Unfolding the Black Box of Recurrent Neural Networks for Path Integration

Tianhao Chu[1,†], Yuling Wu[2,†], Neil Burgess[3,4], Zilong Ji[1,3], Si Wu[1,2,5,*]

[1]School of Psychological and Cognitive Sciences, Key Laboratory of Machine Perception (Ministry of Education), Peking University, China.
[2] Peking-Tsinghua Center for Life Sciences, Academy for Advanced Interdisciplinary Studies, Peking University, China.
[3]Institute of Cognitive Neuroscience, Department of Neuroscience, Physiology and Pharmacology, University College London, UK
[4]UCL Queen Square Institute of Neurology, University College London, UK
[5]Beijing Key Laboratory of Behavior and Mental Health, IDG/McGovern Institute for Brain Research, Center of Quantitative Biology, Peking University, China.
[†]These authors contributed equally to this work.
[*]Correspondence: siwu@pku.edu.cn

Abstract

Path integration is essential for spatial navigation. Experimental studies have identified neural correlates for path integration, but exactly how the neural system accomplishes this computation remains unresolved. Here, we adopt recurrent neural networks (RNNs) trained to perform a path integration task to explore this issue. After training, we borrow neuroscience prior knowledge and methods to unfold the black box of the trained model, including: clarifying neuron types based on their receptive fields, dissecting information flows between neuron groups by pruning their connections, and analyzing internal dynamics of neuron groups using the attractor framework. Intriguingly, we uncover a hierarchical information processing pathway embedded in the RNN model, along which velocity information of an agent is first forwarded to band cells, band and grid cells then coordinate to carry out path integration, and finally grid cells output the agent location. Inspired by the RNN-based study, we construct a neural circuit model, in which band cells form one-dimensional (1D) continuous attractor neural networks (CANNs) and serve as upstream neurons to support downstream grid cells to carry out path integration in the 2D space. Our study challenges the conventional view of considering grid cells as the principal velocity integrator, and supports a neural circuit model with the hierarchy of band and grid cells.

1 Introduction

Path integration refers to the process of computing the position of an agent by continuously integrating self-motion information of the agent over time [1]. It is fundamental for spatial navigation in both humans and rodents, and also plays an important role in robot control [2].

Experimental studies have shown that grid cells in the medial entorhinal cortex (MEC) are deeply involved in path integration [3]. These cells exhibit periodic hexagonal firing patterns in the two-dimensional space, providing a metric representation for implementing path integration [1, 4]. Disruption of grid cells through medial septal inactivation or entorhinal lesions impairs spatial memory and navigation based on path integration [3, 5]. Despite these experimental evidence,

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

exactly how the neural system performs path integration remains largely unknown, and there are unresolved issues. First, in addition to grid cells, experimental data has shown that there exists a rich diversity of spatially modulated cell types in the MEC [6, 7, 8, 9]. In particular, band cells, found in parasubiculum (an upstream region of MEC), exhibit periodic firing patterns that can be modeled as a superposition of bands, and they are hypothesized to perform path integration in a 1D space [9, 10]. However, the functional relationship between grid and band cells remains to be clarified. Second, neural circuit models have been proposed to implement path integration. The conventional models consider only grid cells, and in order to implement 2D path integration, these models either consider a large amount of direction-conjunctive grid cells [11, 12, 13] or assume that the recurrent connections between grid cells are tuned instantly by the velocity signal [14]. A different model considering the hierarchy of band and grid cells for implementing path integration was proposed [15, 16]. The question of which model is biologically more plausible remains debated.

Related RNN-based navigated studies. Recently, AI approaches have been used to study the association between path integration and grid cells. Specifically, recurrent neural networks (RNNs) were trained to perform spatial navigation to ascertain the involvement of grid cells in path integration, and inconclusive results have been reported so far. In one line of works, it was found that training an RNN based solely on motion velocity in a supervised way (i.e., reconstructing the place cell activity at the given location from pure velocity input) can lead to the emergence of grid-cell-like activity pattern in the RNN, especially under the biological constraints such as non-negative firing rate [17, 18]. This result implies that grid cells are associated with path integration. Building on this work, subsequent studies have extended the supervised training paradigm to more unsupervised-guided ones, using objective functions derived from group theory, information efficiency, and actionability, emphasizing the isometric structure, high coding capacity, and integrative functionality of grid cells [19, 14, 20]. These studies have systematically explored the conditions under which grid-like representations emerge in RNNs, examining how the network structure, the training objective, and biological priors jointly shape grid-like representations. In the other line of works, the necessity of grid cells for path integration is challenged. For instance, [21] demonstrated that grid-like firing patterns can arise as the principal components of place cell activities under the non-negativity constraint, without path integration involved. More recently, [22] argued that grid cells may emerge from a principle of efficient pattern formation for reconstructing place fields, rather than from the demand of path integration. Supporting this view, ablation studies of trained RNNs have shown that removing velocity inputs to high grid-score units has negligible impact on the path integration performance [23, 24]. Overall, current RNN-based studies have primarily focused on exploring the conditions under which grid cells emerge and on the necessity of grid cells for path integration. However, they have not addressed the question of how a trained RNN implements path integration, nor examined the functional role of band cells in this process [18, 25]. The internal mechanisms of the trained RNN model thus remain a black box.

In this work, we take the same approach of training RNNs to perform path integration, but our goal is different, that is, we aim to unveil the detailed mechanism concealed in the trained RNNs, and from which, to ascertain the neural mechanism of path integration. To achieve this goal, we will borrow neuroscience prior knowledge and methods to unfold the black box of the trained RNNs in much more detail than in previous studies. These include: clarifying neuron types based on their receptive fields, dissecting information flows between neuron groups by pruning their connections, and analyzing internal dynamics of neuron groups in the attractor framework. It turns out that there exists a hierarchical information processing pathway embedded in the RNN model, along which velocity information is first forwarded to band cells, band and grid cells then coordinate to carry out path integration, and finally grid cells output the location of the agent. Inspired by the RNN-based study and previous mechanistic models [16, 26], we construct a hierarchical neural circuit model for path integration in the brain, in which band and grid cells form 1D and 2D continuous attractor neural networks (CANNs), respectively, and they coordinate to accomplish path integration. Contemporary models often assume that grid cells are primarily responsible for path integration [11, 12, 13]. Our study challenges this view and suggests that the hierarchy of band and grid cells is necessary. Our model also makes predictions testable in neuroscience experiments.

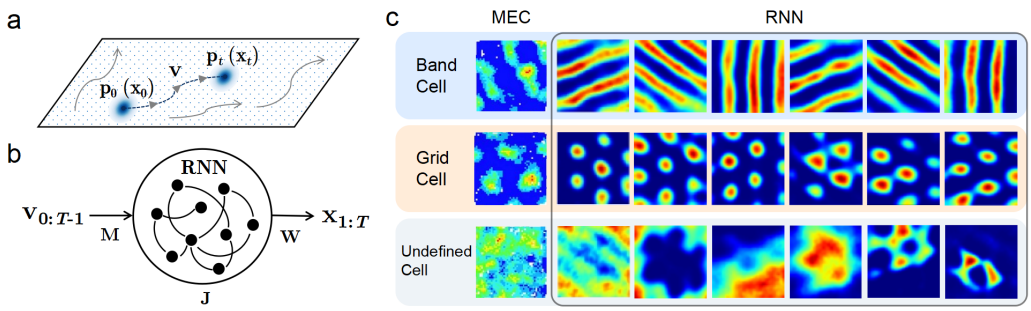

Figure 1: (a) Illustration of the path integration task. (b) The structure of the RNN model. (c) Examples of band cell, grid cell and undefined cell found in MEC (middle panel, adapted from [9]) and in the trained RNN (right panel).

2 Neuroscience for AI: unfolding the black box

2.1 An RNN model for path integration

We adopt a well-established paradigm to train RNNs for path integration [17, 22], obtaining the key results as reported here, which are further validated by consistent results from other modeling frameworks (see Supplement Information (SI) E). As illustrated in Fig. 1a, an agent is moving in a square environment with velocity $\mathbf{v}_t \in \mathbb{R}^2$. Mimicking place cells' position encoding in the neural system, the agent's location $\mathbf{x}_t$ is represented by a bump vector $\mathbf{p}_t \in \mathbb{R}^{N_p}$ in the 2D space centered at $\mathbf{x}_t$, with N_p the number of place cells. The task of path integration is formulated as: given the initial location $\mathbf{x}_0$ of the agent, the RNN model infers its subsequent positions $\mathbf{x}_{1:T}$ by using only the self-motion information of the agent, i.e., the sequence of velocity vectors $\mathbf{v}_{0:T-1}$; no other position cue is available. Within the conceptual framework of the MEC-HPC loop, the model's information flow can be interpreted as follows: the initial state of the RNN (representing a nascent MEC population) is set by a projection from the place cell population (representing hippocampus, HPC), i.e., $\mathbf{r}_0 = \mathbf{M}_0\mathbf{p}_0$, where $\mathbf{p}_0$ corresponds to the initial location $\mathbf{x}_0$ and the matrix $\mathbf{M}_0 \in \mathbb{R}^{N_R \times N_p}$, with N_R the number of neurons in the RNN (Fig. 1b). This initialization embodies a critical **HPC → MEC projection** signal that anchors the path-integration process to an environmental cue.

At each time step, the RNN receives the agent's instantaneous velocity signal $\mathbf{v}_t$ and updates its internal state following the recurrent dynamics, which is written as,

$$\mathbf{r}_{t+1} = \sigma\left(\mathbf{J}\mathbf{r}_t + \mathbf{M}\mathbf{v}_t\right), \tag{1}$$

where $\mathbf{J} \in \mathbb{R}^{N_R \times N_R}$ is the recurrent connection matrix between neurons in the RNN, the matrix $\mathbf{M} \in \mathbb{R}^{N_R \times 2}$, and $\sigma(\cdot)$ is the ReLU function, enforcing non-negative firing rates of neurons.

At each time step, the agent's location is read out by,

$$\hat{\mathbf{p}}_t = \mathbf{W}\mathbf{r}_t. \tag{2}$$

where the read-out matrix $\mathbf{W} \in \mathbb{R}^{N_p \times N_R}$.

The free parameters of a RNN, including $\mathbf{J}$, $\mathbf{M}$, $\mathbf{W}$, and $\mathbf{M}_0$, are optimized using BPTT through minimizing the cross-entropy loss between the network outputs $\hat{\mathbf{p}}_t$ and the ground truth $\mathbf{p}_t$ for all time steps $t = 1, \ldots, T$ and for all randomly generated motion trajectories. The details of the model training and fixed parameters are given in SI A. The trained RNN model effectively implements path integration, with quantitative results (SI D) indicating that it achieved competent performance on the task.

2.2 Interpreting the trained RNN model with neuroscience

It is straightforward to train the RNN model well to achieve a good path integration performance, but unveiling the underlying computational mechanism is challenging, which is known as the black box problem of AI-trained models. Without prior knowledge, we have no clue to interpret the data. Here, we borrow neuroscience prior knowledge and methods to unfold the black box.

