The limbic navigation system as a hierarchical RNN

Zilong Ji[*,†], **Huiwen Zhang**[*], **Krishna Gorantla, Neil Burgess**[†]

UCL Institute of Cognitive Neuroscience, University College London, London, UK
UCL Queen Square Institute of Neurology, University College London, London, UK

[*]Equal contribution.
[†]Corresponding authors.

Abstract

Path-integration-trained recurrent neural networks (RNNs) can reproduce grid-like and head-direction-like representations, but most existing models rely on low-dimensional velocity inputs and do not capture the hierarchical organisation of the brain's self-localisation circuit. We introduce a hierarchical recurrent neural network (HRNN) that separately processes angular velocity and linear speed to perform simultaneous angular and positional path integration. Trained on these tasks, the HRNN generalises to longer timescales than seen during training and develops neural response properties resembling all the key cell types: angular velocity cells, head direction cells, speed cells, conjunctive and pure grid cells. The learned circuit also recovers key motifs of the navigation system, including attractor dynamics and asymmetric connectivity patterns. Perturbation analyses reveal functional information flow across the hierarchy, consistent with experimental observations. Finally, introducing theta-paced mutual inhibition between two learned angular-velocity subpopulations enables the HRNN to reproduce the bidirectional theta sweeps observed in empirical data. These results establish hierarchical task-driven RNNs as a biologically interpretable framework for linking neural representations, circuit motifs, and population dynamics in spatial navigation, while enabling targeted in silico perturbations and the generation of experimentally testable predictions.

1 Introduction

Recurrent neural networks (RNNs) trained with machine learning techniques on cognitive tasks have become a widely adopted tool for neuroscientists to study biological neural circuits [39]. This approach has achieved notable success across multiple domains, including modelling neural activity in visual cortex [20], motor systems [19], and prefrontal cortex [17]. In the domain of spatial navigation, RNNs trained to perform path integration (PI) tasks have been shown to give rise to representations resembling head direction cells [9; 34] and grid cells [8; 1; 28]. Interestingly, beyond reproducing neural activity patterns, these models also exhibit emergent connectivity structures that mirror anatomical motifs observed in biological circuits [9; 7]. This suggests that task-driven RNNs are not merely "black boxes," but can serve as biologically interpretable models that recapitulate key structural principles of neural systems.

In contrast to mechanistic models—such as continuous attractor neural networks for head direction and grid-cell circuits[27; 41; 5]—which rely on predefined connectivity structures, normative models do not impose such assumptions. Instead, they learn synaptic connectivity directly from task demands, often rediscovering structures consistent with those proposed by mechanistic approaches. As a result, these models provide a complementary perspective, offering a framework to explain why neural circuits take their observed forms across both evolutionary processes and postnatal experience-dependent development.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Despite these advances, current approaches remain limited in their ability to capture the complexity of the spatial navigation circuit. This system depends on the coordinated function of multiple brain regions, each specialized for distinct computational roles. Here we use the term ***limbic navigation system*** to refer broadly to the interconnected circuits that support self-localisation and path integration, including upstream pathways for directional information as well as downstream entorhinal-hippocampal pathways for spatial information (Fig. 1A). For instance, head direction cells in the anterodorsal thalamus and postsubiculum encode orientation information and depend on angular velocity signals from the dorsal tegmental nucleus (DTN) and lateral mammillary nuclei (LMN) [30]. Downstream, grid cells in the entorhinal cortex are thought to continuously track an animal's position in space [18; 5], relying on upstream head direction and linear speed signals [22; 38]. It is noteworthy that head direction and movement direction are distinct signals, and movement direction, found in a parallel circuit to head-direction [36], is more directly relevant for path integration. Here, we do not dissociate them and instead assume that they are aligned.

In contrast, most task-driven RNN models rely on a single network trained on a single task with a single optimization objective to capture neural activity patterns, cell types, and circuit motifs, although some studies have begun to explore multi-module or multi-network architectures [34]. Some of these single-RNN models may develop functional submodules during training [40], these emergent structures are often difficult to map onto anatomically distinct areas (but see Cueva et al. [9]). Therefore, they are often limited in their ability to account for the flow of information across interconnected brain regions. Consequently, they fall short of explaining the diversity, organization, and interactions of cell types and circuit motifs distributed across multiple brain regions.

Here, we propose a hierarchical recurrent neural network (HRNN) model trained to perform simultaneous self-localisation of orientation and location. We demonstrate that after training, it recapitulates key organizational principles of the brain's self-localisation system along three dimensions. First, it reproduces a range of cell types observed across the limbic navigation system, including angular velocity cells [25], head direction cells [31], linear speed cells [15; 13], grid cells [11] and conjunctive direction by grid cells [22]. Second, it captures anatomical motifs, including attractor dynamics reported in the head direction circuit [6], asymmetric synaptic connections [33], aligned directional tuning between functionally connected direction cells and conjunctive grid cells [35], as well as transformations across layers of the medial entorhinal cortex [22]. Third, it captures hierarchical population-level dynamics [29; 16; 35], including recently discovered alternating left-right theta sweeps [35].

