial agents. Nature *557*, 429–433. https://doi.org/10.1038/s41586-018-0102-6.

Bellmund, J.L., Deuker, L., Navarro Schröder, T., and Doeller, C.F. (2016). Grid-cell representations in mental simulation. eLife *5*, e17089. https://doi.org/10.7554/eLife.17089.

Ben-Yishai, R., Bar-Or, R.L., and Sompolinsky, H. (1995). Theory of orientation tuning in visual cortex. Proc. Natl. Acad. Sci. USA *92*, 3844–3848. https://doi.org/10.1073/pnas.92.9.3844.

Blair, H. (1996). Simulation of a thalamocortical circuit for computing directional heading in the rat. In Advances in Neural Information Processing Systems 8, D.S. Touretzky, M.C. Mozer, and M.E. Hasselmo, eds. (MIT Press), pp. 152–158.

Boccara, C.N., Nardin, M., Stella, F., O'Neill, J., and Csicsvari, J. (2019). The entorhinal cognitive map is attracted to goals. Science *363*, 1443–1447. https://doi.org/10.1126/science.aav4837.

Burak, Y., and Fiete, I.R. (2009). Accurate path integration in continuous attractor network models of grid cells. PLoS Comput. Biol. *5*, e1000291. https://doi.org/10.1371/journal.pcbi.1000291.

Butler, W.N., Hardcastle, K., and Giocomo, L.M. (2019). Remembered reward locations restructure entorhinal spatial maps. Science *363*, 1447–1452. https://doi.org/10.1126/science.aav5297.

Campbell, M.G., Ocko, S.A., Mallory, C.S., Low, I.I.C., Ganguli, S., and Giocomo, L.M. (2018). Principles governing the integration of landmark and self-motion cues in entorhinal cortical codes for navigation. Nat. Neurosci. *21*, 1096–1106. https://doi.org/10.1038/s41593-018-0189-y.

Conklin, J., and Eliasmith, C. (2005). A controlled attractor network model of path integration in the rat. J. Comput. Neurosci. *18*, 183–203. https://doi.org/10.1007/s10827-005-6558-z.

Constantinescu, A.O., O'Reilly, J.X., and Behrens, T.E.J. (2016). Organizing conceptual knowledge in humans with a gridlike code. Science *352*, 1464–1468. https://doi.org/10.1126/science.aaf0941.

Couey, J.J., Witoelar, A., Zhang, S.J., Zheng, K., Ye, J., Dunn, B., Czajkowski, R., Moser, M.B., Moser, E.I., Roudi, Y., and Witter, M.P. (2013). Recurrent inhibitory circuitry as a mechanism for grid formation. Nat. Neurosci. *16*, 318–324. https://doi.org/10.1038/nn.3310.

Cross, M., and Greenside, H. (2009). Pattern Formation and Dynamics in Nonequilibrium Systems (Cambridge University Press).

Cueva, C.J., and Wei, X.-X. (2018). Emergence of grid-like representations by training recurrent neural networks to perform spatial localization. Int. Conf. Learn. Representat. https://doi.org/10.48550/arXiv.1803.07770.

Diehl, G.W., Hon, O.J., Leutgeb, S., and Leutgeb, J.K. (2017). Grid and nongrid cells in medial entorhinal cortex represent spatial location and environmental features with complementary coding schemes. Neuron *94*, 83–92.e6. https://doi.org/10.1016/j.neuron.2017.03.004.

Doeller, C.F., Barry, C., and Burgess, N. (2010). Evidence for grid cells in a human memory network. Nature *463*, 657–661. https://doi.org/10.1038/nature08704.

Dordek, Y., Soudry, D., Meir, R., and Derdikman, D. (2016). Extracting grid cell characteristics from place cell inputs using non-negative principal component analysis. eLife *5*, e10094. https://doi.org/10.7554/eLife.10094.

Fuhs, M.C., and Touretzky, D.S. (2006). A spin glass model of path integration in rat medial entorhinal cortex. J. Neurosci. *26*, 4266–4276. https://doi.org/10.1523/JNEUROSCI.4353-05.2006.

Fyhn, M., Hafting, T., Witter, M.P., Moser, E.I., and Moser, M.-B. (2008). Grid cells in mice. Hippocampus *18*, 1230–1238. https://doi.org/10.1002/hipo.20472.

Gao, P., and Ganguli, S. (2015). On simplicity and complexity in the brave new world of large-scale neuroscience. Curr. Opin. Neurobiol. *32*, 148–155. https://doi.org/10.1016/j.conb.2015.04.003.

Gardner, R.J., Hermansen, E., Pachitariu, M., Burak, Y., Baas, N.A., Dunn, B.A., Moser, M.-B., and Moser, E.I. (2022). Toroidal topology of population activity in grid cells. Nature *602*, 123–128. https://doi.org/10.1038/s41586-021-04268-7.

Gerlei, K., Passlack, J., Hawes, I., Vandrey, B., Stevens, H., Papastathopoulos, I., and Nolan, M.F. (2020). Grid cells are modulated by local head direction. Nat. Commun. *11*, 4228. https://doi.org/10.1038/s41467-020-17500-1.

Gil, M., Ancau, M., Schlesiger, M.I., Neitz, A., Allen, K., De Marco, R.J., and Monyer, H. (2018). Impaired path integration in mice with disrupted grid cell firing. Nat. Neurosci. *21*, 81–91. https://doi.org/10.1038/s41593-017-0039-3.

Gu, Y., Lewallen, S., Kinkhabwala, A.A., Domnisoru, C., Yoon, K., Gauthier, J.L., Fiete, I.R., and Tank, D.W. (2018). A map-like micro-organization of grid cells in the medial entorhinal cortex. Cell *175*, 736–750.e30. https://doi.org/10.1016/j.cell.2018.08.066.

Guanella, A., Kiper, D., and Verschure, P. (2007). A model of grid cells based on a twisted torus topology. Int. J. Neural Syst. *17*, 231–240. https://doi.org/10.1142/S0129065707001093.

Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., and Moser, E.I. (2005). Microstructure of a spatial map in the entorhinal cortex. Nature *436*, 801–806. https://doi.org/10.1038/nature03721.

Hahnloser, R.H. (2003). Emergence of neural integration in the head-direction system by visual supervision. Neuroscience *120*, 877–891. https://doi.org/10.1016/s0306-4522(03)00201-x.

Hardcastle, K., Maheswaranathan, N., Ganguli, S., and Giocomo, L.M. (2017). A multiplexed, heterogeneous, and adaptive code for navigation in medial entorhinal cortex. Neuron *94*, 375–387.e7. https://doi.org/10.1016/j.neuron.2017.03.025.

Insel, T.R., Landis, S.C., and Collins, F.S. (2013). The NIH BRAIN initiative. Science *340*, 687–688. https://doi.org/10.1126/science.1239276.

Killian, N.J., Jutras, M.J., and Buffalo, E.A. (2012). A map of visual space in the primate entorhinal cortex. Nature *491*, 761–764. https://doi.org/10.1038/nature11587.

Langston, R.F., Ainge, J.A., Couey, J.J., Canto, C.B., Bjerknes, T.L., Witter, M.P., Moser, E.I., and Moser, M.-B. (2010). Development of the spatial representation system in the rat. Science *328*, 1576–1580. https://doi.org/10.1126/science.1188210.

Lindsey, J., Ocko, S.A., Ganguli, S., and Deny, S. (2019). A unified theory of early visual representations from retina to cortex through anatomically constrained deep CNNs. Int. Conf. Learn. Representat. https://doi.org/10.48550/arXiv.1901.00945.

Mante, V., Sussillo, D., Shenoy, K.V., and Newsome, W.T. (2013). Context-dependent computation by recurrent dynamics in prefrontal cortex. Nature *503*, 78–84. https://doi.org/10.1038/nature12742.

Mathis, A., Stemmler, M.B., and Herz, A.V. (2015). Probable nature of higher-dimensional symmetries underlying mammalian grid-cell activity patterns. eLife *4*, e05979. https://doi.org/10.7554/eLife.05979.

McIntosh, L.T., Maheswaranathan, N., Nayebi, A., Ganguli, S., and Baccus, S.A. (2016). Deep learning models of the retinal response to natural scenes. Adv. Neural Inf. Process. Syst. *29*, 1369–1377.

