# Self-Supervised Grid Cells Without Path Integration


[*]**Markus Pettersen**
Department of Numerical Analysis and Scientific Computing
Simula Research Laboratory
Oslo, Kristian Augusts Gate 23
markusb@simula.no

[*]**Vemund Sigmundson Schøyen**
Department of Biosciences
University of Oslo
Oslo, Blindernveien 31
v.s.schoyen@ibv.uio.no

[*]**Mattis Dalsætra Østby**
Department of Physics
University of Oslo
Oslo, Sem Sælandsvei 24
mattis.dalsaetra@gmail.com

**Anders Malthe-Sørenssen**
Department of Physics
University of Oslo
Oslo, Sem Sælandsvei 24
anders.malthe-sorenssen@fys.uio.no

**Mikkel Elle Lepperød**
Department of Numerical Analysis and Scientific Computing
Simula Research Laboratory
Oslo, Kristian Augusts Gate 23
mikkel@simula.no


## Abstract


Grid cells, found in the medial Entorhinal Cortex, are known for their regular spatial firing patterns. These cells have been proposed as the neural solution to a range of computational tasks, from performing path integration, to serving as a metric for space. Their exact function, however, remains fiercely debated. In this work, we explore the consequences of demanding distance preservation over small spatial scales in networks subject to a capacity constraint. We consider two distinct self-supervised models, a feedforward network that learns to solve a purely spatial encoding task, and a recurrent network that solves the same problem during path integration. Surprisingly, we find that this task leads to the emergence of highly grid cell-like representations in both networks. However, the recurrent network also features units with band-like representations. We subsequently prune velocity inputs to subsets of recurrent units, and find that their grid score is negatively correlated with path integration contribution. Thus, grid cells emerge without path integration in the feedforward network, and they appear substantially less important than band cells for path integration in the recurrent network. Our work provides a minimal model for learning grid-like spatial representations, and questions the role of grid cells as neural path integrators. Instead, it seems that distance preservation and high population capacity is a more likely candidate task for learning grid cells in artificial neural networks.


[*]These authors contributed equally, names are ordered alphabetically.


Preprint. Under review.

# 1 Introduction

Known for their striking hexagonal spatial firing fields, grid cells [Hafting et al., 2005] of the medial Entorhinal Cortex (mEC) are thought to underpin several navigational abilities. These include path integration [Hafting et al., 2005, McNaughton et al., 2006, Burak and Fiete, 2009, Gil et al., 2017], forming a neural metric for space [Moser and Moser, 2008, Ginosar et al., 2023], vector navigation [Bush et al., 2015], and supporting memory and inference [Mulders et al., 2021, Whittington et al., 2020]. Given the range of different functions believed to be supported by grid cells, it is natural to investigate which of these tasks, if any, are actually performed by these enigmatic cells.

In the modeling literature, emphasis has been placed on grid cells as path integrators, with computational models establishing that grid cells are capable of doing path integration [Burak and Fiete, 2009]. Recently, it has also been shown that grid-like representations *emerge* in neural networks trained to path integrate [Cueva and Wei, 2018, Banino et al., 2018, Sorscher et al., 2022, Whittington et al., 2020, Xu et al., 2022, Dorrell et al., 2022, Schaeffer et al., 2023], which has been taken as evidence for grid cells performing path integration. However, this argument is based on correlation; under interventional cell ablations grid cells are as important for path integration as randomly selected cells [Nayebi et al., 2021] while band cells are significantly more important [Schøyen et al., 2023]. Moreover, these models are typically complex, featuring different architectures and activation functions, interacting label cell types (e.g. simulated place cell-like targets [Sorscher et al., 2022]), multiple regularization terms, additional constraints (e.g. path invariance [Schaeffer et al., 2023] or conformal isometry [Xu et al., 2022]), and large cell counts. All of these works report grid-like representations, making it difficult to disentangle exactly what function grid cells serve in the various models.

In this work, we therefore propose a minimal model of grid cell function and use this model to approach the question of whether grid cells do path integration. Concretely, we are inspired by the notion of grid cells serving as a metric for space, and propose an objective function that requires distance preservation over small spatial scales. In addition, we impose a capacity constraint, which favors distributed representations that occupy a minimal portion of the state space. We train neural networks to minimize the proposed objective functions, and find that strikingly hexagonal grid-like spatial representations emerge using these two simple ingredients.

To explore whether grid cells do path integration in our model, we ablate path integration itself by training a feedforward (FF) network to minimize a purely spatial version of the proposed objective, alongside a recurrent neural network (RNN) tasked with implicit path integration. We find that the feedforward network learns grid representations on par with those of the path integrating RNN model. However, some RNN units display distinct band cell-like [Krupic et al., 2012] spatial responses. When pruning velocity inputs to sampled subsets of units, we find that path integration contribution is inversely correlated with the mean sample grid score, with band-type cells providing the largest contribution.

Our findings suggest that grid cells may serve as a distributed high-capacity, distance-preserving representation. Unlike popular opinion, however, grid cells do not appear to be defined by the task of path integration. On the contrary, grid cells appear to be relatively unimportant for path integration, suggesting that grid cells may be more suitable for defining neural metrics for space, at least in artificial neural networks.

# 2 Methods

## 2.1 Data

We considered two distinct neural networks in this work, with different inputs. The feedforward network received Cartesian coordinate inputs, sampled randomly and uniformly from a square region with side lengths $4\pi \times 4\pi$. The recurrent network, on the other hand, received velocity inputs along trajectories sampled in the same square region. Additionally, the recurrent network was given the starting location of each trajectory (in Cartesian coordinates), to form a suitable initial state.

To generate trajectories, a bouncing procedure was used. Starting locations were sampled randomly and uniformly within the square arena. At each step, head directions were sampled according to a von Mises distribution with scale $\kappa = 4\pi$, and step sizes drawn from a Rayleigh distribution with

scale parameter $\gamma = 0.15$. Subsequently, we checked whether the resulting step landed outside the enclosure. If so, the component of the current velocity vector was reversed, effectively causing an elastic collision with the wall, and keeping the trajectory inside. Otherwise, the procedure was repeated until the desired number of timesteps and trajectories was achieved.