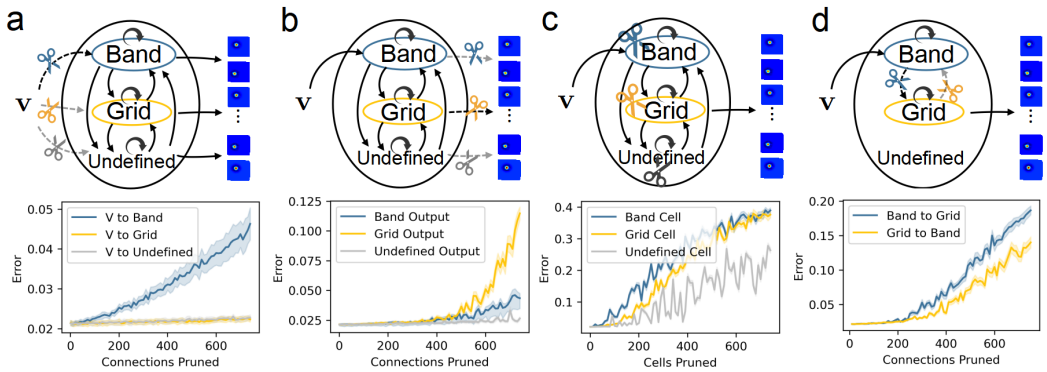

Figure 2: Results of pruning the RNN model. Four pruning operations are carried out step-by-step, with each step building upon the previous one. Upper panels: (a) Pruning velocity inputs to each cell group one-by-one; (b) Pruning read-out connections from each cell group one-by-one; (c) Pruning each cell type one-by-one; (d) Pruning the connections from band to grid and the connections from grid to band cells, respectively. Lower panels: Path integration error vs. the number of connections/cells pruned in each step. The path integration error is measured by the standard division of location prediction errors divided by the arena size. For calculation details, see SI C, E.

2.2.1 Clarifying neuron types

Our first step is to clarify neuron types in the trained RNN based on their receptive fields. Following the common practice in neuroscience, we first compute the spatial tuning maps of neurons in the trained RNN. Specifically, we consider that the agent performs random walks in the environment, covering the whole arena, and the corresponding velocity signals are used as inputs to stimulate neuronal responses in the RNN. We record the activities of each neuron at all locations, which gives a 2D heatmap for each neuron, reflecting their spatial tuning characteristics.

Inspired by the spatial tuning properties of grid and band cells observed in the neural system, we search for analogous unit types in the trained RNN. To classify neuron types quantitatively, we adopt established criteria from neuroscience (see SI B). We have identified three mutually exclusive types based on neuronal receptive fields (Fig. 1c, right panel), which are: 1) Grid cell, which exhibits the hexagonally patterned firing field; 2) Band cell, whose receptive field consists of multiple equally spaced parallel lines (the so-called bands) (see more details below); and 3) Undefined cell, which displays neither grid-like nor band-like spatial tuning. All three cell types emerge robustly in the trained RNNs (see SI E). For example, in one trained RNN with $N_R = 4096$ units, we identify 764 grid cells, 764 band cells, and 2568 undefined cells).

2.2.2 Dissecting information flows between neuron groups

After identifying neuron groups, we further investigate how information is propagated between them by applying pruning studies. We carry out four step-by-step pruning operations (see SI C), with each of them building upon the previous one and aiming to probe one aspect of path integration. See SI E for reproducibility results.

Step 1: Pruning velocity inputs. To identify which cell types are primarily responsible for receiving and processing velocity inputs, we selectively prune the afferent velocity inputs to each cell group one-by-one (Fig. 2a, upper panel). We find that pruning velocity inputs to band cells leads to a significant increase in the path integration error, while pruning velocity inputs to other groups has little effect (Fig. 2a, lower panel). This indicates that the velocity information is predominantly processed by band cells, and we remove velocity inputs to other cell types in the followed pruning.

Step 2: Pruning read-out connections. To identify which cell types are primarily responsible for outputting the path integration result, we selectively prune the read-out connections from each cell group one-by-one (Fig. 2b, upper panel), and find that pruning the read-out connections from grid cells have the largest effect on increasing the path integration error (Fig. 2b, lower panel). This

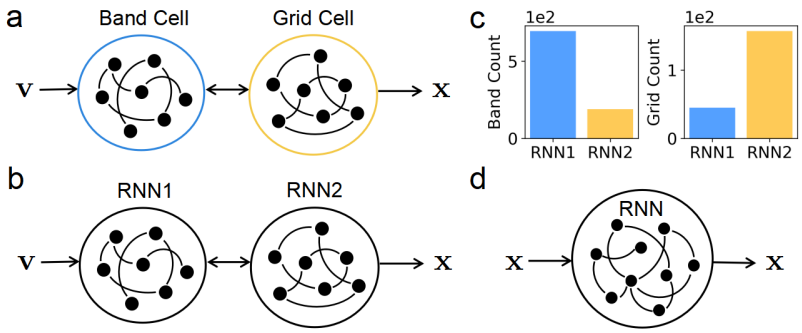

Figure 3: (a) The hierarchical path integration pathway embedded in trained RNNs. (b) A new RNN model taking the hierarchical pathway as a prior. (c) Neuron types found in the newly trained RNN. (d) An RNN model for a location reconstruction task. For simulation details, see SI C.

indicates that grid cells are predominately responsible for outputting the path integration result, and we remove the read-out connections from other cell types in the followed pruning.

Step 3: Pruning cell types. To evaluate the relative contributions of different cell types on path integration, we compare the model performances after pruning one of them in equal numbers (Fig. 2c, upper panel). We find that pruning band or grid cells causes a substantial drop in the model performance, whereas pruning undefined cells has a much less effect (Fig. 2c, lower panel). This suggests that band and grid cells play much more important roles than undefined cells in path integration, and we remove them in the followed pruning.

Step 4: Pruning connections between band and grid cells. To probe the functional relationship between band and grid cells in path integration, we separately prune the connections from band to grid cells and the connections from grid to band cells (Fig. 2d, upper panel). We find that pruning either type of connections has a big impact on the model performance (Fig. 2d, lower panel), indicating that the bidirectional, cooperative interactions between two cell types are critical for path integration.

2.3 A hierarchical path integration pathway embedded in the RNN

Combining pruning results while discarding less relevant components, we identify a hierarchical path integration pathway embedded in the trained RNN (Fig. 3a), along which velocity information is first received and processed by band cells; band and grid cells then interact via reciprocal connections to integrate movement information; finally grid cells output the agent location in the 2D space.

It is surprising that such a well-organized pathway naturally emerges in training a RNN for a path integration task. To further validate that this hierarchical pathway constitutes the functional core structure, we conduct two additional experiments. In one experiment, we construct a new RNN model which takes the hierarchical pathway as a prior (Fig. 3b), that is, the first module receives velocity inputs, intending to play the role of band cells; the second module outputs the agent location, intending to play the role of grid cells; the two modules are reciprocally connected, intending to mimic the interactions between band and grid cells. We train this new model with the same path integration task and have two key observations as expected, which are: 1) the new model achieves a performance comparable to the original model; 2) band and grid cells emerge naturally as the dominant neuron types in the first and second modules, respectively (Fig. 3c). In the other experiment, we change the task from path integration to pure location reconstruction (Fig. 3d), i.e., the RNN receives the agent's location inputs and reconstruct them; no velocity information is available. After training the RNN with the new task, we observe only grid cells without band cells (see SI C), indicating the necessity of path integration for the emergence of band cells. Together, these two additional experiments further reinforce our finding that the hierarchical pathway, characterized by the hierarchy of band and grid cells, and their reciprocal interactions, is the core structure for path integration.

2.3.1 Continuous attractor dynamics of band cells

We now delve deeper into the internal structure and dynamics of band cells in the trained RNN.

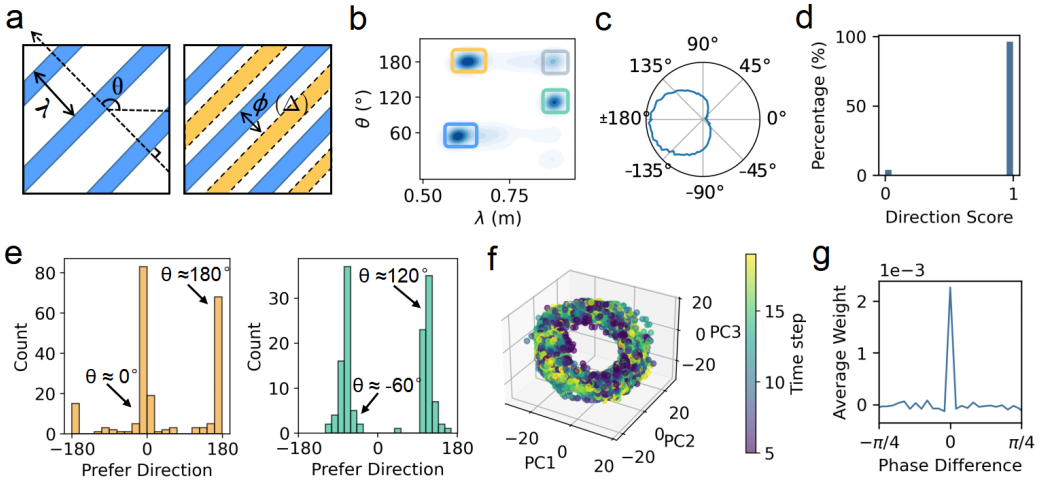

Figure 4: Properties of band cells in the trained RNN. (a) Illustration of the receptive field of an idealized band cells. Left panel: the receptive field consists of parallel bands, which are equally spaced with spacing λ and have the same orientation θ. Right panel: the positional separation between two band cells having the same spacing and orientation (blue and yellow) is quantified by a phase ϕ, whose value is given by $\phi = (\Delta/\lambda) \times 2\pi$, with Δ the offset between the adjacent bands of two band cells. (b) Band cells in the trained RNN are grouped into four clusters according to their spacing λ and orientation θ. (c) An example band cell with direction-tuning at π. (d) Distribution of directional scores (a measurement of the direction-tuning level) of band cells in an example cluster. (e) Histograms of preferred directions of band cells in two example clusters, which have a bimodal shape peaked at either the cluster orientation θ or its opposite direction $\theta + \pi$. (f) A 3D isomap embedding of the population activities of band cells in an example group with the same spacing and direction tuning. Coloured by time steps 5–20 of 200 trajectories. (g) The averaged connection profile of band cells as a function of their phase difference. For detail calculations, see SI B.

Fig. 4a displays the receptive field of an idealized band cell, which consists of multiple equally spaced parallel lines (bands), on which the cell's response is invariant. A band cell can be characterized by three parameters: 1) spacing λ, which is the distance between adjacent bands; 2) orientation θ, which is the orientation of bands in the 2D space; 3) phase ϕ, which defines the positional separation between band cells having the same spacing and orientation. Denote $\Delta \in (-\lambda/2, \lambda/2)$ the offset between the adjacent bands of two band cells, their phase difference is given by $\phi = (\Delta/\lambda) \times \pi \in (-\pi, \pi)$.

First, we inspect the clustering property of band cells in the trained RNN. Using Fourier analysis, we extract the dominant spatial frequency of the receptive field of each band cell and obtain the cell's spacing and orientation (see SI B for details). We find that band cells can be grouped into four well-separated clusters based on their spacing and orientation (Fig. 4b). For each cluster, we further differentiate cells' direction-tuning by calculating how the averaged response of a band cell varies with motion direction of the agent. Interestingly, we find that in each cluster, the majority of band cells exhibit clear direction-tuning (Fig. 4c-d), and they constitute two sub-groups with preferred directions along either the cluster orientation θ or its opposite $\theta + \pi$ (Fig. 4e).