McNaughton, B.L., Barnes, C.A., Gerrard, J.L., Gothard, K., Jung, M.W., Knierim, J.J., Kudrimoti, H., Qin, Y., Skaggs, W.E., Suster, M., and Weaver, K.L. (1996). Deciphering the hippocampal polyglot: the hippocampus as a path integration system. J. Exp. Biol. *199*, 173–185. https://doi.org/10.1242/jeb.199.1.173.

Miller, K.D., Keller, J.B., and Stryker, M.P. (1989). Ocular dominance column development: analysis and simulation. Science *245*, 605–615. https://doi.org/10.1126/science.2762813.

Moser, E.I., and Moser, M.-B. (2013). Grid cells and neural coding in high-end cortices. Neuron *80*, 765–774. https://doi.org/10.1016/j.neuron.2013.09.043.

Ocko, S., Lindsey, J., Ganguli, S., and Deny, S. (2018b). The emergence of multiple retinal cell types through efficient coding of natural movies. In Advances in Neural Information Processing Systems 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds. (Curran Associates, Inc.), pp. 9389–9400.

Ocko, S.A., Hardcastle, K., Giocomo, L.M., and Ganguli, S. (2018a). Emergent elasticity in the neural code for space. Proc. Natl. Acad. Sci. USA *115*, E11798–E11806. https://doi.org/10.1073/pnas.1805959115.

Raudies, F., Brandon, M.P., Chapman, G.W., and Hasselmo, M.E. (2015). Head direction is coded more strongly than movement direction in a population of entorhinal neurons. Brain Res. *1621*, 355–367. https://doi.org/10.1016/j.brainres.2014.10.053.

Raudies, F., and Hasselmo, M.E. (2012). Modeling boundary vector cell firing given optic flow as a cue. PLoS Comput. Biol. *8*, e1002553. https://doi.org/10.1371/journal.pcbi.1002553.

Redish, A.D., Elga, A.N., and Touretzky, D.S. (1996). A coupled attractor model of the rodent head direction system. Network *7*, 671–685. https://doi.org/10.1088/0954-898X_7_4_004.

Samsonovich, A., and McNaughton, B.L. (1997). Path integration and cognitive mapping in a continuous attractor neural network model. J. Neurosci. *17*, 5900–5920. https://doi.org/10.1523/JNEUROSCI.17-15-05900.1997.

Savelli, F., Yoganarasimha, D., and Knierim, J.J. (2008). Influence of boundary removal on the spatial representations of the medial entorhinal cortex. Hippocampus *18*, 1270–1282. https://doi.org/10.1002/hipo.20511.

Saxe, A.M., McClelland, J.L., and Ganguli, S. (2014). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. Int. Conf. Learn. Representat. https://doi.org/10.48550/arXiv.1312.6120.

Saxe, A.M., McClelland, J.L., and Ganguli, S. (2018). A mathematical theory of semantic development in deep neural networks. Proc. Natl. Acad. Sci. USA *116*, 11537–11546. https://doi.org/10.1073/pnas.1820226116.

Seung, H.S. (2009). Reading the book of memory: sparse sampling versus dense mapping of connectomes. Neuron *62*, 17–29. https://doi.org/10.1016/j.neuron.2009.03.020.

Skaggs, W., Knierim, J., Kudrimoti, H., and McNaughton, B. (1994). A model of the neural basis of the rat's sense of direction. Adv. Neural Inf. Process. Syst. *7*, 173–180.

Sorscher, B., Mel, G., Ganguli, S., and Ocko, S. (2019). A unified theory for the origin of grid cells through the lens of pattern formation. Adv. Neural Inf. Process. Syst. *32*, 10003–10013.

Stachenfeld, K.L., Botvinick, M.M., and Gershman, S.J. (2017). The hippocampus as a predictive map. Nat. Neurosci. *20*, 1643–1653. https://doi.org/10.1038/nn.4650.

Sussillo, D., and Barak, O. (2013). Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. Neural Comput. *25*, 626–649. https://doi.org/10.1162/NECO_a_00409.

Sussillo, D., Churchland, M.M., Kaufman, M.T., and Shenoy, K.V. (2015). A neural network that finds a naturalistic solution for the production of muscle activity. Nat. Neurosci. *18*, 1025–1033. https://doi.org/10.1038/nn.4042.

Tanaka, H., Nayebi, A., Maheswaranathan, N., McIntosh, L., Baccus, S.A., and Ganguli, S. (2019). From deep learning to mechanistic understanding in neuroscience: the structure of retinal prediction. Adv. Neural Inf. Process. Syst. *32*, 8537–8547.

Wei, X.-X., Prentice, J., and Balasubramanian, V. (2015). A principle of economy predicts the functional architecture of grid cells. eLife *4*, e08362. https://doi.org/10.7554/eLife.08362.

Whittington, J., Muller, T., Mark, S., Barry, C., and Behrens, T. (2018). Generalisation of structural knowledge in the hippocampal-entorhinal system. Adv. Neural Inf. Process. Syst. *31*, 8484–8495.

Whittington, J.C.R., Muller, T.H., Mark, S., Chen, G., Barry, C., Burgess, N., and Behrens, T.E.J. (2020). The Tolman-Eichenbaum machine: unifying space and relational memory through generalization in the hippocampal formation. Cell *183*, 1249–1263.e23. https://doi.org/10.1016/j.cell.2020.10.024.

Winterer, J., Maier, N., Wozny, C., Beed, P., Breustedt, J., Evangelista, R., Peng, Y., D'Albis, T., Kempter, R., and Schmitz, D. (2017). Excitatory microcircuits within superficial layers of the medial entorhinal cortex. Cell Rep. *19*, 1110–1116. https://doi.org/10.1016/j.celrep.2017.04.041.

Yamins, D.L., and DiCarlo, J.J. (2016). Using goal-driven deep learning models to understand sensory cortex. Nat. Neurosci. *19*, 356–365. https://doi.org/10.1038/nn.4244.

Yamins, D.L., Hong, H., Cadieu, C.F., Solomon, E.A., Seibert, D., and DiCarlo, J.J. (2014). Performanceoptimized hierarchical models predict neural responses in higher visual cortex. Proc. Natl. Acad. Sci. USA *111*, 8619–8624. https://doi.org/10.1073/pnas.1403112111.

Yartsev, M.M., Witter, M.P., and Ulanovsky, N. (2011). Grid cells without theta oscillations in the entorhinal cortex of bats. Nature *479*, 103–107. https://doi.org/10.1038/nature10583.

Zhang, K. (1996). Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. J. Neurosci. *16*, 2112–2126. https://doi.org/10.1523/JNEUROSCI.16-06-02112.1996.

Zomorodian, A., and Carlsson, G. (2005). Computing persistent homology. Discrete Comput. Geom. *33*, 249–274. https://doi.org/10.1007/s00454-004-1146-y.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited Data		
Giocomo lab grid dataset	Lab of Lisa Giocomo.	FigShare: https://doi.org/10.25452/figshare.plus.15041316
Gardner et al. grid cell activity maps	Gardner et al. (2022) manuscript.	https://doi.org/10.1038/s41586-021-04268-7
Banino et al. network activity maps.	Banino et al. (2018) manuscript.	https://doi.org/10.1038/s41586-018-0102-6
Software and Algorithms		
Custom network training/analysis code	Code was written by the authors. Available at https://github.com/ganguli-lab/grid-pattern-formation	Github: https://doi.org/10.5281/zenodo.7110765

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Gabriel Mel (meldefon@stanford.edu).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- This paper analyzes existing, publicly available data. The accession numbers for the datasets are listed in the key resources table.
- All original code has been deposited at https://github.com/ganguli-lab/grid-pattern-formation and is publicly available as of the date of publication. DOIs are listed in the key resources table.
- Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

NOTATION

n_g	number of hidden recurrent neurons
n_p	number of output place cells
n_x	number of spatial arena locations
$\vec{x}$	position in 2d physical space (arena)
$\vec{k}^a$	unit vectors in $a = 0^\circ, 60^\circ, 120^\circ$ directions.
$\vec{s}_i$	position of neuron i on neural sheet
r_i^t	activity of neuron i at time t
$r_i(\vec{x})$	activity of neuron i at arena location $\vec{x}$
φ_i^a	measured phase of neuron i's spatial rate map in $\vec{k}^a$ direction
$\widehat{\varphi}_i^a$	ideal phase of neuron i's spatial rate map in $\vec{k}^a$ direction
β_i	outgoing connectivity bias of neuron i