Due to the simplicity of the data, no datasets were pre-created, and all training data was new to the networks, i.e. created on the fly.

### 2.2 Neural Networks & Training

We consider two distinct architectures: A fully connected feedforward network and a recurrent neural network. The FF model consisted of two hidden layers, with 64 and 128 units, respectively, followed by an output layer of size $n_g = 256$ units, transforming 2D Cartesian coordinates to a latent space of $n_g$ dimensions. We applied the ReLU activation function after each hidden layer, and normReLU after the output layer. normReLU is a normalized ReLU function, which we take to be given by $\text{normReLU}(\mathbf{x}) = \text{ReLU}(\mathbf{x})/\text{maximum}(||\text{ReLU}(\mathbf{x})||, \varepsilon)$, with $\varepsilon = 10^{-12}$ a small constant to avoid zero division. Finally, we initialize the weights of each layer in the FF model uniformly between $-\sqrt{k}$ and $\sqrt{k}$, where $k$ is the number of input features to the layer.

Similar to the feedforward model, the RNN model featured $n_g = 256$ recurrently connected units. The state of the RNN model at a time $t$ was given by

$$\mathbf{g}_t = \text{normReLU}(W\mathbf{g}_{t-1} + W_{in}\mathbf{v}_t) \tag{1}$$

where $W \in \mathbb{R}^{n_g \times n_g}$ is a matrix of recurrent weights, $W_{in} \in \mathbb{R}^{n_g \times 2}$ a matrix of input weights, while $\mathbf{v}_t$ the velocity input at $t$. The initial state of the RNN was encoded with a feedforward neural network, with the exact same architecture as the FF model. In this case, the initial Cartesian position of the intended trajectory was provided as input. The weights in the recurrent matrix $W$ were initialized as the identity, while the weights in the input matrix $W_{in}$ were initialized uniformly, similarly to the FF model.

Both models were trained with a mini-batch size of 64 using the Adam optimizer with a learning rate of $10^{-3}$ [Kingma and Ba, 2017]. We set $n_g = 256$ for both models, and trained the RNN on 10-timestep trajectories. The RNN was trained for a total of 50000 training steps and the FF network for 100000 steps, on unseen data. Note that the RNN initial state encoder was trained from scratch when training the RNN, and did not reuse the previous, independent FF model.

The final parameters needed for training are $\sigma$ and $\alpha$, which define the loss function (4). $\sigma$ is an envelope scale parameter for the distance preservation loss term. $\alpha$ determines the relative weighting of the two terms in the loss, where $\alpha$ is the weight for the distance loss term and $1 - \alpha$ is the weight for the capacity loss term, chosen such that $\alpha \in [0, 1]$. We performed a grid search for $\alpha$ and $\sigma$ for both models to explore the impact of this weighting on the grid score of the representations. The values for $\alpha$ were chosen uniformly between 0.01 and 0.99, while $\sigma$ values were selected empirically around values that were seen to yield models high grid score during initial testing. After the grid search (Fig. 4 a)), the values $\sigma = 1.2$ and $\alpha = 0.54$ were chosen for the FF and RNN models as this yielded high grid scores for both models, even though multiple other combinations of $\alpha$ and $\sigma$ gave similar results.

### 2.3 Velocity Pruning

To investigate the influence of different RNN unit types on path integration ability, we ablated velocity inputs to subsets of recurrent units. Specifically, the velocity-pruned recurrent state was given by

$$\tilde{\mathbf{g}}_t = \text{normReLU}(W\mathbf{g}_{t-1} + \mathbf{m} \odot (W_{in}\mathbf{v}_t)),$$

where $\mathbf{m}$ is a binary mask that silences velocity input to select units, and $\odot$ the Hadamard product. RNN units were subselected so that a given subpopulation featured as many units as the number of low-gridscore units ($n = 29$; grid score cutoff 0.15). The subselection procedure was done for the full ensemble (random), as well as high grid-score units (grid score above cutoff). In each case, 1000 subpopulations were sampled randomly, with equal probability across units.

Since the proposed objective does not feature any decoding into a known target representation such as Euclidean coordinates, an alternate method of assessing whether the network is path integrating

correctly is needed. We therefore computed the mean square error between velocity-pruned representations and a target representation created by running the network on the same velocity input without pruning. In other words,

$$\text{Error}_t = \frac{1}{M}\sum_{i=1}^{M} ||\mathbf{g}_t^i - \tilde{\mathbf{g}}_t^i||^2, \tag{2}$$

where $M = 10000$ is the number of evaluated trajectories, and $T = 10$ the trajectory length. As a baseline, we also computed the mean squared difference with the initial RNN state, i.e.

$$\text{ISD}_t = \frac{1}{M}\sum_{i=1}^{M} ||\mathbf{g}_0^i - \tilde{\mathbf{g}}_t^i||^2, \tag{3}$$

for velocity pruning to randomly sampled subpopulations, as described previously.

### 2.4 Grid Statistics

To quantify the properties of the learned representations, we compute the grid score, orientation, spacing and phase of smoothed unit ratemaps. Smoothing was done using a Gaussian kernel from[The Astropy Collaboration, 2022], with a standard deviation of 2 pixels, for $64 \times 64$ ratemaps of unit activity. Grid scores were computed as the difference between the smallest and largest correlations, when comparing autocorrelogram annuli correlations at 60 and 30 degrees, respectively. Orientations were computed as the smallest angle of the six innermost peaks of the autocorrelogram (excluding the origin) with the horizontal. The grid spacing was computed as the average distance to the six innermost peaks. Finally, grid phases were taken to be the displacement of the closest grid peak to the origin of the ratemap.

### 2.5 Pattern Formation

To better understand how representations emerge in the feedforward and recurrent networks, we visualized the pattern formation process for select units. To do so, we weighted the activity of input ratemaps to a given unit by the network weight relating them. Then, weighted inputs were sorted according to their L2 norm over all spatial bins, and pattern forming was visualized using the cumulative sum of all weighted input ratemaps.