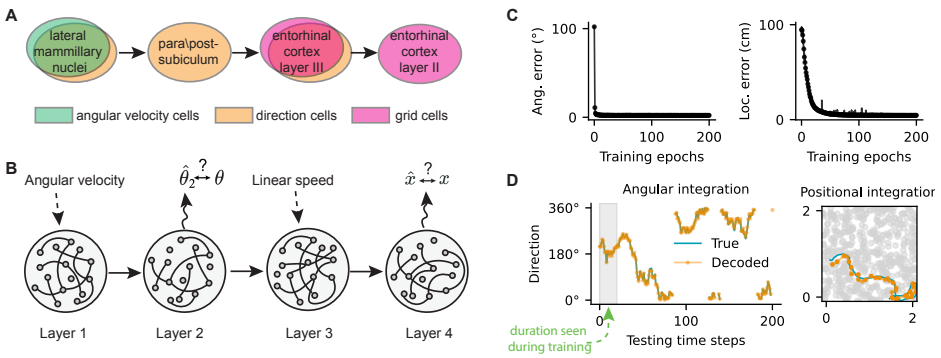

Figure 1: Task-driven hierarchical RNN (HRNN) for simultaneous self-orientation and localisation. (A) Path-integration-related modules in the brain's limbic navigation system. Different colors indicate different cell types involved in path integration. (B) Schematic of the four-layer HRNN, with each layer corresponding to a module in (A). Angular velocity is provided to the first layer, and the second layer predicts orientation. The third layer receives input from the second layer together with linear speed, and the fourth layer predicts location in 2D space. (C) Test errors across training epochs for angular integration (left) and 2D path integration (right). (D) Path-integration performance after training. Left: angular integration. Right: 2D path integration. The shaded region on the left indicates the time window used during training. The trained HRNN can perform path integration over much longer timescales than those seen during training.

The emergence of distinct functional cell types and anatomical motifs enables targeted perturbation experiments in the HRNN, allowing us to investigate the contributions of specific cell populations to self-localisation, in parallel with causal manipulations performed in vivo [38]. Taken together, these results suggest that the HRNN has the potential to serve as a digital twin of the brain's spatial navigation system, providing a powerful platform for mechanistic investigation and targeted perturbation studies that are difficult to perform in vivo.

2 Task-driven hierarchical RNN (HRNN) for self-orientation and localisation

The HRNN consists of four stacked RNNs (Fig. 1B; see Appendix A and Table 1 for detailed parameter settings). Each layer is a "vanilla" RNN [28; 1]. For layer $\ell \in \{1, 2, 3, 4\}$, the recurrent update is

$$\mathbf{r}^{(\ell),t+1} = f\Big(\mathbf{J}^{(\ell)}\mathbf{r}^{(\ell),t} + \mathbf{M}^{(\ell)}\mathbf{I}^{(\ell),t}\Big). \tag{1}$$

Here, $\mathbf{r}^{(\ell),t} \in \mathbb{R}^{n_\ell}$ is the activity vector of layer ℓ at time t, and $\mathbf{J}^{(\ell)} \in \mathbb{R}^{n_\ell \times n_\ell}$ is its recurrent weight matrix. The feedforward input is $\mathbf{I}^{(\ell),t} \in \mathbb{R}^{m_\ell}$, and its weight matrix is $\mathbf{M}^{(\ell)} \in \mathbb{R}^{n_\ell \times m_\ell}$. Specifically, $\mathbf{I}^{(1),t} = [\omega_t]$, $\mathbf{I}^{(2),t} = \mathbf{r}^{(1),t}$, $\mathbf{I}^{(3),t} = [(\mathbf{r}^{(2),t})^\top, s_t]^\top$, and $\mathbf{I}^{(4),t} = \mathbf{r}^{(3),t}$, where ω_t and s_t are angular velocity and linear speed, respectively. Thus, $(m_1, m_2, m_3, m_4) = (1, n_1, n_2 + 1, n_3)$. The function $f(\cdot)$ is a ReLU nonlinearity, ensuring that neural activity remains non-negative.