RNN TRAINING

Path integration task

The task and training protocol shown in 1 were replicated from Banino et al. (2018). Place cell receptive field centers $\vec{c}_i(n_p = 512)$ were distributed randomly over a $(2.2m \times 2.2m)$ environment. The response of the i^{th} place cell was simulated using a difference of softmax tuning curve, similar to a difference of gaussians, but with a slightly different normalization to match the tuning curves in Banino et al. (2018), $p_i(\vec{x}) = e^{-\|\vec{x}-\vec{c}_i\|^2/2\sigma_1^2} / \sum_{j=1}^{n_p} e^{-\|\vec{x}-\vec{c}_j\|^2/2\sigma_1^2} - e^{-\|\vec{x}-\vec{c}_i\|^2/2\sigma_2^2} / \sum_{j=1}^{n_p} e^{-\|\vec{x}-\vec{c}_j\|^2/2\sigma_2^2}$ where x is the current location of the agent, and σ_1 and σ_2 represent the width of the center and surround, respectively. Agent trajectories were generated using the rat motion model described in Raudies and Hasselmo (2012). Velocity signals from the simulated trajectory were given as input, and the network was expected to produce the simulated place cell activities as output.

Network architecture

The trained RNN studied throughout this work (cf. sections "two dimensional attractor dynamics underlies path integration in trained networks" and "mechanisms underlying path integration") is a "vanilla" RNN with the familiar discrete-time dynamics used in conventional ring attractor networks:

$$r_i^{t+1} = \sigma\left[\sum_{j=1}^{n_g} J_{ij} r_j^t + M_{ix} v_x^t + M_{iy} v_y^t\right] \quad \text{(Equation 2)}$$

where r^t is the population activity at time t, J is the (n_g, n_g) recurrent connectivity matrix, $\vec{v}^t$ is the 2-dimensional velocity input at time t, M is the $(n_g, 2)$ matrix of velocity input weights, and σ is a pointwise nonlinearity (either tanh or relu). Predicted place cell outputs $\hat{p}^t$ are read out linearly by a (n_p, n_g) matrix of weights W:

$$\hat{p}_i^t = \sum_{j=1}^{n_g} W_{ij} r_j^t \quad \text{(Equation 3)}$$

Rather than tuning the weights J, M, W by hand, we allowed them to be trained by gradient descent on the objective of reconstructing the true place cell outputs p^t as accurately as possible. The loss function we used for training was a cross-entropy loss. Task, architecture and training hyperparameters are collected in the table below:

Task	
Arena size	(2.2m x 2.2m)
Average agent speed	0.1 m/sec
n_p(# place cells)	512
place cell σ_1, σ_2	20cm, 40cm
Architecture	
n_g(# RNN units)	4096
Input	(v_x, v_y)
Training	
Path length	20
Batch size	200
Number of batches	10000
Optimizer	RMSProp
Learning rate	1e-4
l2 regularization	1e-4

1d RNN

Head direction (HD) networks (Methods S1 and S2; Figures S4 and S5) were trained on a simplified 1D circular version of the spatial navigation task (described below). We simulated 32 head direction cells with preferred HD evenly spaced over 360 degrees. HD cell responses as a function of true head direction were computed as $h_i(\theta) \propto \exp\left(\kappa \cos(\theta - \theta_i)\right)$ with $\kappa = 2$ and θ_i is the preferred head direction of HD cell i. Agent heading trajectories were generated by first sampling turning velocity at each step from a Gaussian distribution with $\mu = 0$ and $\sigma = 0.8$ radians. Velocity trajectories were then smoothed over time with a gaussian filter with $\sigma =$ 1 step, and integrated to give head direction as a function of time. The network was given the trajectory velocities as input, and was expected to produce the responses of the 32 simulated HD cells as output. We used the same "vanilla" RNN architecture as for the 2D integration task, but with $n_g = 512$ hidden neurons rather than 4096.

1-layer neural network

The 1-layer neural network shown in Figure 2B solves the problem of optimally reconstructing place cell activities P from k encoding maps G using readout weights W:

$$\min_{W,G}\|P - WG\|^2. \quad \text{(Equation 4)}$$

Without any additional constraints, this just amounts to low rank matrix factorization. By the Eckart-Young-Mirsky theorem, the optimal low rank WG has the same left and right singular vectors as P, and its singular values are obtained from P's singular values by keeping the top k values and zeroing all others. This gives a particular W and G. All other optimal solutions can be obtained via the transformation $W \rightarrow WM$, and $G \rightarrow M^{-1}G$ for some invertible matrix M. The optimal maps G shown Figure 2B, left were obtained following the above by computing the singular value decomposition of P and extracting the top $k = 9$ left singular vectors.

With positivity constraints on W, and G, the problem becomes nonnegative matrix factorization. We used a standard toolbox for nonnegative matrix factorization (scikit-learn) for $k = 9$ maps. The resulting maps G are shown in Figure 2D, left.

LSTM path integrator network

The task and training protocol were identical to that of the RNN described above. The model architecture shown in Figure 2 was replicated from Banino et al. (2018), consisting of x- and y-velocity inputs to an LSTM with 512 hidden units, followed by a linear layer of 128 units (which the authors called the "g-layer"), followed by a final readout to the estimated place cell activities. We trained both with and without an additional output to a population of head direction cells, as the authors in Banino et al. (2018) did, and obtained similar results. We report results for the network that was trained to predict only place cell outputs. A summary of the task, architectural, and training parameters is given below:

Task	
Arena size	(2.2m x 2.2m)
Average agent speed	0.1 m/sec
n_p(# place cells)	512
place cell σ_1, σ_2	20cm, 40cm
Architecture	
n_g(# LSTM units)	512
# g-layer units	128
Input	(v_x, v_y)
Training	
Path length	50
Batch size	200
Number of batches	10000
Optimizer	RMSProp
Learning rate	1e-4

We found that training the model in Banino et al. (2018) with the following set of hyperparameters produced square grids (Figure 2B), similar to those found in Cueva and Wei (2018),

Task	
Arena size	(2.2m x 2.2m)
Average agent speed	0.1 m/sec
n_p(# place cells)	256
place cell σ_1	5cm
Architecture	
n_g(# LSTM units)	128
nonlinearity	tanh
# g-layer units	512
Input	(v_x, v_y)
Training	
Path length	100
Batch size	100
Number of batches	1500
Optimizer	Adam
Learning rate	1e-2
l2 regularization	1e-5

The square maps in the left panel of Figure 2B were obtained by optimizing a 1-layer network, as described above, using these simulated trajectories and place cell outputs.

Biological constraints

We imposed the following biologically inspired constraints on the RNN analyzed throughout the paper (sections "two dimensional attractor dynamics underlies path integration in trained networks" and "mechanisms underlying path integration") Nonnegativity: In order to achieve regular hexagonal grids, as our theory predicts, we imposed a nonnegativity constraint on the activities of the recurrent units in the RNN. We imposed this constraint by simply swapping the tanh nonlinearity for a relu (cf. Figure 1C), though we found that softer versions of this constraint, such as adding a penalty on negative firing rates to the loss function, also achieved the same effect.

Weight decay: We found that a small penalty $\mathcal{L}_W = \alpha\sum_{i,j=1}^{n_g} J_{ij}^2$ on the magnitudes of the recurrent weights encouraged a representation that generalizes beyond the boundaries of the training environment (Figure 1F). We simply added this penalty to the loss function for $\alpha = 10^{-5}$.

GRID SCORE

Grid score was evaluated as in Banino et al. (2018). A spatial ratemap was computed for each neuron by binning the agent's position into 2cm× 2cm bins, and computing the average firing rate within each bin. Grid score was evaluated by rotating a circular sample of the spatial autocorrelogram of this ratemap in steps of 30°, and computing the correlation between the rotated map and the original. The grid score was defined as the minimum difference between the correlation at the expected peaks, (60°,120°), and the correlation at the expected troughs (30°,90°,150°). The distribution of grid scores for all hidden units in the sigmoidal network with unconstrained firing rates, the null model, and the ReLU network with nonnegative firing rates are shown in Figure 1E.