### 2.6 Out-of-bounds Evaluation

To explore the ability of the networks to generalize beyond their training regime, we evaluated trained feedforward and recurrent networks outside of the square training domain. Concretely, we ran the feedforward network with Cartesian coordinates sampled in a $8\pi \times 8\pi$ square, i.e., double the original enclosure wall lengths. For the RNN, we evaluated two cases, one in which the network started in the original square, and was allowed to move outside, and another wherein the network started out anywhere in the enlarged enclosure. In both cases, the RNN was run for a total of 50 timesteps, and responses aggregated over 10000 trajectories.

### 2.7 Subpopulation Loss Evaluation

To assess whether different cell types contributed differently to the various loss terms, we evaluated randomly sampled subpopulations of recurrent units on the capacity and distance losses separately. For a given sample, the subpopulation vector consisted of the flattened ratemaps of the selected units. Subsequently, the subpopulation vector was re-normalized to allow for a more direct comparison with the full network loss. As a reference, we also evaluated low grid score units ($n = 29$; grid score cutoff = 0.15) on either loss term. Additionally, we computed a baseline loss by randomly shuffling the bins of unit ratemaps, effectively achieving a spatially random representation. All subpopulations featured the same number of units ($n = 29$, as with the low grid score ensemble), and a total of 1000 samplings were performed.

### 2.8 Persistent Homology & Low Dimensional Projection

We used persistent homology to investigate whether the learned representation conformed to a simple geometric structure. Specifically, we used the Ripser python library [Tralie et al., 2018] to compute

persistence diagrams using spatially flattened ratemaps of unit activity. For the feedforward network, the outermost 20 % of ratemap pixels were excluded to avoid boundary effects, and only units with orientation in the range [0.4, 0.8] were included. This was done to avoid mixing possibly independent modules of units with different orientations. For the RNN ratemaps, the ratemaps of the full domain were used, and every unit was included in the analysis. For the analysis, we set $n_perm = 500$, and $max_dim = 2$, with otherwise default parameters.

We further visualized the neural structure of space using UMAP [McInnes et al., 2020] with $n_neigbhours = 2000$ and $min_dist = 0.8$ to project ratemaps of the feedforward and recurrent networks down to three dimensions. We colored the resulting 3D point cloud using the first principle component of the ratemaps, similar to Gardner et al. [2022], highlighting important regions. Ratemaps were processed in the same way as for the persistent homology analysis.

### 2.9 Figures

Figures were created using BioRender.com.

### 2.10 Code availability

Code for reproducing our model and findings is available at `https://github.com/bioAI-Oslo/GridsWithoutPI`.

## 3 Results & Discussion

We consider the problem of training a representation that preserves distances in a neighborhood around a current location, as illustrated in Fig. 1 a). Considering Cartesian coordinates $\mathbf{x}_t$ (e.g. along a trajectory) where $t$ indexes locations, we propose the following objective

$$\mathcal{L} = \mathbb{E}_{t,t'}\left[e^{-\frac{1}{2\sigma^2}\|\mathbf{x}_t-\mathbf{x}_{t'}\|}\left(\|\mathbf{x}_t-\mathbf{x}_{t'}\|-\|\mathbf{g}_t-\mathbf{g}_{t'}\|\right)^2\right]+\mathbb{E}_t[l_{cap}(\mathbf{g}_t)], \tag{4}$$

where $\sigma$ is the envelope scale parameter that determines the width of the neighborhood distance preservation in the exponential term. This creates a window around each spatial location where the difference between physical and neural distances should be minimized. Such gaussian radial basis functions are widely used and have been previously applied in e.g. normative models of grid cells to promote local separation of neural representations, as seen in Dorrell et al. [2022] and Schaeffer et al. [2023]. $\|\cdot\|$ denotes the Euclidean norm. $\mathbf{g}_t$ is the representation we wish to learn, which we parametrize with either a feedforward (FF) neural network or a recurrent neural network (RNN), as illustrated in Fig. 1 c) and described in section 2.2. Although both models must solve the same spatial encoding task, the FF model takes direct Cartesian coordinate inputs, while the RNN only receives an initial Cartesian position and subsequently receives Cartesian velocities. To correctly encode succeeding positions, the RNN therefore also needs to learn to path integrate. Notably, we constrain $\mathbf{g}_t$ to be non-negative and of constant L2 norm, i.e., $g_{it} \geq 0$ and $\|\mathbf{g}_t\| = 1$ for all $i, t$. The first loss term is minimized when Euclidean distances in the learned representation $\mathbf{g}$ equals the target Euclidean distances in a Cartesian representation in a small neighborhood around the current location. The second loss term, $l_{cap}$, is a capacity term. Xu et al. [2022] and Schaeffer et al. [2023] posit that capacity constraints are conducive to grid-like representations, and Schaeffer et al. [2023] proposed an L2 activity-based regularization term to maximize representational capacity. We, on the other hand, propose using an L1 capacity term given by

$$l_{cap}(\mathbf{g}_t) = -\sum_i g_{it}. \tag{5}$$

Using an L1 capacity term is markedly different from an L2 capacity term both empirically and mechanistically. A geometric illustration of when L1 capacity is optimal is illustrated in Fig. 1b). When $\mathbf{g}$ has constant L2 norm and non-negative elements, L2 capacity promotes representations with similar angles, but any angle (in the positive quadrant) is equally rewarded. Akin to L2 capacity constraints, the L1 capacity constraint (5) will also promote representations with similar directions. However, the L1 capacity constraint encourages maximally distributed and correlated cell activities. In other words, the full population vector state space (codomain) is ideally placed near the diagonal vector, with all units coactive.