RNN layer 2 outputs the activities of a group of cells coding for orientation, i.e., direction cells:

$$\hat{d}_i^t = \sum_{j=1}^{n_2} W_{ij}^{(d)} r_j^{(2),t}. \tag{2}$$

Here, $\hat{d}_i^t$ is the predicted activity of direction cell i, n_d is the number of direction cells, and $\mathbf{W}^{(d)} \in \mathbb{R}^{n_d \times n_2}$ is the direction-readout matrix. Layer 4 outputs the activities of a group of cells coding for location, i.e., place cells:

$$\hat{p}_i^t = \sum_{j=1}^{n_4} W_{ij}^{(p)} r_j^{(4),t}. \tag{3}$$

Here, $\hat{p}_i^t$ is the predicted activity of place cell i, n_p is the number of place cells, and $\mathbf{W}^{(p)} \in \mathbb{R}^{n_p \times n_4}$ is the place-readout matrix.

The network is trained to simultaneously reconstruct direction cell activities from layer 2 and place cell activities from layer 4:

$$\mathcal{L}(\Theta) = -\sum_{t=1}^{T} \left[\sum_{i=1}^{n_d} d_i^t \log \hat{d}_i^t + \sum_{i=1}^{n_p} p_i^t \log \hat{p}_i^t \right]. \tag{4}$$

Here, d_i^t and p_i^t denote the target direction- and place-cell activities encoding the ground-truth direction and position, while $\hat{d}_i^t$ and $\hat{p}_i^t$ are the corresponding predictions. The two terms are the cross-entropy losses for the layer-2 direction readout and the layer-4 place readout, respectively. The parameter set Θ contains all recurrent, feedforward, and readout weights (see Table 1). We minimise $\mathcal{L}(\Theta)$ using gradient descent while continuously predicting orientation and position over $T = 20$ time steps; prediction errors decrease rapidly during training (Fig. 1C).

After training, the network successfully predicts direction and position trajectories that closely overlap with the ground truth (Fig. 1D). Notably, although it is trained only on short-duration sequences, the network can perform angular and positional path integration over much longer timescales.

3 HRNN captures multiple neural activity patterns

After training converged, we examined the spatial tuning of units in the HRNN and found a variety of cell types with distinct tuning properties that closely resemble those reported in the experimental literature (Fig. 2) [31; 25; 29; 24; 11; 22; 15].

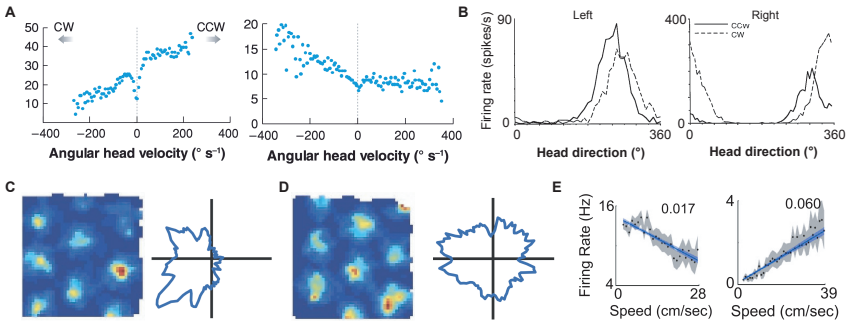

Figure 2: Experimentally identified cell types in the limbic navigation system for self-orientation and localisation. (A) Two angular velocity (AV) cells, adapted from [2]. (B) Two head direction cells showing different firing rates as a function of turning direction, adapted from [29]. (C) A conjunctive grid cell, adapted from [4]. (D) A pure grid cell, also adapted from [4]. (E) Two linear speed cells, one negatively tuned (left) and one positively tuned (right) to linear speed, adapted from [13].

Angular velocity cells: We first showed the emergence of angular velocity (AV) cells in the first two layers (Fig. 3A). Cells were classified as AV-tuned if they satisfied two criteria: an absolute AV tuning index greater than 0.2 and a linear-fit R^2 greater than 0.5 (see Appendix D). Using these criteria, most of the cells in layer 1 showed AV tuning (218/256), whereas a relatively smaller population in layer 2 showed AV tuning (158/256; Fig. 3B; Fisher's exact test, odds ratio = 3.56, p = 2.3×10^{-9}; see Fig. S8 for all AV tuning maps). This result is consistent with the decrease in AV cells along the head-direction circuit, with a higher proportion in the lateral mammillary nuclei (LMN; [29]) than in the downstream postsubiculum (PoS; [25]).

Furthermore, AV cells can be divided into two subgroups (Fig. 3A&C): one increases the firing rate for counter-clockwise (CCW) turns (angle increases from 0° to 360° with a positive slope; AV+), and the other increases the firing rate for clockwise (CW) turns (angle decreases from 360° to 0° with a negative slope; AV-). The numbers of cells in these two groups are balanced. Knocking out one group causes the directional representation to drift continuously in the opposite direction, yielding an experimentally testable prediction (see Section 5 below).