Second, inspired by recent successes in modeling spatially tuned neurons in the hippocampal-entorhinal system using Continuous Attractor Neural Networks (CANNs) [27, 28, 29, 11], we set out to investigate whether the low-dimensional manifold of neural activity and its underlying connectivity conform to CANN principles. Second, to uncover the latent structure and dynamics of band cells in each group having the same spacing and direction-tuning, we performed dimensionality reduction dimensionality reduction analysis (Isomap, details shown in SI B) on the activity trajectories of the group during path integration. Strikingly, we observed that the low-dimensional embedding of neural trajectories in each group forms a continuous ring (Fig. 4f), consistent with the dynamics of a 1D CANN with periodicity. We also inspected the recurrent connectivity pattern between band cells, looking for the signature of the CANN structure. For each band cell group having the same spacing and direction-tuning, we calculated the averaged connection weights between cell pairs

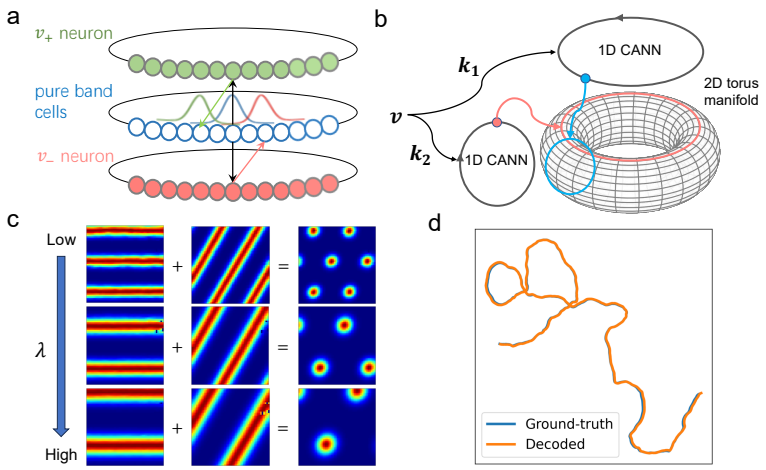

Figure 5: A neural circuit model for path integration. (a) The structure of a band cell module, which consists of three sub-populations: pure band cells, direction-conjunctive band cells v^+, and direction-conjunctive band cells v^-. A band cell module performs 1D path integration in the form of continuous attractor dynamics, along either the module orientation $v^+ = \theta$ or the opposite direction $v^- = \theta + \pi$. (b) Two band cells modules with different orientations support the grid cell module of the 2D CANN structure to accomplish path integration in the 2D space. (c) Examples of band and grid cells in the neural circuit model. (d) Inferred trajectory of the neural circuit model closely matches the ground-truth trajectory, demonstrating the path integration capability of the model.

as a function of their phase differences. The result shows that the connectivity profile exhibits an approximately symmetric shape with weight decreasing with phase difference (Fig. 4g), reminiscent of the center-surround connection pattern in a 1D CANN [30].

In summary, the above study reveals that band cells in the trained RNN constitute modules of different spacing and direction-tuning, and each module displays the 1D CANN-like structure and dynamics.

3 AI for neuroscience: the neural mechanism for path integration

The above RNN-based navigation study also gives us insight into the path integration mechanism in the neural system. Band cells were reported in a neuroscience experiment [9], and an early computational model [15, 16] (in which band cells were called Velocity-Controled-Oscillator (VCO)) was also proposed, which considered that band cells are responsible for 1D path integration, and multiple modules of band cells of different orientations jointly support grid cells to execute path integration in the 2D space. Our RNN-based study tends to support this view, and challenges the conventional models of relying only on grid cells for path integration.

3.1 A neural circuit model for path integration

Inspired by the findings in the trained RNN, we construct a neural circuit model for path integration. It has the similar idea as that in [15, 16], but have different mathematical formulations. In the below, we introduce the main structure of the model, with details presented in SI F.

3.1.1 The modules of band cells

Motivated by the properties of band cells found in the trained RNN and the 1D head-direction system in Drosophila [31], we propose a theoretical model for band cells. Specifically, we consider that band cells are clustered into multiple modules with varying spacing λ and orientation θ. Each module consists of three sub-populations (Fig. 5a), which are: 1) pure band cells, 2) direction-conjunctive band cells v^+, and 3) direction-conjunctive band cells v^-, and they are aligned on a 1D manifold according to their preferred phases $\phi \in (-\pi, \pi]$. Each module of band cells implements path

integration in a 1D direction, along either the module orientation $v^+ = \theta$ or the opposite direction $v^- = \theta + \pi$.

The pure band cell population forms a 1D CANN, whose dynamics is given by,

$$\tau \frac{\partial u_b(\phi,t)}{\partial t} = -u_b(\phi,t) + \rho \int_{-\pi}^{\pi} J_b(\phi,\phi')\, r_b(\phi',t)\, d\phi' + \rho \sum_{m=\pm} \int_{-\pi}^{\pi} W_m(\phi,\phi')\, r_m(\phi',t)\, d\phi' + I_g(\phi,t), \tag{3}$$

where $u_b(\phi,t)$ and $r_b(\phi,t)$ denote the synaptic input and firing rate of pure band cells at ϕ, respectively. $J_b(\phi,\phi')$ denotes the recurrent connections between band cells, and W_m, with $m = \pm$, denotes the connections from direction-conjunctive cells v^m to pure band cells. I_g denotes the input from grid cells (to be defined below). The firing rate of pure band cells is given by $r_b(\phi,t) = u_b^2(\phi,t) / \left[1 + k_b \rho \int_{-\pi}^{\pi} u_b^2(\phi',t)\, d\phi'\right]$, with k_b regulating the global inhibitory strength to ensure a stable bump activity of band cells.

Conjunctive band cells $v^\pm$ receive excitatory inputs from pure band cells and are tuned by the moving direction of the agent, whose dynamics are written as,

$$\tau \frac{\partial u_\pm(\phi,t)}{\partial t} = -u_\pm(\phi,t) + w_b \cdot r_b(\phi,t), \tag{4}$$

$$r_\pm(\phi,t) = \left[g_0 + (\mathbf{v}(t)\cdot\mathbf{k}) u_\pm(\phi,t)\right]_+, \tag{5}$$

where $u_\pm(\phi,t)$ and $r_\pm(\phi,t)$ denote the synaptic input and firing rate of conjunctive cells at phase ϕ, respectively. w_b denotes the connection from pure band to conjunctive cells at the same phase. $\mathbf{k} = (\sin\theta/\lambda, \cos\theta/\lambda)$ denotes a module-specific vector, and the projection of the agent velocity $\mathbf{v}(t)$ on it, $\mathbf{v}(t)\cdot\mathbf{k}$, defines the moving speed of the agent along the orientation θ. g_0 is a baseline constant. $[\cdot]_+$ denotes rectification. Eq.5 reflects that the firing rates of conjunctive cells contain the projected 1D speed information of the agent along the module orientation [31].

Importantly, we set the recurrent connections between pure band cells to be symmetric and translation-invariant, i.e., $J_b(\phi,\phi') = J_b^0 / \left(\sqrt{2\pi}\sigma_b\right) \exp\left[-|\phi-\phi'|_\pi^2/(2\sigma_b^2)\right]$, while the connections from conjunctive cells to pure band cells are offset with a constant value δ, i.e., $W_\pm(\phi,\phi') = W_b^0/(\sqrt{2\pi}\sigma_b) \exp\left[-|\phi-\phi' \mp \delta|_\pi^2/(2\sigma_b^2)\right]$. Together, three populations of band cells in a module achieve 1D path integration along the module orientation [31].

3.1.2 The module of grid cells

A single band cell module supports path integration in two directions θ and $\theta+\pi$. To realize 2D path integration, grid cells integrate the outputs of band cell modules of different orientations. With loss of generality, we consider only two band cell modules in the present study (Fig. 5b). The orientations of two modules are denoted as θ_1 and θ_2, respectively, and they are 60° apart. Grid cells are aligned on a 2D toroidal manifold with a phase vector$\boldsymbol{\phi} = (\phi_1, \phi_1)$. They form a 2D CANN with dynamics given by,

$$\tau_g \frac{\partial u_g(\boldsymbol{\phi},t)}{\partial t} = -u_g(\boldsymbol{\phi},t) + \iint_{-\pi}^{\pi} J_g(\boldsymbol{\phi},\boldsymbol{\phi}')\, r_g(\boldsymbol{\phi}',t)\, d\boldsymbol{\phi}' + I_1(\phi_1,t) + I_2(\phi_2,t), \tag{6}$$

where $u_g(\boldsymbol{\phi},t)$ and $r_g(\boldsymbol{\phi},t)$ are the synaptic input and firing rate of grid cells at phase $\boldsymbol{\phi}$. The recurrent connections between grid cells are symmetric and translation-invariant, i.e., $J_g(\boldsymbol{\phi},\boldsymbol{\phi}') = J_0^g/(2\pi\sigma_g^2) \exp\left[-||\boldsymbol{\phi}-\boldsymbol{\phi}'||_g^2/(2\sigma_g^2)\right]$, with $||\cdot||_g$ denotes circular distance on the torus and σ_g controlling the bump activity size. The firing rate of grid cells also follows divisive normalization, i.e., $r_g(\boldsymbol{\phi},t) = u_g^2(\boldsymbol{\phi},t)\left[1 + k_g \iint u_g^2(\boldsymbol{\phi}',t)\, d\boldsymbol{\phi}'\right]$. $I_1(\phi_1,t)$ and $I_2(\phi_2,t)$ represent the inputs from two band cell modules having orientations θ_1 and θ_2, respectively (to be defined below).

3.1.3 Interactions between band and grid cells

We set the reciprocal connections between band and grid cells to be symmetric, which are written as,

$$W_{gb}(\phi_k^b, \boldsymbol{\phi}^g) = \frac{W_{gb}^0}{\sqrt{2\pi}\sigma_{gb}} \exp\left[-\frac{|\phi_k^b - \phi_k^g|_\pi^2}{2\sigma_{gb}^2}\right], \quad k = 1,2, \tag{7}$$

where $k = 1, 2$ indexes the band cell module.

Thus, the inputs from grid cells to each band cell module are given by,

$$I_g(\phi_k^b, t) = \iint_{-\pi}^{\pi} W_{gb}(\phi_k^b, \boldsymbol{\phi}^{g\prime})\, r_g(\boldsymbol{\phi}^{g\prime}, t)\, d\boldsymbol{\phi}^{g\prime}, \quad k = 1, 2, \tag{8}$$

and the inputs from each band module to grid cells are given by

$$I_k(\phi_k, t) = \int_{-\pi}^{\pi} W_{gb}(\boldsymbol{\phi}^{g}, \phi_k^{b\prime})\, r_b(\phi_k^{b\prime}, t)\, d\phi_k^{b\prime} \quad k = 1, 2. \tag{9}$$

The forward connections from band to grid cells enable the grid cell module to integrate 1D motion trajectories into 2D, while the feedback connections from grid to band cells enable the circuit to amend errors. They jointly support accurate path integration in the 2D space.

3.2 Simulation results

We carry out simulations to test the performance of the neural circuit model (for details, see SI F). We observe that band and grid cells of varying spacings and orientations appear in the hand-designed model (Fig. 5c). The moving trajectory inferred by the model agrees well with ground truth, demonstrating the path integration capacity of the model (Fig. 5d).

It should be noted that the path integration capability demonstrated here builds upon well-established computational frameworks [11, 31], and thus we do not perform additional tests on its robustness in this regard. Furthermore, the absence of accumulated error in this simulation is due to the idealized setting where neither input signals nor the model contain any random noise. In more realistic scenarios where noise is present, small errors would accumulate over time, leading to gradual drift in the estimated position; corresponding results under noisy conditions are provided in SI F.

4 Discussion

In the present study, by using neuroscience prior knowledge and methods, we unfold the black box of RNNs trained to perform a path integration task, and we identify a hierarchical path integration pathway embedded in trained RNNs. Using a combination of analyses (neuron type classification, connection pruning, attractor dynamics inspection et al.), we uncover that along this hierarchical pathway, velocity information of an agent is first conveyed to band cells, band and grid cells then coordinate to carry out path integration, and finally grid cells output the agent location. Furthermore, we reveal that band cells form multiple modules of varying spacing and orientation, and each module constitutes a 1D CANN-like structure. Inspired by the RNN-based study and other neuroscience evidence, we formulate a neural circuit model for path integration. In this model, band cells form multiple functional modules in the form of 1D CANNs, and each of them is responsible for path integration along the module orientation; grid cells form a 2D CANN, which integrates the 1D results of band cell modules to carry out 2D path integration and meanwhile provides feedback to band cells to correct errors. We demonstrate that this computational model works well.