Noise model

The null model used for comparison consisted of low-pass filtered noise maps. Gaussian random coefficients for all frequencies between 0 cycles/pixel and 0.0625 cycles/pixel were generated, and then inverse Fourier transformed, giving rise to random maps with frequency content in the desired range (this is equivalent to generating gaussian random maps over space, and then convolving the map with a low-pass filter with flat spectrum in the desired frequency range and 0 spectrum elsewhere). Maps of the desired size (40x40 pixels) where then cropped out. The same process and comparable values of the frequency cutoff were used in all other experiments where null maps were needed (ie. Figures S1 and S2).

PATTERN FORMATION THEORY PREDICTS STRUCTURE OF LEARNED REPRESENTATIONS

All of the trained models discussed in this work face essentially the same encoding problem of choosing n_g encoding maps that generate n_p place cell spatial maps through a single layer of synaptic weights (graphically depicted in Figure 1). We formalize this objective mathematically as follows:

Define the following matrices: $P \in \mathbb{R}^{n_x \times n_p}$ contains the responses of the n_p place cells at all n_x spatial locations; $G \in \mathbb{R}^{n_x \times n_g}$ contains the responses of the n_g encoding cells at all n_x spatial locations; $W \in \mathbb{R}^{n_g \times n_p}$ contains the readout weights from encoding cells to place cells. The goal is then to minimize

$$\mathcal{E}(G,W) = \|P - \widehat{P}\|_F^2, \text{ where } \widehat{P} = GW. \tag{Equation 5}$$

Here we consider an L2 penalty on encoding errors because it simplifies the analysis, but we observe similar results for numerical simulations with a softmax cross-entropy loss (see Figure 3).

Because we would like to understand the dominant patterns learned by the hidden neurons G, we make two simplifications to the above objective. First, we replace W by its optimal value for fixed G:

$$0 = \frac{\partial \mathcal{E}}{\partial W} = -2G^TP + 2G^TGW \tag{Equation 6}$$

$$\Rightarrow W^*(G,P) = \left(G^TG\right)^{-1} G^TP. \tag{Equation 7}$$

Observing that the objective $\mathcal{E}$ in (5) is invariant to any invertible transformation Z of the form $G \rightarrow GZ$, $W \rightarrow Z^{-1}W$, we can simplify our objective by choosing Z so that G's columns are orthonormal. (Because we will eventually consider the simpler case of a single encoding map, in which case orthogonality holds trivially, transforming by Z has no effect on our final conclusions.) Plugging (7) into (5) (and multiplying by a factor of $\frac{1}{n_p}$ for convenience), we obtain the following constrained optimization problem for G:

$$\min \frac{1}{n_p} \|P - GG^T P\|_F^2, \text{ s.t.} G^T G = I. \qquad \text{(Equation 8)}$$

This constrained optimization problem can be solved by considering the Lagrangian

$$\mathcal{L} = \mathrm{Tr}\left[G^T \Sigma G - \lambda\left(G^T G - I\right)\right], \qquad \text{(Equation 9)}$$

where $\Sigma = (1/n_p)PP^T$ is the $n_x \times n_x$ place cell similarity matrix (shown in Figure 3A).

To build intuition, we consider the optimal pattern learned by a *single* hidden neuron. Replacing G with g, an $n_x \times 1$ vector of activations of a single hidden neuron at all points in space, we obtain the single-neuron Lagrangian

$$\mathcal{L} = g^T \Sigma g + \lambda(1 - g^T g). \qquad \text{(Equation 10)}$$

This is the simplest version of the position encoding objective. If we model the training of our neural network as performing gradient ascent on this objective, then the learning dynamics take the following form:

$$\frac{d}{dt} g = -\lambda g + \Sigma g. \qquad \text{(Equation 11)}$$

This is a **pattern forming dynamics**. As gradient ascent proceeds, the firing fields at two locations $g_x, g_{x'}$ will mutually excite (inhibit) one another if the place cell similarity $\Sigma_{xx'}$ at the two locations is positive (negative), and over time g will develop stable patterns across space. Solving these dynamics subject to the normalization constraint $g^T g = 1$, we find that the stable fixed point corresponds to the top eigenmode of Σ.

The top eigenmodes of Σ then take a very simple form. Assuming the place cell receptive fields uniformly cover space, then in the limit of many place cells, their similarity structure Σ is translation invariant: $\Sigma_{x,x'} = (1/n_p)\sum p_i(x)p_i(x') = (1/n_p)\sum p_i(x+\Delta)p_i(x'+\Delta) = \Sigma_{x+\Delta, x'+\Delta}$. Without the boundaries, or with periodic boundary conditions on the box, this translation invariance would imply that Σ 's eigenvectors are exactly Fourier modes across space. However, even with the boundaries, $\Sigma_{x,x'}$ has a Toeplitz structure and its eigenmodes are still well approximated by Fourier plane waves across space (Figure 3B). To compute the eigenvalue λ_k associated to the k^{th} Fourier mode f^k, let $p(x)$ be the place cell tuning curve over space, and Δ_i be the receptive field center of the i^{th} place cell. Then

$$\lambda_k = f^{k\dagger} \Sigma f^k \qquad \text{(Equation 12)}$$

$$= \frac{1}{n_p} \sum_{i,x,x} f^k(x)^* f^k(x') p_i(x) p_i(x') \qquad \text{(Equation 13)}$$

$$= \frac{1}{n_p} \sum_i \left(\sum_x f^k(x) p(x - \Delta_i)\right)^* \left(\sum_{x'} f^k(x') p(x' - \Delta_i)\right) \qquad \text{(Equation 14)}$$

$$= \frac{1}{n_p} \sum_i \left(e^{i\varphi(\Delta_i)} \widehat{p}_k\right)^* \left(e^{i\varphi(\Delta_i)} \widehat{p}_k\right) \qquad \text{(Equation 15)}$$

$$= |\widehat{p}_k|^2. \qquad \text{(Equation 16)}$$

In essence, the eigenvalue associated to the k^{th} Fourier mode is just the power of that Fourier mode in the place cell tuning curve, so that the optimal pattern g will be the Fourier mode with maximum power.

If the place cells are distributed isotropically across space, and their tuning curve is circularly symmetric then $\Sigma_{Rx,Rx'} = \Sigma_{x,x'}$ for any rotation matrix R, and consequently all rotations of the optimal Fourier mode will also be optimal:

$$\lambda_{Rk} = \sum_{x,x'} \Sigma_{x,x'} \left(f_x^{Rk}\right)^* f_{x'}^{Rk} \qquad \text{(Equation 17)}$$

$$= \sum_{x,x'} \Sigma_{x,x'} \left(f_{R^{-1}x}^k\right)^* f_{R^{-1}x'}^k \qquad \text{(Equation 18)}$$

$$= \sum_{x,x'} \Sigma_{Rx,Rx'} \left(f_x^k\right)^* f_{x'}^k \qquad \text{(Equation 19)}$$

$$= \sum_{x,x'} \Sigma_{x,x'} \left(f_x^k\right)^* f_{x'}^k \qquad \text{(Equation 20)}$$

$$= \lambda_k \qquad \text{(Equation 21)}$$

Thus, the top eigenspace of Σ is degenerate and consists of all Fourier modes whose wavevector k lies on a ring centered around the origin in Fourier space (Figure 3C). In other words, the optimal map g is any linear combination of plane waves of optimal wavelength $1/|k^*|$, which can combine to form square, or hexagonal or even amorphous grid maps (Figure 3D). As we show below, this multiplicity of solutions is a special feature due to the lack of constraints. Once a nonlinear constraint such as non-negativity is added, the optimization favors a single type of map corresponding to hexagonal grid cells.