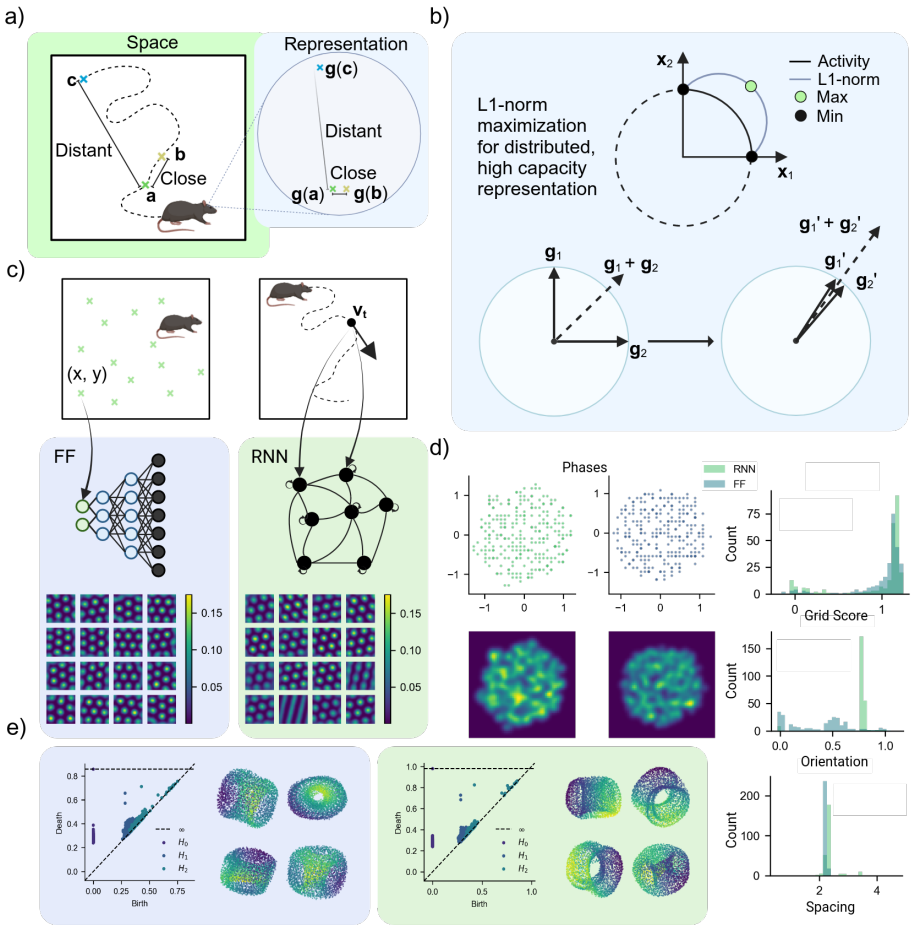


Figure 1: **Overview of models and objective function**. a) Illustration of the objective function: distant locations should be represented by distant population vector, close locations by close population vectors. b) Illustration of the L1 capacity constraint. c) Illustration of the investigated neural network architectures (feedforward and recurrent) and inputs. Below, ratemaps of a random selection of units are inset. d) Distributions of phases, grid scores, orientations, and spacing for both models. e) Persistence diagram and 3D UMAP projection of population ratemaps, for feedforward and recurrent networks. For the feedforward network, results are shown for units with orientation in the range $[0.4, 0.8]$.

Surprisingly, both FF and RNN models learn highly grid-like representations, as seen in the ratemaps in Fig. 1c), and quantified through histogram of the grid scores in Fig. 1 d). We further find in Fig. 1 d) that their phase distribution is seemingly random and uniform within the unit cell of the grid pattern. The histogram of grid spacings is unimodally peaked, indicating a single module in both models. However, the orientation histogram for the FF model is bimodal, suggesting two modules, but with identical spacing. Finally, Fig. 1e) shows the population vectors projected to three dimensions using UMAP alongside persistence diagrams quantifying the number of persistent 1,2 and 3-dimensional cocycles. Both models show one 0D, two 1D, and one 2D hole, indicating a toroidal manifold, which can also be seen visually from the UMAP projection.

Apart from the similarities in representations learned by the two models, we noticed that the RNN had a small set of cells with band-like representations, which is also visible in Fig. 1d) as a small bump near zero-grid score. Why would the model add this extra band-like subpopulation to the RNN? One key difference between the FF and the RNN is that the RNN is required to path integrate. It therefore seems especially strange that the amount of grid-like cells is reduced when the network is also required to path integrate, considering that mechanistic theories advocate grid cells as the neural substrate for path integration, and normative models argue that grid cells appear in RNNs trained for

path integration. Additionally, when evaluating the loss terms on distinct subpopulations, we observe in Fig. 4 b) that units with high grid scores achieve a somewhat lower distance loss, while band-type cells afford slight improvements in capacity. This supports the notion that grid cells are optimal for distance preservation.

To investigate the path integration role of the different cell types, we selectively pruned velocity inputs to different cells as illustrated in Fig. 2a). We categorized cell types with a grid score of less than 0.15 as band-like and the rest grid-like, as seen in Fig. 2 b). Pruning velocity input to the band-like cells shows a stark increase in (path integration) error, as given by Eq. 2, over time, as shown in Fig. 2c). Comparably, pruning velocity input to high grid score subpopulations showed almost no change in error over time. Figure 2e) displays the corresponding error when pruning uniformly across any cell (including both band and grid-like), which is accompanied by higher error. We hypothesise that this is due to the fact that low grid score units can be included in the subpopulation. Schøyen et al. found similar results when pruning band-like cells in Sorscher et al.'s RNN model of grid cells. We further demonstrate in Fig. 2d) how the error is linearly related to grid score, where pruning low grid score units has a high impact, and conversely, pruning high grid score units yields low error. Finally, we also compare the initial state difference (ISD), as described in Equation 3, to later states along different trajectories when pruning in Fig. 2 f). We see the ISD rise fastest with no pruning, as would be expected when moving away from the initial state. When pruning high grid score units, the trend is similar to no pruning, i.e., the neural representation is moved away from its initial state over time, as one would expect if the network was still path integrating. However, when pruning low grid score units (band cells), we see a much flatter ISD, indicating that the neural state does not change as much. In other words, the neural path integrator is close to being turned off. This provides compelling evidence that band cells, not grid cells, do path integration, at least in recurrently connected artificial neural network models.