Related neural circuit models. Previous neural circuit models for path integration based only on grid cells require a large amount of direction-conjunctive grid cells, whereby these conjunctive cells receive velocity inputs and shift the activity of the downstream pure grid cells along the same direction to perform location updating [11]. Compared to this model, which requires N^2 pure grid cells and at least $4N^2$ conjunctive grid cells to represent four allocentric directions (north, south, east, and west), the hierarchical model we propose requires only $2N$ pure band cells, $4N$ conjunctive band cells ($2N$ per each module), and N^2 grid cells. Thus, the total number of neurons required to perform path integration is significantly reduced from $5N^2$ to $6N + N^2$, without sacrificing the performance. This computational efficiency offers new insight for the advantage of employing the combination of band and grid cells. To reduce the neuron number, another neural circuit model based only on grid cells considers that the recurrent connection weights between grid cells are velocity-dependent [14]. This requires that neuronal connections are instantly tuned by the agent' velocity and is hence unlikely biologically feasible. Our neural circuit model has the similar structure as that in [15, 16], but has different mathematical formulations.

Model predictions. This body of work also generates predictions testable in experiments, including: 1) band cells should be organized into clusters based on their orientation and spacing in the brain, analogous to the way grid cells are organized along the dorsal-ventral axis in MEC [32]; 2) band cells in the same module should further be organized into attractor networks, similar to head-direction cells in the thalamic nuclei and postsubiculum [33]; 3) disruption of band cells should lead to deficits in path integration in downstream grid cells, potentially offering insights into the mechanisms underlying navigation impairments observed in the early stages of Alzheimer's disease [34]. Investigating these computationally derived predictions using experiments will not only prompt a re-evaluation of the functional role of grid cells in providing a spatial metric for the environment, but also advance our understanding of the computational contribution of band cells in neural circuits.

Limitation and future works. Several important aspects remain to be explored in future studies. For example, it is not fully clear whether undefined cells in the trained RNN hold certain spatial tuning properties, such as those seen in head direction cells [35] or boundary vector cells [36], and whether they play a role in path integration; and if so, at which point along the hierarchical pathway they contribute. Additionally, the nature of interactions between band and grid cells in the trained RNN remains not fully solved, and it needs to be inspected more thoroughly whether these interactions correspond directly to the connectivity profile in our proposed mechanistic neural model. Finally, the current training paradigm is restricted to path integration in physical space. It will be valuable to explore whether this framework is generalizable to path integration in abstract spaces (e.g., social network, value space in decision making) and whether the same hierarchical pathway is preserved [37].

Cross-talk between AI and neuroscience. AI approaches are a powerful tool to train network models for executing tasks, but they also face the challenge that the trained networks are often hard to interpret. Without prior knowledge, one has no cue to open up the black box. Here, we demonstrate that by using neuroscience prior knowledge and methods, we can uncover the detailed hierarchical pathway embedded in the trained RNNs for path integration. On the other hand, the RNN-based findings can advance our understanding of the neural circuit mechanism underlying path integration in the brain. This study presents an example of using cross-field knowledge to facilitate AI interpretation and neuroscience research.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (no. T2421004 to S.W.), the National Key Research and Development Program of China (2024YFF1206500), the Science and Technology Innovation 2030-Brain Science and Brain-inspired Intelligence Project (no. 2021ZD0200204, S.W.), and the Wellcome Principal Research Fellowship (222457/Z/21/Z, N.B.).

References

[1] 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, 2006.

[2] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[3] 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, 2018.

[4] 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.

[5] Johnson Ying, Alexandra T Keinath, Raphael Lavoie, Erika Vigneault, Salah El Mestikawy, and Mark P Brandon. Disruption of the grid cell network in a mouse model of early alzheimer's disease. *Nature Communications*, 13(1):886, 2022.

[6] Emilio Kropff, James E Carmichael, May-Britt Moser, and Edvard I Moser. Speed cells in the medial entorhinal cortex. *Nature*, 523(7561):419–424, 2015.

[7] Francesca Sargolini, Marianne Fyhn, Torkel Hafting, Bruce L McNaughton, Menno P Witter, May-Britt Moser, and Edvard I Moser. Conjunctive representation of position, direction, and velocity in entorhinal cortex. *Science*, 312(5774):758–762, 2006.

[8] Øyvind Arne Høydal, Emilie Ranheim Skytøen, Sebastian Ola Andersson, May-Britt Moser, and Edvard I Moser. Object-vector coding in the medial entorhinal cortex. *Nature*, 568(7752):400–404, 2019.

[9] Julija Krupic, Neil Burgess, and John O'Keefe. Neural representations of location composed of spatially periodic bands. *Science*, 337(6096):853–857, 2012.

[10] Neil Burgess, Caswell Barry, and John O'keefe. An oscillatory interference model of grid cell firing. *Hippocampus*, 17(9):801–812, 2007.

[11] Yoram Burak and Ila R Fiete. Accurate path integration in continuous attractor network models of grid cells. *PLoS computational biology*, 5(2):e1000291, 2009.

[12] Mark C Fuhs and David S Touretzky. A spin glass model of path integration in rat medial entorhinal cortex. *Journal of Neuroscience*, 26(16):4266–4276, 2006.

[13] Alexis Guanella, Daniel Kiper, and Paul Verschure. A model of grid cells based on a twisted torus topology. *International journal of neural systems*, 17(04):231–240, 2007.

[14] Rylan Schaeffer, Mikail Khona, Tzuhsuan Ma, Cristobal Eyzaguirre, Sanmi Koyejo, and Ila Fiete. Self-supervised learning of representations for space generates multi-modular grid cells. *Advances in Neural Information Processing Systems*, 36:23140–23157, 2023.

[15] Christopher P Burgess and Neil Burgess. Controlling phase noise in oscillatory interference models of grid cell firing. *Journal of Neuroscience*, 34(18):6224–6232, 2014.

[16] Daniel Bush and Neil Burgess. A hybrid oscillatory interference/continuous attractor network model of grid cell firing. *Journal of Neuroscience*, 34(14):5065–5079, 2014.

[17] Andrea Banino, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, Martin J Chadwick, Thomas Degris, Joseph Modayil, et al. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705):429–433, 2018.

[18] Christopher J Cueva and Xue-Xin Wei. Emergence of grid-like representations by training recurrent neural networks to perform spatial localization. *arXiv preprint arXiv:1803.07770*, 2018.

[19] 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. *arXiv preprint arXiv:2210.02684*, 2022.

[20] William Dorrell, Peter E Latham, Timothy EJ Behrens, and James CR Whittington. Actionable neural representations: Grid cells from minimal constraints. *arXiv preprint arXiv:2209.15563*, 2022.

[21] Yedidyah Dordek, Daniel Soudry, Ron Meir, and Dori Derdikman. Extracting grid cell characteristics from place cell inputs using non-negative principal component analysis. *Elife*, 5:e10094, 2016.

[22] 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*, 111(1):121–137, 2023.

[23] 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), 2023.

[24] Markus Pettersen, Vemund Sigmundson Schøyen, Mattis Dalsætra Østby, Anders Malthe-Sørenssen, and Mikkel Elle Lepperød. Self-supervised grid cells without path integration. *bioRxiv*, pages 2024–05, 2024.

[25] William Redman, Francisco Acosta, Santiago Acosta-Mendoza, and Nina Miolane. Not so griddy: Internal representations of rnns path integrating more than one agent. *Advances in Neural Information Processing Systems*, 37:22657–22689, 2024.

[26] Changmin Yu, Timothy EJ Behrens, and Neil Burgess. Prediction and generalisation over directed actions by grid cells. *arXiv preprint arXiv:2006.03355*, 2020.

[27] Tianhao Chu, Zilong Ji, Junfeng Zuo, Yuanyuan Mi, Wen-hao Zhang, Tiejun Huang, Daniel Bush, Neil Burgess, and Si Wu. Firing rate adaptation affords place cell theta sweeps, phase precession, and procession. *Elife*, 12:RP87055, 2024.

[28] Zilong Ji, Tianhao Chu, Si Wu, and Neil Burgess. A systems model of alternating theta sweeps via firing rate adaptation. *Current Biology*, 35(4):709–722, 2025.

[29] Zilong Ji, Tianhao Chu, Xingsi Dong, Changmin Yu, Daniel Bush, Neil Burgess, and Si Wu. Dynamical modulation of hippocampal replay sequences through firing rate adaptation. *BioRxiv*, pages 2024–09, 2024.

[30] Si Wu, Kosuke Hamaguchi, and Shun-ichi Amari. Dynamics and computation of continuous attractors. *Neural computation*, 20(4):994–1025, 2008.

[31] Wenhao Zhang, Ying Nian Wu, and Si Wu. Translation-equivariant representation in recurrent networks with a continuous manifold of attractors. *Advances in Neural Information Processing Systems*, 35:15770–15783, 2022.

[32] Hanne Stensola, Tor Stensola, Trygve Solstad, Kristian Frøland, May-Britt Moser, and Edvard I Moser. The entorhinal grid map is discretized. *Nature*, 492(7427):72–78, 2012.

[33] Jeffrey S Taube. The head direction signal: origins and sensory-motor integration. *Annu. Rev. Neurosci.*, 30(1):181–207, 2007.

[34] Coco Newton, Marianna Pope, Catarina Rua, Richard Henson, Zilong Ji, Neil Burgess, Christopher T Rodgers, Matthias Stangl, Maria-Eleni Dounavi, Andrea Castegnaro, et al. Entorhinal-based path integration selectively predicts midlife risk of alzheimer's disease. *Alzheimer's & Dementia*, 20(4):2779–2793, 2024.

[35] Jeffrey S Taube, Robert U Muller, and James B Ranck. Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis. *Journal of Neuroscience*, 10(2):420–435, 1990.

[36] Colin Lever, Stephen Burton, Ali Jeewajee, John O'Keefe, and Neil Burgess. Boundary vector cells in the subiculum of the hippocampal formation. *Journal of Neuroscience*, 29(31):9771–9777, 2009.

[37] James CR Whittington, Timothy H Muller, Shirley Mark, Guifen Chen, Caswell Barry, Neil Burgess, and Timothy EJ Behrens. The tolman-eichenbaum machine: unifying space and relational memory through generalization in the hippocampal formation. *Cell*, 183(5):1249–1263, 2020.

A Training details of RNN

To train RNNs on path integration tasks, we implemente an established computational framework adapted from the open-source repository developed by Ganguli Lab (`https://github.com/ganguli-lab/grid-pattern-formation`). Code used for this paper is publicly available at `https://github.com/yuling-wu/band_grid_hierarchy`. Vanilla RNN architectures are trained using the hyperparameters specified in Table S1. Our weight matrices are initialized randomly using PyTorch's default scheme and there are no hand-crafted or localized initialization tricks. The trajectory used to train each iteration or test is sampled randomly from a distribution. The training loss is presented in Fig. S1.