A nonnegativity constraint favors hexagonal grids

We have seen empirically that a nonnegativity constraint tends to produce hexagonal grids (Figure 2D). To understand this effect, we add a softened nonnegativity constraint to our objective function as follows

$$\mathcal{L} = g^T \Sigma g + \lambda\left(1 - g^T g\right) + \sigma(g), \qquad \text{(Equation 22)}$$

where $\sigma(g)$ penalizes negative activities in the map g. It will be convenient to write g_x as $g(\vec{x})$, treating g as a scalar field defined for all points in space. Our objective then takes the form

$$\mathcal{L}[g(\vec{x})] = \iint_{x,x'} g(\vec{x})\Sigma(\vec{x} - \vec{x}')g(\vec{x}') + \lambda\left(1 - \int_{\vec{x}} g^2(\vec{x})\right) + \int_{\vec{x}} \sigma(g(\vec{x})). \qquad \text{(Equation 23)}$$

We can approximate the negativity penalty by Taylor expanding about 0: $\sigma(g) \approx \sigma_0 + \sigma_1 g + \sigma_2 g^2 + \sigma_3 g^3$. Our Lagrangian then has a straightforward form in Fourier space

$$\tilde{\mathcal{L}}\left[\tilde{g}\left(\vec{k}\right)\right] \approx \int_{\vec{k}} \left|\tilde{g}\left(\vec{k}\right)\right|^2 \tilde{\Sigma}\left(\vec{k}\right) + \lambda\left(1 - \int_{\vec{k}} \left|\tilde{g}\left(\vec{k}\right)\right|^2\right)$$
$$+ \left[\sigma_0 + \sigma_1 \tilde{g}\left(\vec{0}\right) + \sigma_2 \int_{\vec{k}} \left|\tilde{g}\left(\vec{k}\right)\right|^2 + \sigma_3 \iiint_{\vec{k},\vec{k}',\vec{k}''} \tilde{g}\left(\vec{k}\right)\tilde{g}\left(\vec{k}'\right)\tilde{g}\left(\vec{k}''\right)\delta\left(\vec{k} + \vec{k}' + \vec{k}''\right)\right]. \qquad \text{(Equation 24)}$$

σ_0, σ_1, and σ_2 will not qualitatively change the structure of the solutions: σ_0 simply shifts the optimal value of $\mathcal{L}$, but not its argmax; σ_1 controls the amount of the constant mode in the maps, and does not affect their qualitative shape; and σ_2 can be absorbed into $\tilde{\lambda}$ (Cross and Greenside, 2009). Critically, however, the cubic term σ_3 introduces an interaction between wavevector triplets $\vec{k}, \vec{k}', \vec{k}''$ whenever the three sum to zero (3I).

In the limit of weak σ_3, the maps will be affected in two separate ways. First, weak σ_3 will pull the maps slightly outside of the linear span of the optimal plane-waves, or eigenmodes of Σ of largest eigenvalue. As $\sigma_3 \to 0$, this effect shrinks and effectively disappears, so that we can assume the optimal maps are still constrained to be linear combinations of plane waves, with wave-vectors on the same ring in Fourier space. The second, stronger effect is due to the fact that no matter how small σ_3 is made, it will break $\mathcal{L}$'s symmetry, effectively forcing it to choose one solution from the set of previously degenerate optima. Therefore, in the limit of small σ_3, we can determine the optimal maps by considering which wavevector mixture on the ring of radius k^* maximizes the nonlinear term

$$\mathcal{L}_{\text{int}} = \iiint_{\vec{k},\vec{k}',\vec{k}''} \tilde{g}\left(\vec{k}\right)\tilde{g}\left(\vec{k}'\right)\tilde{g}\left(\vec{k}''\right)\delta\left(\vec{k} + \vec{k}' + \vec{k}''\right). \qquad \text{(Equation 25)}$$

Subject to the normalization constraint $\int |\tilde{g}(k)|^2 = 1$, this term is maximized when $\tilde{g}$ puts all weight on a single wavevector triplet which sums to zero: $\vec{k} + \vec{k}' + \vec{k}'' = \vec{0}$. The only such combination on the ring of radius $|k^*|$ is an equilateral triangle, so that the optimal solutions (up to rotation) are $\tilde{g}(k) = (1/\sqrt{6}) \sum_{a=0,60,120} \delta(\vec{k} - \vec{k}^a) + cc^1$ (Figure 3I; Note that cc. is shorthand for complex conjugate. For any real solution g(x) to Equation 22, $\tilde{g}(k) = \tilde{g}\dagger(-k)$. Therefore, for each wavevector k we must also include its negative, $-$k). Therefore, rather than arbitrary linear combinations of plane waves, the optimal solutions consist of three plane waves with equal amplitude and wavevectors that lie on an equilateral triangle.

$$g(x) = \frac{1}{\sqrt{6}}\left(e^{i\overrightarrow{k}^{1}\cdot\overrightarrow{x}+\varphi_1} + e^{i\overrightarrow{k}^{2}\cdot\overrightarrow{x}+\varphi_2} + e^{i\overrightarrow{k}^{3}\cdot\overrightarrow{x}+\varphi_3} + cc.\right). \quad \text{(Equation 26)}$$

The interaction $\mathcal{L}_{\text{int}}$ is maximized when $\varphi_1 + \varphi_2 + \varphi_3 = 0$, in which case the three plane waves interfere to form a regular hexagonal lattice (Figure 3J).

Hexagonal grids and $g \rightarrow -g$ symmetry breaking

We see from the above argument that the rectification nonlinearity is but one of a large class of nonlinearities which will favor hexagonal grids. A generic nonlinearity with a non-trivial cubic term in its Taylor expansion will break the $g \rightarrow -g$ symmetry, and introduce a three-body interaction which picks out hexagonal lattices. While nonnegativity is a specific nonlinearity motivated by biological considerations, a broad class of nonlinearities will achieve the same effect (Sorscher et al., 2019).

Numerical simulation of pattern-forming systems

As we show above, the problem of optimally encoding place cell activities yields the following Lagrangian:

$$\mathcal{L} = g^T \Sigma g + \lambda\left(1 - g^T g\right) + \sigma(g) \quad \text{(Equation 27)}$$

where $\Sigma = P^T P$ is a $(n_x \times n_x)$ matrix encoding the similarity of the place cell activities at any given pair of spatial locations, and σ captures the nonnegativity constraint by penalizing negative activities.

Place cell tuning curves are modelled as above in the path integration task. We sampled these tuning curves ($n_P = 512$) on a grid of spatial locations ($n_x \times n_x = 100 \times 100$) to obtain the matrix of place cell responses, P, and computed $\Sigma = P^T P$ (Figure 3A). Difference of softmax tuning curves yield a similarity matrix with a characteristic center-surround structure (Figure 3A), very similar to the similarity matrix for difference gaussian tuning curves, which can be computed analytically (see appendix for details), although the location of the ring of minima may be different, as the different normalization of the difference of softmax tuning curves can lead to a different location of the ring of minima. As we have seen, this center-surround structure leads to periodic grid-like maps.

In the unconstrained case $\sigma = 0$, the optimum can be obtained directly by sampling from the top eigenspace of Σ. This is shown in Figure 3B,C. For other choices of σ, we numerically optimized $\mathcal{L}$ via gradient descent: $g \rightarrow g + \eta\frac{\partial\mathcal{L}}{\partial g}$. Optimization was run until approximate convergence. Figure 3J shows one such map for $\sigma = \text{relu}(x)$.

Interpretation of the encoding objective

In the Lagrangian form of Equation 22, one can interpret the optimization as attempting to capture place cell activity (first term, $g^T \Sigma g$), with minimum total neural activity (second term, $\lambda(1 - g^T g)$), and minimum negativity penalty (third term, $\sigma(g)$). Interestingly, by repeating the above analysis, instead optimizing out the activity G rather than the weights W (cf. Equation 7), one arrives at a Lagrangian for the weights which is identical in form to Equation 22. Thus hexagonal grids can alternatively be interpreted as maps which fit place cell activity subject to minimum *synaptic weight* and minimum *synaptic* negativity penalty.

ATTRACTOR MANIFOLD ANALYSIS

For the analysis of Figure 4, fixed points of the network dynamics are defined as population activity patterns r_i^* that are left invariant by a step of the RNN dynamics, in the absence of velocity inputs

$$r_i^* = \sigma\left[\sum_{j=1}^{n_g} J_{ij} r_j^*\right]. \quad \text{(Equation 28)}$$

We identified slow points of the network dynamics by minimizing the scalar function

$$q(r) = \sum_i \left(r_i - \sigma\left[\sum_{j=1}^{n_g} J_{ij} r_j\right]\right)^2 \quad \text{(Equation 29)}$$

using the procedure outlined in Sussillo and Barak (2013). We collected a set of 100^2 fixed points by initializing the network on a grid of 100×100 spatial locations in the environment.