To investigate whether all hexagonal patterns are created equal, we examined the pattern formation, connectivity structure, and generalization capabilities of the two models. In the bottom row of Fig. 3a), a sorted subset of feedforward unit ratemaps is shown, scaled by their outgoing weights to a selected output unit (displayed as a large ratemap to the right), as detailed in section 2.5. For the FF model, ratemaps of the penultimate layer serve as the basis representations forming the grid pattern in the final layer. This basis representation is diverse, encompassing various patterns such as place-like (see the final three ratemaps) and single-band-like (see ratemap in the final row, second column) ratemaps. These basis representations are non-periodic, indicating that the corresponding downstream grid representations cannot maintain periodicity outside their training domain. This non-periodic nature can also be seen directly in the FF-network architecture, which uses non-periodic activation functions. The bottom row of Fig. 3c) confirms that the pattern does not generalize the grid pattern outside its training domain.

The RNN, while facing a similar out-of-domain initialization challenge as the FF network, can potentially generalize beyond the training arena boundaries using a learned, periodic path integrator circuit. Subsequent ratemaps and their cumulative pattern formation rely on previously similar periodic patterns, as observed in the top rows of Fig. 3 a). Fig. 3c demonstrates that initializing the RNN inside its training domain and allowing it to path integrate beyond the domain reveals a clear periodic generalization. Conversely, starting outside its training domain results in the same non-generalization behavior as the FF network. Additionally, Fig. 3b shows that the connectivity profile of the recurrent cells follows a short-range excitation and long-range inhibition principle with respect to cells with neighboring phases, which is a known structure for generating grid-like representations [Burak and Fiete, 2009]. Interestingly, the band-like cells exhibit an excitation-inhibition connectivity profile shifted by their phase. Combined with the previous finding that band cells function as the neural path integrator circuit in the RNN, this suggests a mechanism for path integration through an excitation-inhibition shift of population activity in the wave direction of the band.

### 3.1 Summary and outlook

Our approach, featuring a minimal model (only two loss terms and a model without path integration), has allowed us to isolate key factors contributing to the emergence of grid-like representations. This simplification contrasts with previous complex models, which, while effective in generating grid-like patterns, often obscure the specific contributions of grid cells to navigational tasks due to their many-sided nature. Our findings demonstrate that grid cells, as encoded in feed-forward and recurrent

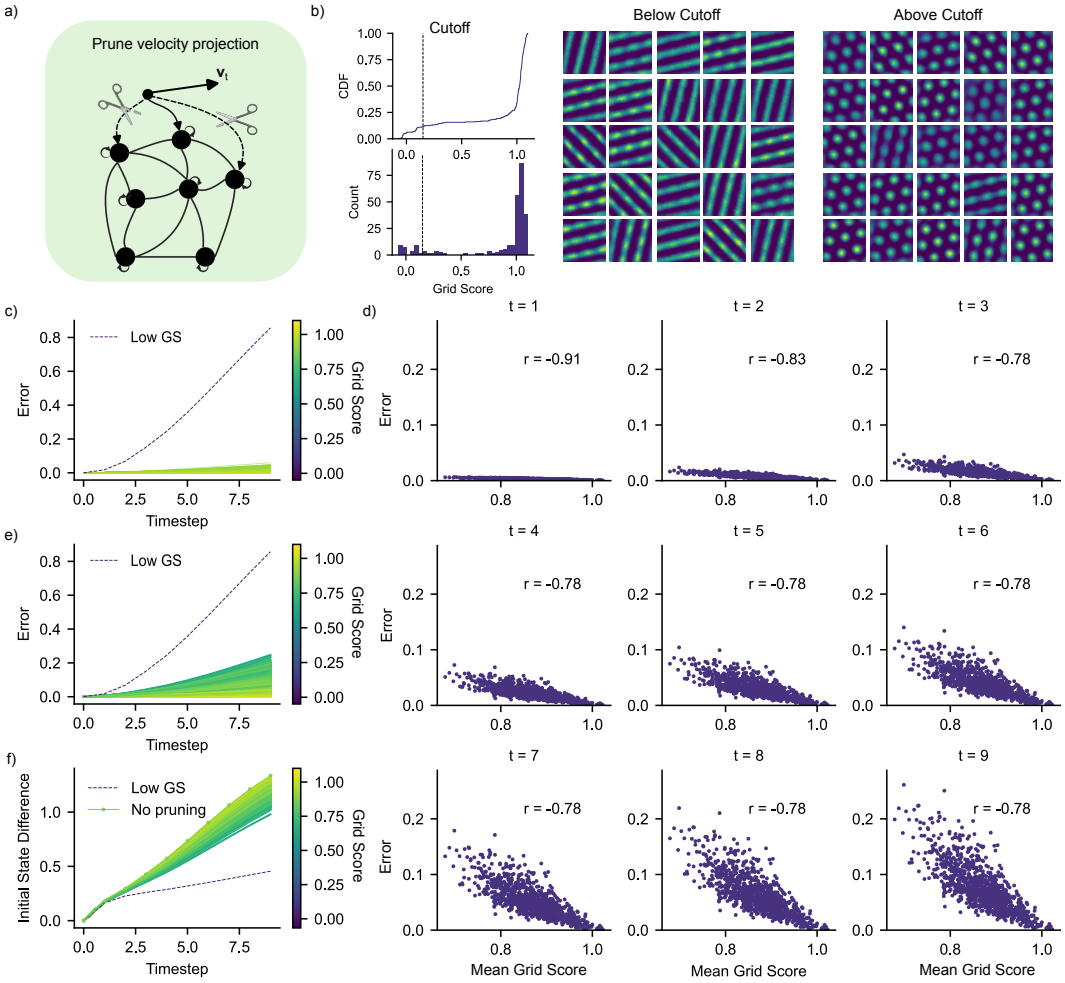


Figure 2: **Results of pruning velocity inputs to the RNN**. a) Illustration of the velocity input pruning: during pruning, velocity projections to randomly selected subsets of units are silenced. b) Grid score distribution, with indicated cutoff for low grid score units. Shown are also ratemaps for low and high grid score units. c) Path integration error for pruning of 1000 subpopulations among units with high grid score. Also shown is the error for low grid score units (dashed). d) Path integration error at every timestep, for pruning of randomly sampled subpopulations of units. Inset is Pearsons correlation coefficient between the subpopulation mean grid score and the PI error. e) As in c), for random subsamplings of the full population. f) Average initial state distance, for pruning random subpopulations of the full population. Also shown is the initial state distance for the low grid score selection (dashed).