Table S1: **Parameters used to train Vanilla RNNs**

Parameter	Value
Arena size	(2.2m x 2.2m)
Average agent speed	0.1 m/sec
Place cells (N_p)	512
σ_1	0.12 m
σ_2	0.24 m
Recurrent units (N_R)	4096
Path length	20
Epochs	100
Batch size	200
Batches per epoch	1000
Learning rate	10^{-4}
l2 regularization	10^{-4}
Optimizer	RMSProp

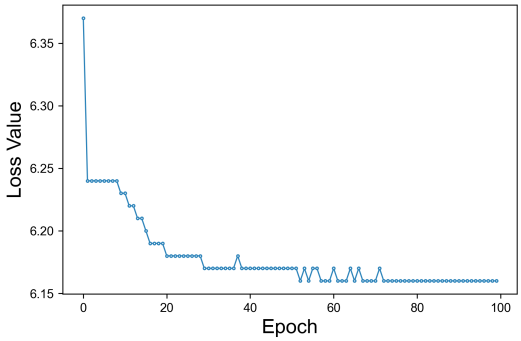

Figure S1: Training loss for the vanilla RNN.

Place cell activity: The receptive field centers of place cells are randomly distributed across the square environment. Each place cell's firing activity is modeled using a difference-of-softmax tuning curve, analogous to the difference-of-Gaussians function:

$$p_i(t) = \mathrm{DoG}[\mathbf{x}(t), \mathbf{c}_i] = \frac{e^{-\|\mathbf{x}-\mathbf{c}_i\|^2/2\sigma_1^2}}{\sum_{j=1}^{N_p} e^{-\|\mathbf{x}-\mathbf{c}_j\|^2/2\sigma_1^2}} - \frac{e^{-\|\mathbf{x}-\mathbf{c}_i\|^2/2\sigma_2^2}}{\sum_{j=1}^{N_p} e^{-\|\mathbf{x}-\mathbf{c}_j\|^2/2\sigma_2^2}} \tag{10}$$

where $\mathbf{x}$ is the current location of the agent, $\mathbf{c}_i$ denotes the center of the i_{th} place cell's receptive field, and σ_1 and σ_2 parameterize the width of the center and surround, respectively. The network received velocity inputs derived from the simulated trajectory and was trained to produce the simulated place cell activities as output.

Path integration error: Locations of the agent are decoded from the place cell activities by considering three maximally active place cells and averaging the x and y coordinates of their place

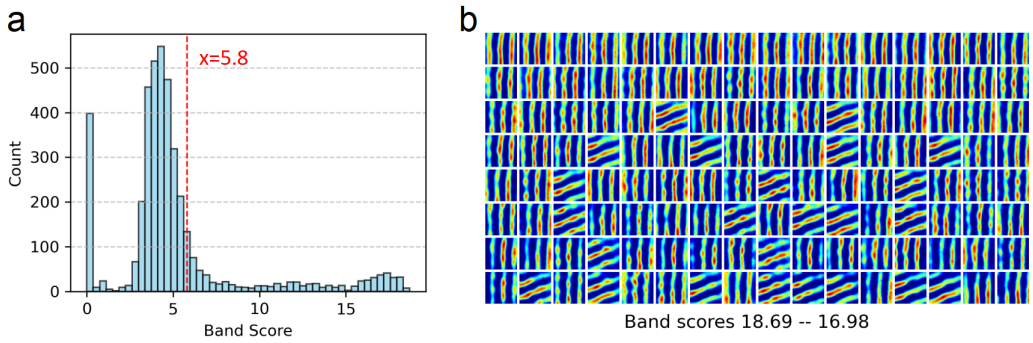

Figure S2: (a) Band score distribution. (b) The top 128 band cells with the highest scores.

field centers. We quantified decoding accuracy using the mean Euclidean distance between the decoded positions and ground truth locations:

$$\mathcal{E} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \tag{11}$$

where (x_i, y_i) denotes the true position and $(\hat{x}_i, \hat{y}_i)$ represents the decoded position for the i-th sample.

B Using neuroscience knowledge to analyse RNN

B.1 Rate map construction

Rate mapping—a fundamental technique in systems neuroscience—quantifies how neuronal firing rates vary with spatial location, providing crucial insights into neural representations of space. When applied to RNNs trained on path integration, this approach has revealed the spontaneous emergence of functionally specialized units analogous to biological grid cells, and band cells observed in biological navigation systems. To do this, we generate a test batch ($n = 200$ trajectories) as inputs to the RNN and record the activation states of all recurrent layer units, the agent's current position and head direction at each step.

To construct spatial rate maps, we partition the environment into an $m \times m$ grid, resulting in m^2 spatial bins, and compute the mean activation for each unit within each bin. Here we set $m = 20$ to compute a set of low resolution maps to use for evaluating grid score and set $m = 50$ to compute a set of high resolution maps to plot the resulting tuning curves and visualize the spatial firing patterns. To construct angular activation profiles, we partition the head direction ranging from $-\pi$ to π into 100 bins, and compute the mean activation for each unit within each bin.

B.2 Neuron type classification

Band cell: Ideal band cells exhibit a defining electrophysiological signature characterized by spatially periodic firing fields that form bands across the environment [9]. To quantify this, we developed a spectral analysis method. We first compute the 2D Fourier transform of the spatial rate map, suppressing negative frequencies. The resulting power spectrum is fitted with a 2D Gaussian model parameterized by amplitude (A), spatial frequency (k), orientation angle (θ), and bandwidth (σ) using constrained nonlinear optimization. The band score is calculated as the normalized correlation between the actual spectral power distribution and the Gaussian model, weighted by σ:

$$\text{Band score} = \frac{\langle \mathbf{F}, \mathbf{G} \rangle}{\|\mathbf{F}\| \cdot \|\mathbf{G}\|} \cdot \frac{1}{\sigma} \tag{12}$$

where:

- $\mathbf{F}$ is the spectral power matrix obtained from the 2D Fourier transform of the rate map:

$$F(u, v) = |\mathcal{F}\{R(x, y)\}|_{\text{shifted}} \tag{13}$$

with negative frequencies suppressed ($F(u, v) = 0$ for $u < \frac{N}{2}$)

- $\mathbf{G}$ is the fitted 2D Gaussian model:

$$G(x, y) = A \exp\left(-\frac{(x - k\cos\phi)^2 + (y - k\sin\phi)^2}{2\sigma^2}\right) \tag{14}$$

- Parameters are optimized via:

$$(A^*, k^*, \phi^*, \sigma^*) = \underset{A,k,\phi,\sigma}{\text{argmin}} \|\mathbf{F} - \mathbf{G}\|^2 \tag{15}$$

  subject to $k \in [0.2, 1]$, $\phi \in [0, \pi]$, $\sigma \in [0.05, 0.5]$

- The spatial frequency components are derived as:

$$k_x = \frac{k^* \cos\phi^*}{2\Delta x}, \quad k_y = \frac{k^* \sin\phi^*}{2\Delta x} \tag{16}$$

  where Δx is the spatial bin size.

Band score distribution is show in Fig. S2a and cells with a band score of over 5.8 are classified as band cells. Example band cells are shown in Fig. S2b.

Grid cell: Ideal grid cells exhibit a defining electrophysiological signature characterized by spatially periodic firing fields that form a hexagonal lattice tessellation across the environment [4]. Grid score was quantified through a rotational analysis of the autocorrelogram derived from the spatial rate map. Specifically, the autocorrelogram underwent circular rotations in 30° increments, followed by computation of Pearson correlations between each rotated version and the original map. The final grid score metric was calculated as:

$$\text{Grid Score} = \frac{\rho(60^\circ) + \rho(120^\circ)}{2} - \frac{\rho(30^\circ) + \rho(90^\circ) + \rho(150^\circ)}{3} \tag{17}$$

where $\rho(\theta)$ represents the correlation coefficient after θ-degree rotation. This measure captures the hexagonal periodicity signature by comparing correlations at the theoretical peaks (60° and 120°) versus troughs (30°, 90°, and 150°) of an ideal grid pattern. Finally, cells with a grid score of over 0.88 are classified as grid cells.

Undefined cell: Any cells that are neither band nor grid cells are classified as undefined cells.

B.3 Properties of band cells tuning

Spatial tuning: To quantitatively characterize band cell activity patterns, we derive three key spatial parameters from the spatial rate maps:

- **Spacing** λ, the distance between adjacent bands, calculated by $\lambda = 2\pi/\sqrt{k_x^2 + k_y^2}$, where k_x and k_y are the same as in Eq.16;
- **Orientation** θ, the orientation of bands in the 2D space, computed through the following spectral analysis procedure:

$$\theta = \left(\pi - \left[\frac{1}{2}\text{angle}\left(\sum_{\omega} P(\omega) e^{i2\theta(\omega)}\right) + \pi\right] - \frac{\pi}{2}\right) \mod \pi \tag{18}$$

  where $\theta(\omega) = \tan^{-1}(f_y/f_x)$ is the orientation of frequency component ω (constrained to $[0, \pi]$), $P(\omega)$ is the spectral power at frequency ω, the $\frac{1}{2}\text{angle}(\cdot)$ handles the π-periodicity of band patterns, the $\pi - (\cdot) - \pi/2$ sequence performs the requested axis transformation, and the modulo π operation ensures angular continuity;
- **Phase** ϕ, the positional separation between band cells having the same spacing and orientation. Denote $\Delta \in (-\lambda/2, \lambda/2)$ the offset between the adjacent bands of two band cells, their phase difference is given by $\phi = (\Delta/\lambda) \times \pi \in (-\pi, \pi)$.

  As illustrated in Fig. S3a, the connection weight of band cells varies with their phase difference, reminiscent to the center-surround connection pattern.

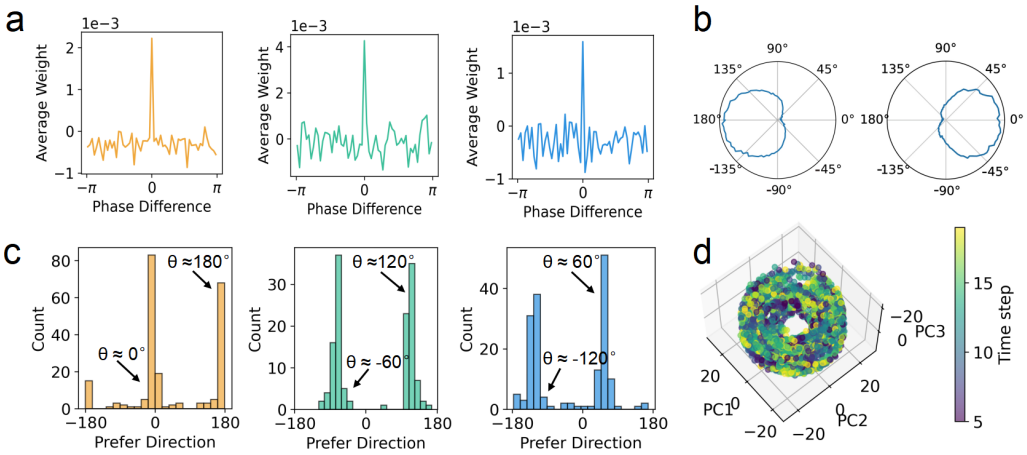

Figure S3: (a) The averaged connection profile of band cells as a function of their phase difference. In figures a and c, colors indicate distinct band cell clusters, and each cluster has the same spacing and orientation. (b) Example band cells with opposite preferred direction. (c) Histograms of preferred directions of band cells in three clusters, which have a bimodal shape peaked at either the cluster orientation θ or its opposite direction $\theta + \pi$. (d) 3D isomap embedding population activity from a band cell cluster.