In order to identify the low-dimensional structure of population activity, we computed three spatial phases for each neuron's rate map,

$$\varphi_i^a = \arg\left[\int d\overrightarrow{x}\, e^{-\overrightarrow{k}^{a}\cdot\overrightarrow{x}}\, r_i(\overrightarrow{x})\right], a = 0, 60, 120. \quad \text{(Equation 30)}$$

where $r_i(\vec{x})$ is the activity of neuron i at location $\vec{x}$, and $\vec{k}^0, \vec{k}^{60}, \vec{k}^{120}$ are the 0°, 60°, and 120° unit vectors. We then projected the population activity onto the following three pairs of axes,

$$v_i^a \equiv \cos(\varphi_i^a), w_i^a \equiv \sin(\varphi_i^a), a = 0, 60, 120. \quad \text{(Equation 31)}$$

If each neuron in the network was a perfect hexagonal grid cell, then its firing rate could be written as,

$$r_i(\vec{x}) = \sum_a \cos\left(\vec{k}^a \cdot \vec{x} - \varphi_i^a\right) \quad \text{(Equation 32)}$$

$$= \sum_a \cos(\varphi_i^a)\cos\left(\vec{k}^a \cdot \vec{x}\right) + \sin(\varphi_i^a)\sin\left(\vec{k}^a \cdot \vec{x}\right) \quad \text{(Equation 33)}$$

$$= \sum_a v_i^a \cos\left(\vec{k}^a \cdot \vec{x}\right) + w_i^a \sin\left(\vec{k}^a \cdot \vec{x}\right) \quad \text{(Equation 34)}$$

Hence projecting the population activity onto the three pairs of axes v_i^a, w_i^a defined above would reveal a set of three perfect rings. In the case of the trained neural network, projecting the slow points onto the three pairs of axes v_i^a, w_i^a yields three pronounced rings 4D. The subspace spanned by the six vectors v_i^a, w_i^a explains 52% of the total variance of the population activity.

As an additional measure to ensure that the twisted torus captures the dominant structure of the attractor manifold, we computed pairwise distances $d(\vec{x}) = \|\vec{r}(\vec{x}) - \vec{r}(\vec{x}_0)\|^2$ between the population activity $\vec{r}(\vec{x}_0)$ at a reference point in the environment, and the population activity $\vec{r}(\vec{x})$ at all other points in the environment, and plotted $d(\vec{x})/\max_{\vec{x}} d(\vec{x})$ as a function of space 4C.

Persistent homology

As a model-free alternative to the torus-based analysis described immediately above, we performed persistent homology on the attractor manifold of activity patterns. This analysis was performed on the time-averaged rate map data $r_i(x)$, which approximately describes the fixed point in n_g -dimensional neural activity space while the agent passes over spatial location x.

The maps were first preprocessed by extracting a central $L \times L$ square from each rate map. This was done to minimize the effect of rate map artifacts near the periphery of the environment. Results were broadly similar even when this preprocessing step was skipped. Data corresponding to the n_g -dimensional neural activity was then projected into 7 dimensions via PCA in order to reduce the computational cost of persistent homology (results did not depend sensitively on the exact dimensionality chosen). The resulting 7-dimensional data was used as input to the persistent homology analysis provided by the Ripser python library (coefficient field: p_{47}, metric: cosine, maximum dimensionality: 2; see documentation at https://ripser.scikit-tda.org/en/latest/index.html). The output of this analysis is a list of birth and death radii for each cycle present in the data, and for each dimensionality analyzed (in our case, 0, 1, and 2). These birth and death times are plotted in 4C. The beige background is obtained by the same analysis performed on shuffled data where each neuron's rate map is permuted independently.

IDEALIZED PATH INTEGRATOR MODELS

For all experiments probing trained RNN mechanisms, we implemented an idealized model for comparison (cf. section "velocity based updating of positional information in two dimensions"; Methods S1 and S2; Figures S4 and S5). Idealized models were designed to capture the two core mechanisms of previously published models: center-surround connectivity and offset recurrent weights. Update equations were

$$r_i^{t+1} = \sigma\left[\sum_{j=1}^{n_g} J_{ij} r_j^t + M_{ix} v_x^t + M_{iy} v_y^t + b_i\right]. \quad \text{(Equation 35)}$$

where J is the recurrent weight matrix, M is the velocity input weights, v is 2-dimensional velocity of the agent, b is a constant vector representing feedforward drive, and σ is the neuron nonlinearity. Neurons were placed uniformly over a grid of integer points on a 2-dimensional neural sheet with side length $L = \sqrt{n_g}$. For two neurons with neural sheet positions $\vec{s}_i$ and $\vec{s}_j$, synapse weight J_{ij} was set as

$$J_{ij} = f\left(\vec{s}_i - \vec{s}_j - \vec{\beta}_j\right) \quad \text{(Equation 36)}$$

$$f(\vec{x}) = \sum_{a\in\{0,60,120\}} \cos\left(\frac{2\pi}{L}\vec{k}^a \cdot \vec{x}\right) \quad \text{(Equation 37)}$$

where $\vec{k}^0, \vec{k}^{60}, \vec{k}^{120}$ are the 0°, 60°, and 120° unit vectors as defined above and β_j is the outgoing connectivity bias of neuron *j.* For all models, we set $b_i = 1$. Following Burak and Fiete (2009), for a neuron with neural sheet position $\vec{s}_i = \begin{pmatrix} p \\ q \end{pmatrix}$ we set $M_{ix} = (q\text{mod}2)(-1)^p$ and $M_{iy} = (p\text{mod}2)(-1)^q$ (equivalent to dividing the neural sheet into 2x2 neuron zones each with a north, east, south and west-motion sensitive cell).

For Figure 5, $\vec{\beta}_j$ was set to 0 (ie. weights had no offset) to isolate the stable firing mechanism. For later figures probing updating mechanisms, $\vec{\beta}_i = \begin{pmatrix} M_{ix} \\ M_{iy} \end{pmatrix}$ (intuitively, cells recieving north (south) velocity input have north (south) connectivity offsets; similarly for east/west). For Figure 6, the nonlinearity σ is the rectifying linear function.

The idealized 1-dimensional path integrator model referenced in Figures S4 and S5 was implemented analogously. The update equation was the same as above, with $v \in \mathbb{R}$ now representing the agent's 1-dimensional velocity. Each neuron now has a position s_i in a 1-dimensional neural ring. The weight between neurons at ring positions s_i and s_j was set analogously as

$$J_{ij} = \cos\left[\frac{2\pi}{N}(s_i - s_j - \beta_j)\right] \quad \text{(Equation 38)}$$

For all models, we set $M_i = (-1)^i$ and $b_i = 1$. For Figure S4, β_j was set to 0 (ie. weights had no offset) to isolate the stable firing mechanism. For later figures probing updating mechanisms, $\beta_i = (-1)^i\delta$. For Figure S5, the nonlinearity σ is the rectifying linear function.

SORTING RNN UNITS ONTO A NEURAL LINE AND A NEURAL SHEET

1-dimensional path integrator networks

Two methods were employed for sorting units in 1-dimensional path integrator networks onto a neural line, one based on neuron activity and one based on neuron connectivity. The first method consists of sorting neurons by preferred head direction (defined as the angle eliciting peak response for each neuron). This sort was used to produce the Fourier-transformed weight matrix in Figure S4F and for Figure S5.

To sort neurons by connectivity, each neuron i was assigned a random embedding coordinate θ_i. The θ's were then adjusted to maximize the energy function $\mathcal{E} = \sum_{ij} W_{ij}\cos(\theta_i - \theta_j)$ by gradient ascent in the direction: $\theta_i' = \frac{\partial\mathcal{E}}{\partial\theta_i} = \sum_j (W_{ij} + W_{ji})\sin(\theta_j - \theta_i)$. The dynamics were simulated until improvement in $\mathcal{E}$ was small. The resulting θ coordinates represent a continuous approximation to the discrete neuron ordering that optimizes the similarity between the weights and the center-surround connectivity matrix $S_{ij} = \cos\left(\frac{2\pi}{N}(i - j)\right)$. Neurons were then ordered by their θ coordinate, giving the final sort. This sort was used for Figure S4H.