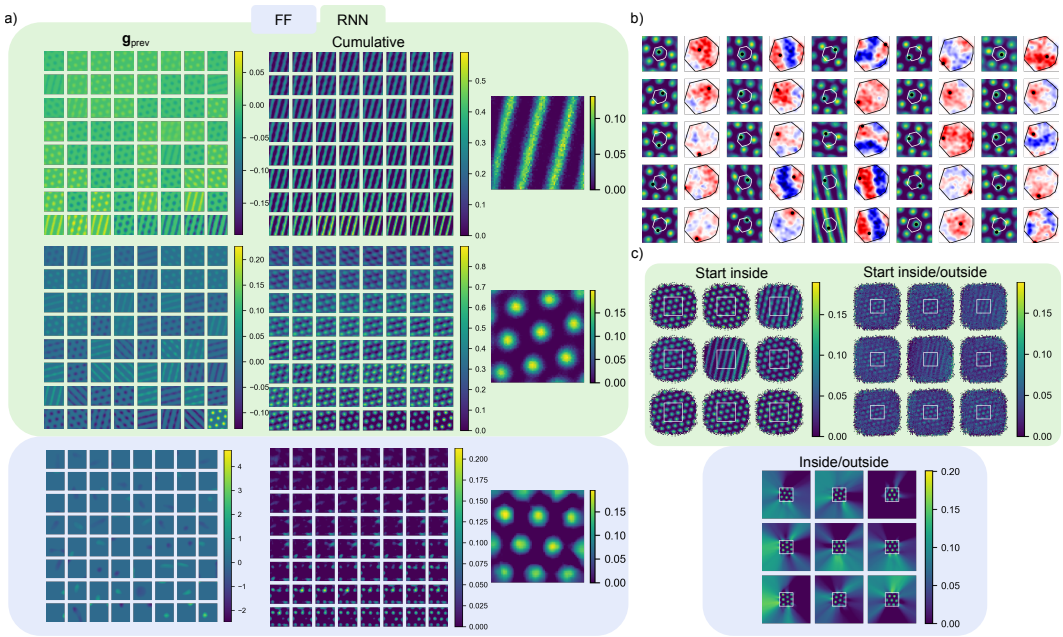


Figure 3: **Pattern formation and out-of-bounds behaviors**. a) Example pattern formation for select recurrent units (top and middle), and a feedforward unit (bottom); ratemaps inset on the right. $\mathbf{g}_{prev}$ denotes previous activations (penultimate layer activation for FF, second-to-last state for RNN), while cumulative representations shows the cumulative sum of weighted unit inputs. b) Ratemaps, alongside spatial connectivity. Shown is the outgoing recurrent weight, as a function of unit spatial phase. Also inset is the convex hull of all ensemble phases. Red indicates excitation (positive weight), blue inhibition (negative weight). c) Evaluation of recurrent (top) and feedforward units (bottom) units when the network is the environment is extended beyond the original training regime (white square). For the RNN, representations are shown of trajectories starting within the training enclosure, and both inside and outside.

neural networks, can emerge when optimized to preserve local spatial distances under a capacity constraint. This aligns with previous theories positing that grid cells contribute to spatial navigation by providing a consistent metric for space [Moser and Moser, 2008, Stemmler et al., 2015, Ginosar et al., 2023, Schøyen et al., 2024].

Our results challenge the widely-held view that grid cells are integral to path integration. In our experiments, feed-forward networks, which lack an explicit path integration mechanism, still developed grid-like representations comparable to those in the path-integrating recurrent network. This observation suggests that the emergence of grid cells is not contingent on the process of path integration. Instead, it indicates that grid cells may primarily function to encode spatial relationships.

Pruning velocity inputs further allowed us to disentangle the contributions of different cell types to path integration. By comparing the representations of a pruned network to a non-pruned network, we could directly ascertain that path integration was not critically dependent on grid cells, but seemingly rather on band-type units, echoing other recent findings [Schøyen et al., 2023]. The inverse correlation between the contribution to path integration and the mean sample grid score further reinforces the notion that grid cells might not be as crucial for path integration as previously thought.

In summary, our findings suggests a reevaluation of the role of grid cells within the neural navigation framework. While grid cells clearly provide a powerful spatial metric, their necessity for path integration is questionable. This insight not only advances our understanding of grid cell functionality but also prompts a reconsideration of how spatial representations are constructed and utilized in both biological and artificial systems. Future research should continue to explore the interplay between different cell types in the entorhinal cortex, as well as investigate how these findings can inform the development of more efficient and interpretable models of spatial navigation.

### 3.2 Limitations

While our work provides evidence supporting the role of grid cells as a high capacity distance preserving representation, our model operates in a simplified domain, with Cartesian coordinate inputs. One could therefore consider extending the recurrent model to include e.g. more expressive input projections (such as [Schaeffer et al., 2023]), or use other self-motion information, such as simulated head direction cell [Taube et al., 1990] input.

From a biological perspective, the use of label distances during training is also implausible. However, we have begun to explore a sampling-based alternative to the current training regime, which might be more transferable to learning in biological systems. A related limitation is the use of Euclidean distances in computing the loss function. While this might be sufficient in an open arena such the one used in this work, more naturalistic settings, with e.g. interior walls may require other distance functions to capture the observed deformation of grid patterns in e.g. exotic geometries [Ginosar et al., 2023]. Using the Euclidean distance between state vectors may also lead to inaccuracies when computing the loss. As an alternative, one could compute state distances using the metric induced by the neural network model. However, when comparing over smaller distances (which are implicitly enforced by $\sigma$), the Euclidean distance function could still be a decent approximation.