Directional tuning: To quantitatively characterize band cell direction, we calculate two parameters from the angular activation profiles:

- The **direction score** quantifies directional tuning by fitting angular activation patterns $\mathbf{y}_i(\psi)$ with a circular Gaussian function:

$$\mathbf{y}_{\text{model}}(\psi) = A \exp\left(-\frac{d(\psi,\mu)^2}{2\sigma^2}\right) \tag{19}$$

  where $d(\psi,\mu) = \min(|\psi-\mu|, 2\pi - |\psi-\mu|)$ is the circular distance between angle ϕ and preferred direction μ, and σ determines tuning width. Parameters (A,μ,σ) are estimated via nonlinear least-squares fitting initialized at $[1,0,1]$. The direction score is computed as the normalized projection between observed ($\mathbf{y}_{\text{data}}$) and fitted ($\mathbf{y}_{\text{model}}$) activation patterns:

$$\text{Direction Score} = \frac{\mathbf{y}_{\text{data}} \cdot \mathbf{y}_{\text{model}}}{\|\mathbf{y}_{\text{data}}\| \|\mathbf{y}_{\text{model}}\|} \tag{20}$$

- The **preferred direction** $\xi_i \in [-\pi,\pi)$ for each band cell i is derived from its angular activation profile $y_i(\psi_k)$ through circular moment analysis, computed as:

$$\xi_i = \arg\Big(\sum_{k=1}^{N_\psi} y_i(\psi_k) e^{i\psi_k}\Big) \tag{21}$$

  where $\psi_k = -\pi + 2\pi k/N_\psi$ ($k = 0, ..., N_\psi - 1$) defines N_ψ equally spaced angular bins, $e^{i\psi_k}$ represents the unit vector in complex space, and $\arg(\cdot)$ extracts the phase of the resultant vector.

  We find that, for each cluster with the same spacing and direction tuning, the majority of band cells form two subgroups with opposite preferred directions (Fig. S3b-c).

B.4 Population activity analysis

Neural population dynamics were analyzed using the following pipeline:

- **Data extraction**: A test batch ($n = 200$ trajectories) was generated as inputs to the RNN to generate the network's corresponding latent representations. Then neuron subset $\mathbf{X}_{\text{raw}}$ was extracted (shape: $T \times N$, where T is timepoints and N is neurons to be analyzed).

- **Preprocessing**: Neural activity was standardized using z-score normalization:

$$\mathbf{X}_{\text{scaled}} = \text{scaler}(\mathbf{X}_{\text{raw}}) \tag{22}$$

- **Dimensionality reduction**:
  - PCA projected data to 15 principal components
  - Isomap further reduced to 3D ($k = 8$ neighbors, geodesic distance metric)

$$\mathbf{X}_{\text{lowdim}} = \text{Isomap}(\text{PCA}(\mathbf{X}_{\text{scaled}})) \tag{23}$$

- **Visualization**: The 3D neural manifold was plotted against animal trajectory (5-frame offset corrected) using time-indexed coloring (viridis colormap).

  Fig. S3d displays an alternative Isomap embedding for a distinct band cell group not analyzed in the main text.

C Dissecting information flows in RNN

C.1 Pruning experiments

To systematically investigate the computational roles of recurrent layer units in path integration, we performed four distinct pruning experiments with rigorous controls:

- **Pruning velocity inputs**: For each pruning level $n \in \{1, \ldots, 700\}$, we randomly selecte n units across cell types and nullify their velocity input weights:

$$\mathbf{M}^{(n)}_{i,\cdot} \leftarrow \mathbf{0}, \quad i \in \mathcal{S}_n$$

  where $\mathcal{S}_n$ denotes a size-n random subset of units, and $\mathbf{M} \in \mathbb{R}^{N_R \times 2}$ encodes velocity-to-recurrent layer weights.

- **Pruning read-out connections**: We prune output projections from n randomly selected units by zeroing their readout weights:

$$\mathbf{W}^{(n)}_{\cdot,j} \leftarrow \mathbf{0}, \quad j \in \mathcal{S}_n$$

  with $\mathbf{W} \in \mathbb{R}^{N_p \times N_R}$ representing the recurrent-to-output weight matrix.

- **Pruning cell types**: For comprehensive cell inactivation, we simultaneously nullified all incoming and outgoing connections of selected units:

$$\mathbf{J}_{i,\cdot} \leftarrow \mathbf{0}, \quad \mathbf{J}_{\cdot,i} \leftarrow \mathbf{0}, \quad \forall i \in \mathcal{S}_n$$

  where $\mathbf{J} \in \mathbb{R}^{N_R \times N_R}$ is the recurrent weight matrix.

- **Pruning connections between band and grid cells**:

  For the cell-type specific pruning analysis, we conducted bidirectional connection pruning between band (A) and grid (B) cell populations:

  - **A→B Pruning**: Nullified all forward connections from a random subset of n band cells to *every* grid cell:

$$\mathbf{J}^{(n)}_{A_k,B_j} \leftarrow 0, \quad \forall k \in \mathcal{S}_n, \forall j \in \{1, ..., N_B\}$$

  - **B→A Pruning**: Symmetrically ablated all feedback connections from a random subset of n grid cells to *every* band cell:

$$\mathbf{J}^{(n)}_{B_k,A_j} \leftarrow 0, \quad \forall k \in \mathcal{S}_n, \forall j \in \{1, ..., N_A\}$$

Error bars represent $\pm$ standard deviation across $n = 30$ independent repetitions, reflecting the variability in individual measurements.

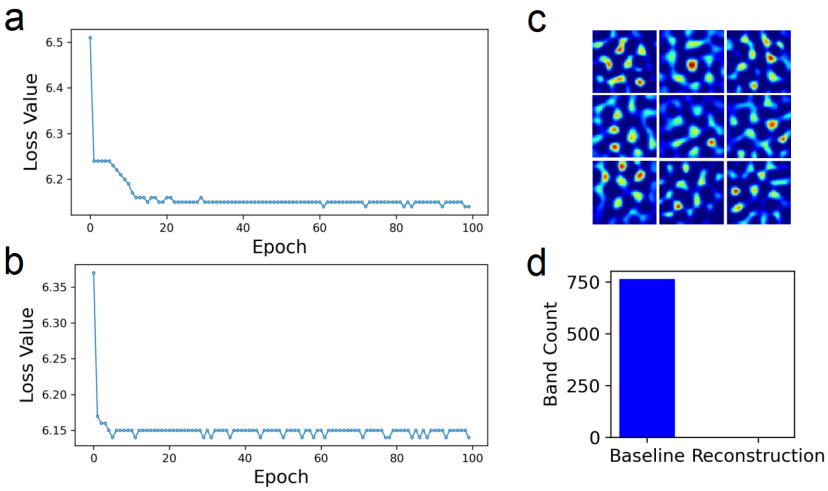

Figure S4: (a) Training loss for the RNN with predefined hierarchical structure. (b) Training loss for RNN for location reconstruction. (c) Example grid cells found in the newly trained RNN for location reconstruction. (d) The number of band cells found in the Vanilla RNN (Baseline) and newly trained RNN (Reconstruction).

C.2 RNN with predefined hierarchical structure

We implement a hierarchical RNN architecture that takes the hierarchical pathway as a prior (Fig. 4b). The model consists of two reciprocally connected modules:

- **First module (RNN1)**: Processes velocity inputs ($\mathbf{v}_t$) combined with feedback from the second module, designed to emulate the properties of band cells:

$$\mathbf{r}_{t+1}^{(1)} = \sigma\left(\mathbf{J}_1\mathbf{r}_t^{(1)} + \mathbf{J}_2\mathbf{r}_t^{(2)} + \mathbf{M}\mathbf{v}_t\right) \tag{24}$$

  where $\mathbf{r}_t^{(2)}$ represents the feedback from RNN2, $\mathbf{J}_1$ and $\mathbf{J}_2$ represent the recurrent weights of RNN1 and RNN2 respectively.

- **Second module (RNN2)**: Receives inputs from RNN1 and generates spatial representations, mimicking grid cell functionality:

$$\mathbf{r}_{t+1}^{(2)} = \sigma(\mathbf{J}_1\mathbf{r}_{t+1}^{(1)} + \mathbf{J}_2\mathbf{r}_t^{(2)}) \tag{25}$$

We train this hierarchical RNN model on the standard path integration task using the same hyperparameters specified in Table S1. The training loss is presented in Fig. S4 a.

C.3 RNN trained for location reconstruction

We modify the task from path integration to pure location reconstruction by replacing velocity inputs of the vanilla RNN with place cell activations at each timestep, expressed as:

$$\mathbf{r}_{t+1} = \sigma\left(\mathbf{J}\mathbf{r}_t + \mathbf{M}\mathbf{p}_t\right) \tag{26}$$

where $\mathbf{p}_t \in \mathbb{R}^{N_g}$ denotes the ground truth place cell activation pattern corresponding to the current position. All hyperparameters are the same as in Table S1. The training loss is presented in Fig. S4 b. After training the RNN on this modified task, we observe the emergence of grid-like activity patterns while notably lacking band cell representations (Fig. S4 c-d).

D Quantitative Performance Analysis

D.1 Error Distribution Analysis

We quantified path integration performance across 30 independent trials for each of 10 different RNN implementations (Seeds 0–9). The results demonstrate consistent performance, with errors (normalized by arena size) reported as mean $\pm$ standard deviation (units: %):

$$\begin{array}{ll} \text{Seed } 0 : 2.09 \pm 0.06 & \text{Seed } 5 : 2.07 \pm 0.04 \\ \text{Seed } 1 : 2.09 \pm 0.05 & \text{Seed } 6 : 2.08 \pm 0.05 \\ \text{Seed } 2 : 2.08 \pm 0.05 & \text{Seed } 7 : 2.09 \pm 0.06 \\ \text{Seed } 3 : 2.07 \pm 0.05 & \text{Seed } 8 : 2.13 \pm 0.06 \\ \text{Seed } 4 : 2.12 \pm 0.07 & \text{Seed } 9 : 2.11 \pm 0.07 \end{array}$$

D.2 Computational Complexity Scaling

We analyzed the relationship between hidden layer size (N_R) and path integration error, with errors (normalized by arena size) reported as mean $\pm$ standard deviation (units: %):

$$\begin{aligned} N_R = 1024 &: 2.57 \pm 0.14 \\ N_R = 2048 &: 2.30 \pm 0.08 \\ N_R = 4096 &: 2.09 \pm 0.06 \\ N_R = 8192 &: 2.06 \pm 0.05 \end{aligned}$$

This analysis demonstrates that our model achieves stable performance ($< 2.19\%$ error) with $N \geq 4096$ neurons, suggesting our approach provides computationally efficient spatial representation.

D.3 Pruning Baseline

For each pruning condition, we prune connections selected from random groups as baseline to assess the relative importance of our specific pruning choices. The key results are presented in Table S2, which demonstrates that pruning specific functional connections (input-to-band or grid-read-out) produces significantly larger errors compared to random pruning, suggesting specialized functional roles for these specific cell groups.

Table S2: Path integration error increase (mean $\pm$ SD across 30 trials when 750 connections are pruned). Error is measured by the standard deviation of location prediction errors divided by the arena size (unit: %).