2-dimensional path integrator networks

The guiding intuition behind sorting was that network mechanisms would be easier to discern if neurons with similar spatial response properties were physically nearby on the neural sheet (Figures 5 and 6). To do this, we obtain rate maps for all units in the network. For each neuron's map, we compute three spatial phases

$$\varphi_i^a = \arg\left[\int d\vec{x}\, e^{-\vec{k}^a \cdot \vec{x}} r_i(\vec{x})\right], a = 0, 60, 120 \quad \text{(Equation 39)}$$

where $r_i(\vec{x})$ is the activity of neuron i when the model is at spatial location $\vec{x}$, and $\vec{k}^0, \vec{k}^{60}, \vec{k}^{120}$ as above are the 0°, 60°, and 120° unit vectors. In the case of a perfect grid map, these phases uniquely determine the rate map of the neuron to be

$$r_i(\vec{x}) = \sum_a \cos\left(\vec{k}^a \cdot \vec{x} - \varphi_i^a\right) \quad \text{(Equation 40)}$$

Under an ideal sort, moving from neuron to neuron along the neural sheet should continuously shift the rate map. We'll posit that this shift of the rate map is proportional to the displacement on the neural sheet - that is a grid cell at neural sheet location $\vec{s}_i$ would have rate map

$$\hat{r}_i(\vec{x}) = \sum_a \cos\left(\vec{k}^a \cdot \left(\vec{x} - \vec{s}_i\right)\right) = \sum_a \cos\left(\vec{k}^a \cdot \vec{x} - \hat{\varphi}_i^a\right) \quad \text{(Equation 41)}$$

where $\hat{\varphi}_i^a$ are the "ideal phases" associated with a neuron at sheet location $\vec{s}_i$ (and where we've slightly abused notation by treating $\vec{s}_i$, which is a neural sheet location, as a real-space location in the the formula above. In reality, there would be a proportionality constant relating neural sheet units to real-space units, which we take to be equal to 1 and ignore here).

To approximate this ideal sort, we optimize the position of neuron i on the neural sheet to match the ideal phases $\widehat{\varphi}_i^a$ to the measured phases φ_i^a:

$$\widehat{s}_i = \mathrm{argmax}_{s_i} \sum_a \cos\left[\varphi_i^a - \widehat{\varphi}_i^a\left(\overrightarrow{s}_i\right)\right] \qquad \text{(Equation 42)}$$

This gives a set of 2d coordinates for each neuron $\overrightarrow{s}_i$. We bin the first coordinate into quantiles ($n = 64$) to obtain each neuron's first sheet index, and then sort the neurons in each quantile by their second coordinate to obtain each neuron's second sheet index (this procedure can be thought of as discretizing the first coordinate and then performing a lexographic sort on the second coordinates). The resulting indices place each of the 4096 neurons onto a single, unique node within a 64x64 grid. This 2D sorting procedure was used to extract the neural sheet from the activity maps in Figures 5 and 6.

FOURIER ANALYSIS OF RECURRENT WEIGHTS

We used Fourier analysis on the connectomes of our trained networks to explain the stability of their spatial representations (see "stable storage of positional information in two dimensions"; Figure 5; Method S1; Figure S4). 1-dimensional network units were sorted by preferred head direction. The weight matrix was reordered to reflect this sort. The matrix was then transformed to the real Fourier basis using the usual change of basis formula, $\tilde{W} = F^{-1}WF$, where F is the real Fourier matrix defined as

$$F_{ij} = \begin{cases} \cos\left(\frac{2\pi j}{n_g} i\right) & 0 \le j < n_g/2 \\ \sin\left(\frac{2\pi j}{n_g} i\right) & n_g/2 \le j < n_g \end{cases} \qquad \text{(Equation 43)}$$

with appropriate normalization of each column. For the plots in Figures S4C and S4F, the Fourier transformed weight matrices were shifted so that low frequency pattern weights appear in the middle of the weight matrix ("fftshift"), and the matrix was cropped to a smaller window around the center so peaks were more easily visible.

For 2-dimensional networks (see "stable storage of positional information in two dimensions"), units and weight matrix were first sorted by map phases as described above. The weight matrix was then reshaped to have 4 indices - the 2-d neural sheet coordinates of the input, (j_x, j_y), and output (i_x, i_y) neuron. The matrix was then transformed to the 2d Fourier basis, now using the 2d Fourier matrix defined as

$$F_{i_x, i_y, j_x, j_y} = \begin{cases} \cos\left(\frac{2\pi}{n_g}(j_x i_x + j_y i_y)\right) & 0 \le j_y < n_g, 0 \le j_x < n_g/2 \\ \sin\left(\frac{2\pi}{n_g}(j_x i_x + j_y i_y)\right) & 0 \le j_y < n_g, n_g/2 \le j_x < n_g \end{cases} \qquad \text{(Equation 44)}$$

again with appropriate normalization of each 2-dimensional column $F_{:,:,j_x,j_y}$. Change of basis was accomplished by the analog of the usual change of basis formula $\tilde{W}_{i_x,i_y,j_x,j_y} = \sum_{m,n} F^{-1}_{i_x,i_y,m_x,m_y} W_{m_x,m_y,n_x,n_y} F_{n_x,n_y,j_x,j_y}$. For plots in Figures 5C and 5G, we again shifted low frequencies to the center, extracted the diagonal $\tilde{W}_{i_x,i_y,i_x,i_y}$ and cropped around the center.

Peak strength statistics

To quantify the significance of weight matrix "peaks" in Fourier space (cf. Figure S5) we defined the peak strength as the fraction of matrix power on the lowest frequency sine and cosine self-weight: $\left(\tilde{W}^2_{1,1} + \tilde{W}^2_{-1,-1}\right) \Big/ \sum \tilde{W}^2_{i,j}$. We trained 1000 1-dimensional path integrator networks using the protocol above, and obtained initial and final weight matrices under initial and final sorts (initial and final sorts defined below). These scores are histogrammed in 6.

Two null distributions are shown in Figure S6. The first is the distribution of peak strengths of randomly generated matrices (entries drawn iid from a standard normal distribution) in their original sort (Figure S6B, first column). The second is the distribution of peak strengths of optimally sorted random matrices (Figure S6B, second column). Matrices were generated as before, but were then sorted by the connectivity-based sort designed to maximize peak strength described above (see "sorting RNN units onto a neural line and a neural sheet").

LINEARIZED RNN DYNAMICS

To gain a more quantitative understanding of how the 4 circuit components described above interact, we use the RNN's update equations to track the effect of a small velocity input dv added to a stable bump pattern (see "velocity based updating of positional information in two dimensions"; Figure 6; Method S2, and Figure S5).

$$r_i^{t+1} = \sigma\left[\sum_{j=1}^{n_g} J_{ij} r_j^t + M_{ix} dv_x + M_{iy} dv_y\right] \qquad \text{(Equation 45)}$$

$$\approx \sigma\left[\sum_{j=1}^{n_g} J_{ij} r_j^t\right] + \sigma_i'\left(M_{ix} dv_x + M_{iy} dv_y\right). \qquad \text{(Equation 46)}$$

where σ' is a diagonal matrix containing the derivatives of the neuron nonlinearities. Because the system is assumed to be at a fixed point, we can replace $\sigma\left[\sum_{j=1}^{n_g} J_{ij} r_j^t\right]$ with r_i^t, giving

$$r_i^{t+1} = r_i^t + \sigma_i'\left(M_{ix} dv_x + M_{iy} dv_y\right). \qquad \text{(Equation 47)}$$

The term on the right captures the effect of the velocity cell activation dv, travelling through its input weights M, and subsequently through the neuron nonlinearity σ'. Note, however, that after one step, the velocity input has not yet passed through the offset recurrent weights J. This will only occur during the *next* step:

$$r_i^{t+2} = \sigma\left[\sum_{j=1}^{n_g} J_{ij} r_j^{t+1}\right] \qquad \text{(Equation 48)}$$

$$= \sigma\left[\sum_{j=1}^{n_g} J_{ij} r_j^t + \underbrace{J_{ij}\sigma_j'\left(M_{jx} dv_x + M_{jy} dv_y\right)}_{\overrightarrow{\Delta}_i}\right]. \qquad \text{(Equation 49)}$$

To compute $\overrightarrow{\Delta}$, we note that because σ is the rectifying linear function, $\sigma'(x) = \mathbb{I}[x > 0]$, so that $\overrightarrow{\Delta}_i(\theta) = \sum_{j \text{ active}} J_{ij}(M_{jx} dv_x + M_{jy} dv_y)$.