## References


Andrea Banino, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, Martin J. Chadwick, Thomas Degris, Joseph Modayil, Greg Wayne, Hubert Soyer, Fabio Viola, Brian Zhang, Ross Goroshin, Neil Rabinowitz, Razvan Pascanu, Charlie Beattie, Stig Petersen, Amir Sadik, Stephen Gaffney, Helen King, Koray Kavukcuoglu, Demis Hassabis, Raia Hadsell, and Dharshan Kumaran. Vector-Based Navigation Using Grid-like Representations in Artificial Agents. *Nature*, 557 (7705):429–433, May 2018. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-018-0102-6. URL http://www.nature.com/articles/s41586-018-0102-6.

Yoram Burak and Ila R. Fiete. Accurate Path Integration in Continuous Attractor Network Models of Grid Cells. *PLoS Computational Biology*, 5(2):e1000291, February 2009. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1000291. URL https://dx.plos.org/10.1371/journal.pcbi.1000291.

Daniel Bush, Caswell Barry, Daniel Manson, and Neil Burgess. Using Grid Cells for Navigation. *Neuron*, 87(3): 507–520, August 2015. ISSN 08966273. doi: 10.1016/j.neuron.2015.07.006. URL https://linkinghub.elsevier.com/retrieve/pii/S0896627315006285.

Christopher J. Cueva and Xue-Xin Wei. Emergence of Grid-like Representations by Training Recurrent Neural Networks to Perform Spatial Localization. arXiv: 1803.07770, March 2018. URL http://arxiv.org/abs/1803.07770.

William Dorrell, Peter E. Latham, Timothy E. J. Behrens, and James C. R. Whittington. Actionable Neural Representations: Grid Cells from Minimal Constraints, September 2022. URL http://arxiv.org/abs/2209.15563. arXiv:2209.15563 [q-bio].

Richard J. Gardner, Erik Hermansen, Marius Pachitariu, Yoram Burak, Nils A. Baas, Benjamin A. Dunn, May-Britt Moser, and Edvard I. Moser. Toroidal Topology of Population Activity in Grid Cells. *Nature*, 602 (7895):123–128, February 2022. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-021-04268-7. URL https://www.nature.com/articles/s41586-021-04268-7.

Mariana Gil, Mihai Ancau, Magdalene I. Schlesiger, Angela Neitz, Kevin Allen, Rodrigo J. De Marco, and Hannah Monyer. Impaired path integration in mice with disrupted grid cell firing. *Nature Neuroscience*, 21(1):81–91, December 2017. doi: 10.1038/s41593-017-0039-3. URL https://doi.org/10.1038/s41593-017-0039-3. Publisher: Springer Science and Business Media LLC.

Gily Ginosar, Johnatan Aljadeff, Liora Las, Dori Derdikman, and Nachum Ulanovsky. Are Grid Cells Used for Navigation? On Local Metrics, Subjective Spaces, and Black Holes. *Neuron*, 2023. ISSN 0896-6273. doi: 10.1016/j.neuron.2023.03.027. URL https://www.sciencedirect.com/science/article/pii/S0896627323002234.

Torkel Hafting, Marianne Fyhn, Sturla Molden, May-Britt Moser, and Edvard I. Moser. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–806, 2005. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature03721. URL http://www.nature.com/articles/nature03721.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. arXiv: 1412.6980 [cs], January 2017. URL http://arxiv.org/abs/1412.6980.

Julija Krupic, Neil Burgess, and John O'Keefe. Neural Representations of Location Composed of Spatially Periodic Bands. *Science*, 337(6096):853–857, August 2012. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1222403. URL https://www.science.org/doi/10.1126/science.1222403.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction, 2020. _eprint: 1802.03426.

Bruce L. McNaughton, Francesco P. Battaglia, Ole Jensen, Edvard I Moser, and May-Britt Moser. Path integration and the neural basis of the 'cognitive map'. *Nature Reviews Neuroscience*, 7(8):663–678, August 2006. ISSN 1471-003X, 1471-0048. doi: 10.1038/nrn1932. URL https://www.nature.com/articles/nrn1932.

Edvard I. Moser and May-Britt Moser. A metric for space. *Hippocampus*, 18(12):1142–1156, December 2008. ISSN 1050-9631, 1098-1063. doi: 10.1002/hipo.20483. URL https://onlinelibrary.wiley.com/doi/10.1002/hipo.20483.

Dounia Mulders, Man Yi Yim, Jae Sung Lee, Albert K. Lee, Thibaud Taillefumier, and Ila R. Fiete. A structured scaffold underlies activity in the hippocampus, November 2021. URL http://biorxiv.org/lookup/doi/10.1101/2021.11.20.469406.

Aran Nayebi, Alexander Attinger, Malcolm Campbell, Kiah Hardcastle, Isabel Low, Caitlin S Mallory, Gabriel Mel, Ben Sorscher, Alex H Williams, Surya Ganguli, Lisa Giocomo, and Dan Yamins. Explaining heterogeneity in medial entorhinal cortex with task-driven neural networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 12167–12179. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/656f0dbf9392657eed7feefc486781fb-Paper.pdf.

Rylan Schaeffer, Mikail Khona, Tzuhsuan Ma, Cristóbal Eyzaguirre, Sanmi Koyejo, and Ila Rani Fiete. Self-Supervised Learning of Representations for Space Generates Multi-Modular Grid Cells. 2023. doi: 10.48550/ARXIV.2311.02316. URL https://arxiv.org/abs/2311.02316. Publisher: arXiv Version Number: 1.

Vemund Schøyen, Markus Borud Pettersen, Konstantin Holzhausen, Marianne Fyhn, Anders Malthe-Sørenssen, and Mikkel Elle Lepperød. Coherently remapping toroidal cells but not Grid cells are responsible for path integration in virtual agents. *iScience*, 26(11):108102, November 2023. ISSN 25890042. doi: 10.1016/j.isci.2023.108102. URL https://linkinghub.elsevier.com/retrieve/pii/S258900422302179X.