Pruning Type	Band cell	Grid cell	Undefined cell	Random cell
Pruning velocity inputs	2.32 ± 0.43	0.11 ± 0.07	0.14 ± 0.09	0.46 ± 0.07
Pruning read-out connections	2.67 ± 0.50	9.15 ± 0.71	0.30 ± 0.11	0.81 ± 0.22

E Reproducibility

E.1 Experiments Across Different Seeds

we repeat the experiments across 10 random seeds for Vanilla RNN. Our results robustly demonstrate that band cells consistently serve as the primary recipients of velocity inputs and grid cells reliably encode spatial location information. The additional results are:

Neuron Counts and Pruned Connections:

Seed	Band Cells	Grid Cells	Undefined Cells	Pruned Connections
0	868	443	2785	443
1	818	526	2752	526
2	914	267	2915	267
3	823	146	3127	146
4	782	457	2857	457
5	854	442	2800	442
6	972	228	2896	228
7	535	324	3237	324
8	553	597	2946	553
9	810	394	2892	394

Pruning Performance (mean $\pm$ SD):

Velocity Input Pruning:

Seed	Band	Grid	Undefined
0	1.8282 ± 0.1026	0.0003 ± 0.1069	0.0106 ± 0.0990
1	2.5350 ± 0.1058	0.0401 ± 0.0616	0.0475 ± 0.0720
2	0.7015 ± 0.0786	0.0247 ± 0.0687	0.0248 ± 0.0736
3	0.3086 ± 0.0679	0.0022 ± 0.0744	0.0254 ± 0.0701
4	2.2449 ± 0.1073	0.0430 ± 0.0955	0.0173 ± 0.0710
5	2.0076 ± 0.0884	0.0217 ± 0.0480	0.0172 ± 0.0733
6	0.6119 ± 0.0963	0.0123 ± 0.0683	0.0003 ± 0.0732
7	1.2673 ± 0.0806	0.0242 ± 0.0697	0.0603 ± 0.0803
8	2.7330 ± 0.1191	0.0411 ± 0.0625	0.2117 ± 0.0608
9	1.3750 ± 0.1083	0.0124 ± 0.0815	0.0225 ± 0.0795

Read-out Connection Pruning:

Seed	Band	Grid	Undefined
0	0.9446 ± 0.2144	4.2009 ± 0.6104	0.1558 ± 0.1297
1	1.7184 ± 0.2905	7.4244 ± 0.7040	0.4433 ± 0.1778
2	0.4024 ± 0.1543	0.7765 ± 0.1850	0.1569 ± 0.1119
3	0.1272 ± 0.0737	0.2362 ± 0.1024	0.1058 ± 0.0894
4	0.5466 ± 0.1479	1.6829 ± 0.2364	0.2440 ± 0.1109
5	0.7177 ± 0.2362	2.3296 ± 0.3476	0.1877 ± 0.1093
6	0.2365 ± 0.1038	0.7850 ± 0.2799	0.0657 ± 0.0990
7	0.3730 ± 0.1556	1.0181 ± 0.2656	0.0985 ± 0.0967
8	2.6436 ± 0.3403	8.3318 ± 0.6056	0.2729 ± 0.1016
9	0.6141 ± 0.1355	2.8032 ± 0.5064	0.1566 ± 0.1104

E.2 Different Training Paragram

To check the flexibility of our results, we trained three additional path-integration RNN variants:

- **Vanilla RNN $\rightarrow$ LSTM transition** (different in net structure).
- **Xu et al. (2022) [19] training paradigm** (different in net structure (RNN and LSTM), place field (Gaussian), and loss function (conformal isometry and path integration loss)).
- **Petterson et al. (2024) [24] training paradigm** (different in initial state (MLP), loss function (distance preservation and capacity loss), and decoder (none)).

The quantitative analysis of path integration performance across the three RNN variants are presented below, which demonstrates that velocity inputs are preferentially processed by band cells across all three RNN variants, as evidenced by their greater sensitivity to velocity pruning. However, the absence of decoder modules in some variants prevented systematic pruning read-out experiments, limiting direct comparisons of read-out performance.

E.2.1 Vanilla RNN→LSTM transition

The pruning experiments show that models degrade severely when velocity inputs to band cells are pruned, but remain relatively stable when inputs to grid cells or undefined cells are pruned under the same conditions:

Neuron Counts and Pruned Connections:

Seed	Band Cells	Grid Cells	Undefined Cells	Pruned Connections
0	346	534	3216	346
1	258	276	3562	258
2	524	554	3018	524
3	394	305	3397	305
4	500	186	3410	186
5	313	136	3647	136
6	294	57	3745	57
7	324	225	3547	225
8	258	298	3540	258
9	363	124	3609	124

Velocity Input Pruning Performance (mean $\pm$ SD):

Seed	Band Cells	Grid Cells	Undefined Cells
0	1.3948 ± 0.1030	0.0070 ± 0.0667	0.0809 ± 0.0785
1	1.3914 ± 0.1392	0.0261 ± 0.0729	0.0708 ± 0.0718
2	1.8552 ± 0.1261	0.0638 ± 0.0974	0.1554 ± 0.1085
3	1.1153 ± 0.1696	0.0715 ± 0.1594	0.0511 ± 0.1202
4	0.2639 ± 0.0781	0.0128 ± 0.0605	0.0268 ± 0.0613
5	0.3274 ± 0.0774	0.0150 ± 0.0647	0.0046 ± 0.0776
6	0.0811 ± 0.0649	0.0067 ± 0.0548	0.0036 ± 0.0597
7	0.8140 ± 0.1018	0.0022 ± 0.0689	0.0610 ± 0.0805
8	0.7389 ± 0.0691	0.0303 ± 0.0568	0.0905 ± 0.0544
9	0.2011 ± 0.1214	0.0223 ± 0.1245	0.0191 ± 0.1048

E.2.2 Xu et al. (2022)

The pruning experiments reveal that band cells exhibit significantly greater sensitivity to input manipulation compared to grid or undefined cells, which is consistently observed across all experimental seeds except seed 1:

Neuron Counts and Pruned Connections:

Seed	Band Cells	Grid Cells	Undefined Cells	Pruned Connections
0	347	403	1050	347
1	313	424	1063	313
2	303	363	1134	303
3	316	369	1115	316
4	312	384	1104	312
5	311	382	1107	311
6	318	427	1055	318
7	302	452	1046	302
8	318	418	1064	318
9	312	401	1087	312

Velocity Input Pruning Performance (mean ± SD):

Seed	Band Cells	Grid Cells	Undefined Cells
0	8.3573 ± 1.3764	0.3688 ± 0.2071	2.9723 ± 1.1574
1	9.6032 ± 1.7588	0.0395 ± 0.2031	10.0732 ± 1.8683
2	13.5328 ± 2.1336	0.0767 ± 0.1197	4.0376 ± 0.6832
3	14.8651 ± 1.8007	0.8835 ± 0.2180	4.9369 ± 0.8729
4	15.2718 ± 2.3267	-0.0064 ± 0.0720	2.6171 ± 0.4573
5	10.3343 ± 2.0101	0.3419 ± 0.3565	4.1531 ± 0.7071
6	9.5627 ± 1.3337	-0.0611 ± 0.1900	5.5746 ± 0.9934
7	9.3729 ± 1.6383	0.2143 ± 0.3917	5.8331 ± 1.1967
8	11.2179 ± 1.9553	3.3365 ± 1.3821	4.1106 ± 0.7282
9	5.1514 ± 1.3389	-0.6553 ± 0.8521	3.9936 ± 1.6025

E.2.3 Petterson et al. (2024)

Our 10-seed replication of Petterson et al. (2024) demonstrates robust preferential processing of velocity inputs by band cells. The additional results are:

Neuron Counts and Pruned Connections:

Seed	Band	Grid	Undefined	Pruned Connections
0	37	196	23	23
1	26	200	30	26
2	26	209	21	21
3	38	194	24	24
4	26	194	36	26
5	38	203	15	15
6	24	202	30	24
7	35	201	20	20
8	34	205	17	17
9	36	202	18	18

Pruning Performance (mean ± SD):

Seed	Band	Grid	Undefined
0	2.5472 ± 0.3644	0.0368 ± 0.0278	0.6565 ± 0.0061
1	5.2265 ± 0.0205	0.0198 ± 0.0121	0.7424 ± 0.1176
2	3.4828 ± 0.2735	0.0297 ± 0.0187	1.2256 ± 0.0086
3	3.1603 ± 0.3524	0.0147 ± 0.0075	0.6654 ± 0.0072
4	3.3173 ± 0.0183	0.0315 ± 0.0250	1.5762 ± 0.3514
5	1.8896 ± 0.2988	0.0122 ± 0.0071	0.1567 ± 0.0012
6	3.6600 ± 0.0223	0.0223 ± 0.0114	1.5868 ± 0.2246
7	2.8943 ± 0.3955	0.0117 ± 0.0074	0.4566 ± 0.0030
8	2.5261 ± 0.3254	0.0147 ± 0.0099	0.3661 ± 0.0035
9	2.3297 ± 0.3849	0.0205 ± 0.0176	0.4389 ± 0.0034

F Neural circuit model details

F.1 Overview of the model architecture

We provide a mathematical overview of the neural circuit model for 2D path integration, consisting of band cell modules and a grid cell module. Each component performs specific computations described below.

1. Pure band cell dynamics (1D CANN)

Each band cell module maintains a bump of activity along a 1D phase space $\phi \in (-\pi, \pi]$, via recurrent dynamics:

$$\tau \frac{\partial u_b(\phi,t)}{\partial t} = -u_b(\phi,t) + \rho \int J_b(\phi,\phi')\, r_b(\phi',t)\, d\phi' + \rho \sum_{m=\pm} \int W_m(\phi,\phi')\, r_m(\phi',t)\, d\phi' + I_g(\phi,t), \tag{27}$$

where u_b and r_b are synaptic input and firing rate of pure band cells. The recurrent connectivity is symmetric and translation-invariant:

$$J_b(\phi, \phi') = \frac{J_b^0}{\sqrt{2\pi}\sigma_b} \exp\left[-\frac{|\phi - \phi'|_\pi^2}{2\sigma_b^2}\right]. \tag{28}$$

Their firing rates are normalized to stabilize the bump:

$$r_b(\phi, t) = \frac{u_b^2(\phi, t)}{1 + k_b \rho \int u_b^2(\phi', t)\, d\phi'}. \tag{29}$$

2. Direction-conjunctive band cells

These cells receive excitatory input from pure band cells and encode the agent's projected velocity along the module's preferred orientation θ and its opposite direction $\theta + \pi$. Their dynamics are defined as:

$$\tau \frac{\partial u_\pm(\phi, t)}{\partial t} = -u_\pm(\phi, t) + w_b \cdot r_b(\phi, t), \tag{30}$$

$$r_\pm(\phi, t) = \left[g_0 + (\mathbf{v}(t) \cdot \mathbf{k}) u_\pm(\phi, t)\right]_+, \tag{31}$$

where $u_\pm(\phi, t)$ and $r_\pm(\phi, t)$ are the synaptic input and firing rate of the conjunctive cells. The velocity projection vector is $\mathbf{k} = (\sin\theta/\lambda, \cos\theta/\lambda)$, determining the speed component along the module's axis. The firing rates reflect the movement direction and strength of the agent.

Feedback to pure band cells. Direction-conjunctive cells modulate the activity bump in pure band cells via asymmetric projections. The connection weights from conjunctive to pure band cells are:

$$W_\pm(\phi, \phi') = \frac{W_b^0}{\sqrt{2\pi}\sigma_b} \exp\left[-\frac{|\phi - \phi' \mp \delta|_\pi^2}{2\sigma_b^2}\right]. \tag{32}$$

These connections are offset by $\pm\delta$ along the phase axis and thus induce a translation of the activity bump, driving 1D path integration along the module orientation. Specifically, W_+ promotes bump motion in the forward direction (θ), and W_- in the opposite direction ($\theta + \pi$).