For the 1-dimensional network (Method S2; Figure S5), the ground truth shifting term $\overrightarrow{T}$ is defined as the tangent to the attractor manifold, $\overrightarrow{T}_i(\theta) = \frac{\partial r_i(\theta)}{\partial \theta}$, where $r_i(\theta)$ is the activity of the i^{th} neuron for head direction θ. This can be approximated as the difference between patterns at two nearby head directions: $\overrightarrow{T}_i(\theta) \approx \frac{r_i(\theta + \Delta) - r_i(\theta)}{\Delta}$.

For the 2-dimensional network (see "velocity based updating of positional information in two dimensions"; Figure 6), the empirical and ground truth shifting terms depend on the 2-dimensional velocity $\overrightarrow{dv}$. The same analysis as above gives $\overrightarrow{\Delta}_i(\overrightarrow{x}) = \sum_{j,k \text{ active}} J_{ij} M_{jk} \overrightarrow{dv}_k$ for the empirical shifting term. The ground truth shifting term was computed as $\overrightarrow{T}(\overrightarrow{x})_i \approx \frac{r_i(\overrightarrow{x} + \overrightarrow{dv}) - r_i(\overrightarrow{x})}{|\overrightarrow{dv}|}$, or the partial derivative of the attractor manifold in the direction $\overrightarrow{dv}$.

EXTRACTING FOURIER MODES FROM TOP CONNECTIVITY EIGENVECTORS

2d RNN

Just as the population activity structure was not clear when projected onto its top PC eigenvectors, but became clear when we rotated to a more useful subspace, the top eigenvectors of the connectivity matrix J appear as linear combinations of low-frequency plane waves when viewed on the sorted neural sheet. However, with a simple orthogonal combination of the top eigenvectors, we can disentangle the consituent plane waves and construct 3 pairs of approximate pure modes, corresponding to the 3 pairs of plane waves used in traditional attractor model (see "stable storage of positional information in two dimensions"; Figure 5). The eigenvectors of the traditional attractor model can be written as

$$\widehat{v}_i^a = \cos\left(\overrightarrow{k}^a \cdot \overrightarrow{s}_i\right), \widehat{w}_i^a = \sin\left(\overrightarrow{k}^a \cdot \overrightarrow{s}_i\right), a = 0, 60, 120 \qquad \text{(Equation 50)}$$

To see how closely the top eigenvectors $\overrightarrow{u}_j, j = 1, \ldots, 10$, of the trained RNN connectivity matrix J approximate the perfect plane waves of the traditional model, we find the best orthogonal combination of the $\overrightarrow{u}_j$. The connectivity matrix J of the trained RNN has ten large eigenvalues (Figure 5J), so we use only the top ten eigenvectors. Collecting the continuous attractor eigenvectors in a $n_g \times 6$ matrix $V = [\widehat{\vec{v}}^a, \widehat{\vec{w}}^a]$, and the top ten trained network eigenvectors in a $n_g \times 10$ matrix $U = [\vec{u}_1, \ldots, \vec{u}_{10}]$, we identify the optimal linear transformation O, a 10×6 matrix, by minimizing the mean squared error,

$$O = \underset{O}{\operatorname{argmin}} \|V - UO\|^2, \quad \text{(Equation 51)}$$

We then orthogonalize O using the Lowdin symmetric orthogonalization $O' = S^{-\frac{1}{2}} O$ where S is the symmetric overlap matrix $S_{ij} = O_i \cdot O_j$, and $S^{1/2} = U S_{diag}^{-\frac{1}{2}} U^\dagger$, where S_{diag} is obtained by diagonalizing S, $S_{diag} = U^\dagger S U$. The transformed connectivity eigenvectors $[v^a_{\text{conn.}}, w^a_{\text{conn.}}] = UO$ look like pure plane waves across the neural sheet, closely matching those in the traditional attractor model (Figure 5H).

CONNECTIVITY BIAS

In idealized path integrator networks, pattern updating is accomplished by discrete groups of cells with biased outgoing connectivity. To determine whether a similar pattern of biased outgoing connectivity exists in the trained networks, we first sorted the neurons onto a neural sheet (see "sorting RNN units onto a neural line and a neural sheet" above). Then, for each neuron, we defined the connectivity bias as the displacement from the neuron's own neural sheet position to the center of mass of its outgoing synaptic weights over the neural sheet:

$$\vec{\beta}_i = \frac{\sum_{\vec{s}_j} J_{ij} \vec{s}_j}{\sum_{\vec{s}_j} J_{ij}} - \vec{s}_i \quad \text{(Equation 52)}$$

Defined this way, if a neuron projects isotropically around itself on the neural sheet, its connectivity bias is $\vec{0}$.

The 2-dimensional displacements for the trained 2D integrator network are histogrammed in Figure 6E (right), along with those of the idealized model (left; showing one population of north, west, south, and east projecting cells).

ANALYSIS OF HETEROGENEOUS CELLS

Ablation experiments

Ablation experiments were performed by zeroing the activities of the ablated neurons at each timestep, preventing them from participating the recurrent dynamics. This mimics an experiment in an animal in which neurons are selectively knocked out, both preventing downstream neurons from reading out a spatial estimate, and also disrupting the recurrent dynamics of the network.

Readout of place cell maps and value functions

In brief, readout performance of either a) place cell maps (Figure 7C) or b) value functions (Figure S7) was compared for increasing numbers of either a) pristine grid maps, or b) maps from the trained RNN, which exhibit significant heterogeneity.

Regressor maps

Pristine grid cell maps were constructed to approximate those generated by a continuous attractor network (Burak and Fiete, 2009), with a spatial scale chosen to match that of the trained RNN ($\approx$ 0.05 cycles/pixel). Specifically, a basic 50x50 pixel template grid map was generated by adding 3 plane waves with wave vectors at 120 degrees to one another and phases that sum to zero (cf. Equation 26). The full set of pristine maps was obtained by randomly translating this pattern. These maps were compared to the RNN rate maps, obtained as in "grid score," and then down-sampled to 50x50 pixels to match the grid maps. Both sets of maps were scaled to a mean activity value of 1.

Target maps

In one experiment (Figure 7C), regressor maps were used to fit the place cell maps used during training (see "rnn training" for details on how these maps were constructed). Each place cell map was scaled to sum to 1 over all spatial locations. In another (Figure S7), regressor maps were used to fit random value functions. These were constructed using the same procedure as in "noise model" (0.05 cycles/pixel). Maps were then shifted to be nonnegative and scaled to sum to 1, allowing an interpretation as a reward distribution over space.

Fitting procedure

Pristine grid cells were ordered randomly, while the RNN cells were ordered in descending order by grid score. Cells were progressively added to each network. For each number of cells, weights were fit to minimize cross entropy between the target maps and regressor maps (optimizer: Adam, 200 steps; learning rate: 2×10^{-4}; l2 regularization coefficient: 10^{-4}). The final cross entropy is reported in Figures 7 and S7.

Brain fitting

Electrophysiology recordings of 778 MEC neurons in awake, behaving rats were obtained from Butler et al. (2019). Models of MEC were evaluated based on their ability to regress each MEC neuron's spatial firing rate map under a sparsity (L1) penalty. Regressions were performed using the Lasso method in the sklearn.linear_model class. Robustness of this analysis to the choice of the sparsity penalty was ensured by plotting regression curves across the range of all possible regressor sparsities (Figure 7D), where regressor sparsity is measured as the fraction of regressor weights equal to zero.

Max-entropy model

The max-entropy model in Figure 7D was fitted to the MEC neuron spatial rate maps $x_i \in \mathbb{R}^{n_x}, i = 1 \ldots n_g$, by extracting their first and second-order statistics, $\mu = (1/n_g) \sum_{i=1}^{n_g} x_i$, $\Sigma = \frac{1}{n_g - 1} \sum_{i=1}^{n_g} (x_i - \mu)(x_i - \mu)^T$. A new population of n_g synthetic maps $\widehat{x}_i, i = 1 \ldots n_g$ was then generated from this max-entropy model by sampling from the multivariate normal distribution $\widehat{x}_i \sim \mathcal{N}(\mu, \Sigma)$.