Vemund Schøyen, Constantin Bechkov, Markus Borud Pettersen, Erik Hermansen, Konstantin Holzhausen, Anders Malthe-Sørenssen, Marianne Fyhn, and Mikkel Elle Lepperød. Hexagons all the way down: Grid cells as a conformal isometric map of space. preprint, Neuroscience, February 2024. URL http://biorxiv.org/lookup/doi/10.1101/2024.02.02.578585.

Ben Sorscher, Gabriel C. Mel, Samuel A. Ocko, Lisa M. Giocomo, and Surya Ganguli. A unified theory for the computational and mechanistic origins of grid cells. *Neuron*, page S0896627322009072, October 2022. ISSN 08966273. doi: 10.1016/j.neuron.2022.10.003. URL https://linkinghub.elsevier.com/retrieve/pii/S0896627322009072.

Martin Stemmler, Alexander Mathis, and Andreas V. M. Herz. Connecting Multiple Spatial Scales to Decode the Population Activity of Grid Cells. *Science Advances*, 1(11):e1500816, December 2015. ISSN 2375-2548. doi: 10.1126/science.1500816. URL https://www.science.org/doi/10.1126/science.1500816.

Js Taube, Ru Muller, and Jb Ranck. Head-Direction Cells Recorded from the Postsubiculum in Freely Moving Rats. I. Description and Quantitative Analysis. *The Journal of Neuroscience*, 10(2):420–435, February 1990. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.10-02-00420.1990. URL https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.10-02-00420.1990.

The Astropy Collaboration. The Astropy Project: Sustaining and Growing a Community-oriented Open-source Project and the Latest Major Release (v5.0) of the Core Package, June 2022. URL http://arxiv.org/abs/2206.14220. arXiv:2206.14220 [astro-ph].

Christopher Tralie, Nathaniel Saul, and Rann Bar-On. Ripser.py: A lean persistent homology library for python. *The Journal of Open Source Software*, 3(29):925, Sep 2018. doi: 10.21105/joss.00925. URL https://doi.org/10.21105/joss.00925.

James C.R. Whittington, Timothy H. Muller, Shirley Mark, Guifen Chen, Caswell Barry, Neil Burgess, and Timothy E.J. Behrens. The Tolman-Eichenbaum Machine: Unifying Space and Relational Memory through Generalization in the Hippocampal Formation. *Cell*, 183(5):1249–1263.e23, November 2020. ISSN 00928674. doi: 10.1016/j.cell.2020.10.024. URL https://linkinghub.elsevier.com/retrieve/pii/S009286742031388X.

Dehong Xu, Ruiqi Gao, Wen-Hao Zhang, Xue-Xin Wei, and Ying Nian Wu. Conformal Isometry of Lie Group Representation in Recurrent Network of Grid Cells, 2022. _eprint: 2210.02684.

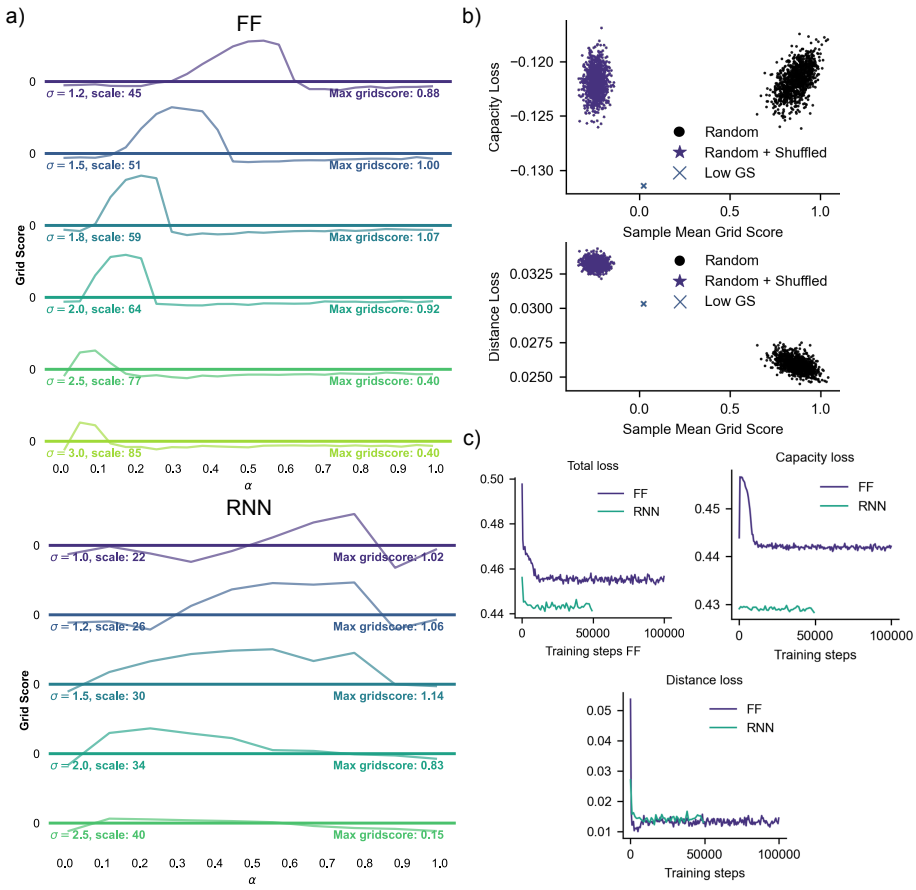


Figure 4: **Hyperparameter search, loss evaluation and training results**. a) Grid score as a function of $\sigma$ and $\alpha$ for FF (top) and RNN models (bottom), where grid scale is calculated for the model with the highest grid score at a particular $\sigma$. b) Loss evaluation on 1000 randomly selected (random), spatially shuffled RNN unit ratemaps (random + shuffled) and low grid score (low GS) subpopulations. c) Loss training history for FF and RNN models.

# A Appendix / supplemental material