3. Grid cells

Grid cells are organized on a 2D toroidal manifold and integrate inputs from two band cell modules with orientations θ_1 and θ_2, which are $60°$ apart. They form a 2D continuous attractor neural network (CANN) with the following dynamics:

$$\tau_g \frac{\partial u_g(\boldsymbol{\phi}, t)}{\partial t} = -u_g(\boldsymbol{\phi}, t) + \iint_{-\pi}^{\pi} J_g(\boldsymbol{\phi}, \boldsymbol{\phi}')\, r_g(\boldsymbol{\phi}', t)\, d\boldsymbol{\phi}' + I_1(\phi_1, t) + I_2(\phi_2, t), \tag{33}$$

where $u_g(\boldsymbol{\phi}, t)$ and $r_g(\boldsymbol{\phi}, t)$ are the synaptic input and firing rate of a grid cell at phase $\boldsymbol{\phi} = (\phi_1, \phi_2)$. The recurrent connectivity J_g is translation-invariant and shaped to support hexagonal grid patterns:

$$J_g(\boldsymbol{\phi}, \boldsymbol{\phi}') = \frac{J_0^g}{2\pi\sigma_g^2} \exp\left[-\frac{||\boldsymbol{\phi} - \boldsymbol{\phi}'||_g^2}{2\sigma_g^2}\right], \tag{34}$$

with firing rates given by a divisive normalization:

$$r_g(\boldsymbol{\phi}, t) = \frac{u_g^2(\boldsymbol{\phi}, t)}{1 + k_g \iint u_g^2(\boldsymbol{\phi}', t)\, d\boldsymbol{\phi}'}. \tag{35}$$

The distance metric $||\cdot||_g$ between grid cell phases is defined to produce a hexagonal pattern. Specifically, letting $\Delta\boldsymbol{\phi} = \boldsymbol{\phi} - \boldsymbol{\phi}'$, we apply periodic wrapping and a linear transformation:

$$\Delta\phi_1 = \mathrm{wrap}(\phi_1 - \phi_1'), \tag{36}$$

$$\Delta\phi_2 = \mathrm{wrap}(\phi_2 - \phi_2'), \tag{37}$$

$$\delta_x = \Delta\phi_1, \tag{38}$$

$$\delta_y = \frac{2}{\sqrt{3}}\left(\Delta\phi_2 - \frac{1}{2}\Delta\phi_1\right), \tag{39}$$

$$||\boldsymbol{\phi} - \boldsymbol{\phi}'||_g = \sqrt{\delta_x^2 + \delta_y^2}, \tag{40}$$

where $\mathrm{wrap}(\cdot)$ maps values to $(-\pi, \pi]$ to account for periodic boundary conditions. The linear transformation aligns the phase coordinates with the axes of a hexagonal lattice.

This design ensures that the bump activity on the grid cell manifold has a hexagonal structure rather than a square one, consistent with biological grid cell tuning.

4. Reciprocal connections between grid and band modules

Bidirectional connections allow information exchange between grid and band cells.

From grid cells to band cells:

$$I_g(\phi_k^b, t) = \iint W_{gb}(\phi_k^b, \boldsymbol{\phi}')\, r_g(\boldsymbol{\phi}', t)\, d\boldsymbol{\phi}', \quad k = 1, 2 \tag{41}$$

From band cells to grid cells:

$$I_k(\phi_k, t) = \int W_{gb}(\boldsymbol{\phi}, \phi')\, r_b(\phi', t)\, d\phi', \quad k = 1, 2 \tag{42}$$

The weights are symmetric and phase-aligned:

$$W_{gb}(\phi_k^b, \boldsymbol{\phi}) = \frac{W_{gb}^0}{\sqrt{2\pi}\sigma_{gb}} \exp\left[-\frac{|\phi_k^b - \phi_k^g|_\pi^2}{2\sigma_{gb}^2}\right]. \tag{43}$$

F.2 Parameters

The model is composed of multiple *path integration modules*, each responsible for encoding spatial location at a specific scale. Each module consists of two **band cell modules** and one **grid cell module**. The two band modules within a path integration module differ only in their preferred orientation (offset by $60°$), but share all other dynamic and connectivity parameters. The five modules operate at distinct spatial scales $\lambda^{(i)}$, which influence their velocity projection vectors $\mathbf{k}^{(i)}$ and resulting grid periodicity.

Within each module, band cells perform velocity integration using directionally tuned input, coupled through asymmetric recurrent and cross-module connections. Their outputs drive the activity of grid cells, which integrate spatial phase across the two orientations to generate hexagonal grid patterns. The grid cell dynamics are shaped by their own recurrent connectivity and normalization mechanisms.

All modules share a common set of parameters for band and grid dynamics, except for the spatial scale λ and the orientation θ, which define the velocity-to-phase projection. These design choices allow for a consistent multi-scale encoding of space, which is later used for spatial decoding via a grid-to-place projection.

The table below summarizes all parameters used in the model and their hierarchical roles.

F.3 Simulation details

We performed two simulations to evaluate the functionality of the proposed neural circuit model for path integration.

Simulation 1: Activity mapping in full environment. We simulated a freely foraging rat in a square box environment, generating a trajectory of $100{,}000$ time steps with a fixed time step size $\Delta t = 0.05$. The simulated trajectory densely covers the entire spatial environment. At each time step, we recorded the activity of different neural populations in the model, including band cells (from each module) and grid cells. We then computed the average firing rate at each spatial location to obtain the spatial tuning maps (i.e., heatmaps) for different cells.

Simulation 2: Decoding trajectory from neural activity. In this simulation, we generated a continuous trajectory of $1{,}000$ time steps, again with $\Delta t = 0.05$. Using the spatial tuning maps (heatmaps) of grid cells obtained from Simulation 1, we decoded the animal's position at each time step from the instantaneous population activity of grid cells. The decoding procedure was carried out in two stages:

Table S3: Parameters of the path integration model. Each path integration module consists of two band cell modules (offset by 60° in orientation) and one grid cell module, with shared dynamics and connectivity parameters.

Symbol	Description	Value
Global Parameters		
Δt	Simulation time step	5 ms
n_λ	Number of path integration modules (scales)	5
$\lambda^{(i)}$	Spatial scale of module i	$2.5 + 0.3i$
$\theta^{(1)}, \theta^{(2)}$	Orientation of the two band cell modules	$0°, 60°$
Band Cell Modules (per module, shared across scales)		
τ	Time constant of pure band cell dynamics	100 ms
τ	Time constant of conjunctive band cell dynamics	10 ms
N_b	Neuron number of pure band cells	180
N_+	Neuron number of v^+ band cells	180
N_-	Neuron number of v^- band cells	180
J_b^0	Strength of recurrent excitation in pure band cells	1.1
σ_b	Width of Gaussian kernel for recurrent/feedforward connections	$\frac{2}{9}\pi$
k_b	Inhibition strength in divisive normalization	$5 * 10^{-4}$
w_b	Strength of pure-to-conjunctive band cell connections	1
W_b^0	Strength of conjunctive-to-pure feedback connections	0.2
δ	Offset in phase space from direction-specific input $v^\pm$	0.265
g_0	Baseline activity level of conjunctive band cells	0.2
Grid Cell Module (per module, shared across scales)		
τ_g	Time constant of grid cell dynamics	10ms
J_0^g	Strength of recurrent connectivity among grid cells	1
σ_g	Width of grid cell recurrent kernel	$\frac{1}{9}\pi$
k_g	Inhibition strength in grid cell divisive normalization	$5 * 10^{-3}$
W_{gb}^0	Strength of connection between band and grid cells	0.1
σ_{gb}	Width of Gaussian kernel for band-to-grid connections	$\frac{2}{9}\pi$

1. **Grid-to-place cell transformation:** The grid cell activity vector at each time step was linearly projected onto a population of place cells. Each place cell was assigned a spatial location (x, y) on a uniform 2D grid. To define the synaptic connection between a grid cell and a place cell, we first projected the place cell's spatial location into the phase space (ϕ_1, ϕ_2) of the two band cell modules used to construct grid cells. This projection was done by computing the inner product of the place field center (x, y) with the two module wave vectors $\mathbf{k}_1$ and $\mathbf{k}_2$, i.e.,

$$\phi_k = 2\pi \cdot \mathrm{mod}\left(\mathbf{k}_k \cdot (x, y), 1\right) - \pi, \qquad k = 1, 2.$$

This yields the wrapped phases (ϕ_1, ϕ_2) corresponding to the grid cell's toroidal phase space. The connection strength between each grid cell (with preferred phase $\boldsymbol{\phi}^g$) and a place cell (located at (x, y) with corresponding $\boldsymbol{\phi}^{\mathrm{pc}} = (\phi_1, \phi_2)$) is then defined by a Gaussian kernel on the torus:

$$W_{\mathrm{pc},g}(\boldsymbol{\phi}^{\mathrm{pc}}, \boldsymbol{\phi}^g) = \frac{W_0^{\mathrm{pc}}}{2\pi\sigma_{\mathrm{pc}}^2} \exp\left(-\frac{\|\boldsymbol{\phi}^{\mathrm{pc}} - \boldsymbol{\phi}^g\|_g^2}{2\sigma_{\mathrm{pc}}^2}\right),$$

where the toroidal phase distance $\|\cdot\|_g$ is computed using the same hexagonal metric as in the grid cell CANN. Specifically, for $\boldsymbol{d} = \boldsymbol{\phi}^{\mathrm{pc}} - \boldsymbol{\phi}^g$ (wrapped to $[-\pi, \pi]$),

$$\|\boldsymbol{d}\|_g = \sqrt{\delta_x^2 + \left(\frac{\delta_y - \frac{1}{2}\delta_x}{\sqrt{3}/2}\right)^2},$$

with $\delta_x = d_1$ and $\delta_y = d_2$.

This construction ensures that the grid-to-place connection matrix respects the periodic and hexagonal structure of the underlying grid cell code.

2. **Population vector decoding:** After obtaining the activity of place cells from the grid cell input, we decoded the position by computing a population vector (center-of-mass) over place field centers, weighted by the activity of each place cell at each time step.

We then evaluated the model's path integration accuracy by comparing the decoded positions with the true trajectory. Results obtained under idealized conditions—where both the input signals and the model were free of random noise—are presented in Fig. 5. In more realistic noisy scenarios, small errors accumulate over time, resulting in a gradual drift in the estimated position, as illustrated in Fig. S5.

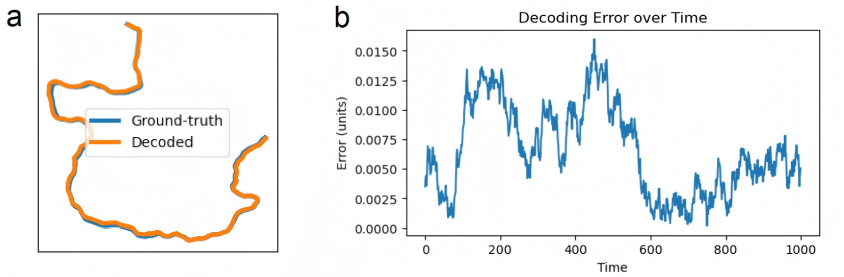

Figure S5: (a) Inferred trajectory of the noisy neural circuit model compared with the ground-truth trajectory. (b) Decoding error of the neural circuit under noisy conditions.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction truthfully summarize the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The discussion section explicitly examines the study's limitations, ensuring transparent acknowledgment of the work's boundaries.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides sufficient methodological details, hyperparameters, and experimental setups to allow reproduction of the key results supporting its claims. Details are presented in Sec. 2, 3 and SI.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the link to our publicly available Github repository in the SI.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper thoroughly documents all the training and test details required to comprehend and replicate the reported findings in SI.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports error bars and provides appropriate information about the statistical significance of the experiments, as evidenced in the presented figures and SI.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The manuscript does not explicitly report computational resource requirements, as the lightweight architecture of the proposed model renders such specifications non-essential for reproducibility.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research fully complies with the NeurIPS Code of Ethics in all aspects by ensuring transparency, reproducibility, and fairness in reporting experimental results.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We appropriately acknowledge all third-party code and data sources through proper citations of original works and the code base is also linked in SI.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The completeness of documentation for newly introduced assets will be verified upon release of the data and code on GitHub, ensuring all reproducibility requirements are met.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.