Head direction cells: Most of the cells in the first two layers are also conjunctively tuned to directional information (Fig. 3C; see also Fig. S8 for all directional tuning maps). Furthermore, AV+ cells show preferred firing for CCW turns while AV- cells show preferred firing for CW turns. Interestingly, directional tuning is stronger in layer 2 compared to layer 1 (Fig. 3D; median mean vector length (MVL) [IQR]: 0.74 [0.67, 0.80] in layer 2 vs. 0.60 [0.53, 0.65] in layer 1; Mann-Whitney $U = 6.6 \times 10^3$, $p = 1.2 \times 10^{-53}$; see Appendix E for calculation). This result recapitulates the stronger directional tuning observed in later hierarchies of the direction circuit than in earlier hierarchies in the rodent brain [3].

Conjunctive and pure grid cells: In the HRNN, layer 3 receives directional input from layer 2 together with linear speed input and relays this combined signal to layer 4, which is trained for 2D path integration. Cells in layer 3 exhibited tuning maps resembling conjunctive direction-by-grid cells (Fig. 3E; see Appendix F for calculation details), whereas cells in layer 4 resembled pure grid cells (Fig. 3F). This recapitulates empirical findings that mEC layer-3 grid cells show conjunctive directional tuning, whereas mEC layer-2 grid cells downstream are not [11; 22]. The HRNN recapitulates this pattern without explicit constraints (Fig. 3G,H). Using a grid-score threshold of 0.3 and a directional mean-vector-length threshold of 0.3, we identified 33/512 conjunctive grid cells in layer 3 and only 3/512 in layer 4. We also found a large population of cells with weak grid and head-direction tuning (55/512 in layer 3 and 242/512 in layer 4; Fig. S9), which may still be important for accurate position readout because a single grid module alone yields ambiguous location estimates.

Speed cells: Because linear speed input was provided to layer 3, we also observed speed cells in layers 3 and 4 (Fig. 3I; Fig. S9), consistent with experimental reports [15; 13; 14]. In layer 3,

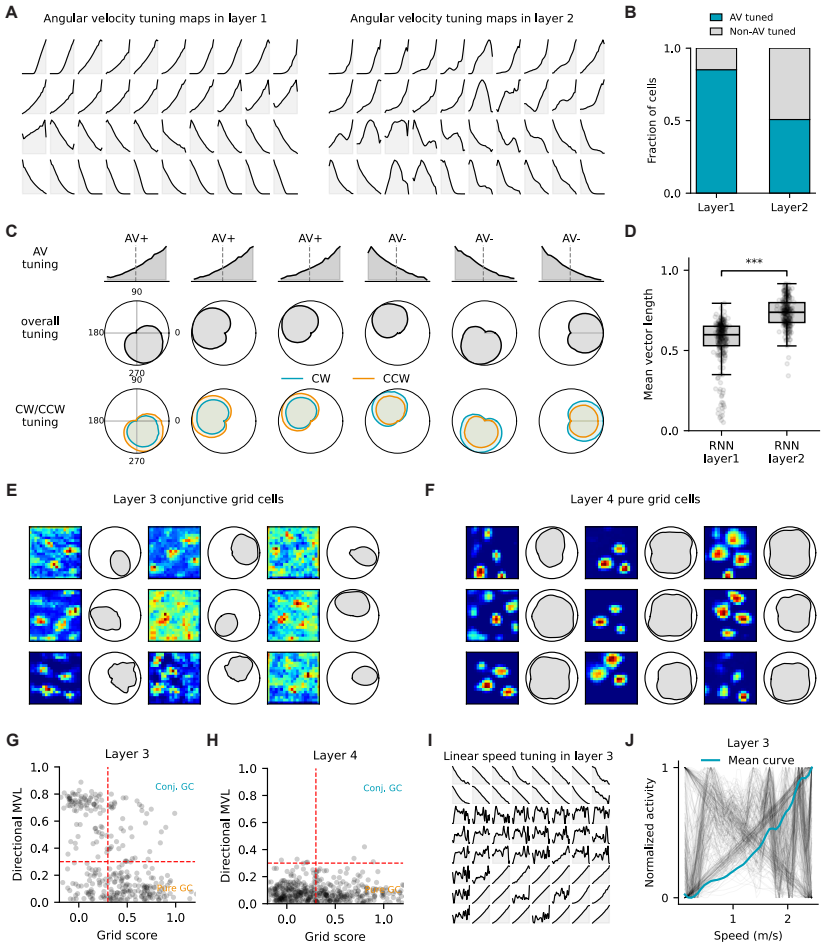

Figure 3: HRNN recapitulates multiple neural activity patterns observed in the self-localisation system. (A) Example angular velocity (AV) tuning cells in layers 1 and 2, respectively. (B) AV tuning strength in layers 1 and 2, showing a significant decrease in AV tuning strength in layer 2 (Fisher's exact test, odds ratio = 3.56, p = 2.3×10^{-9}). (C) Examples of conjunctive AV tuning (row 1) and head-direction tuning (row 2), with clockwise (blue) and counter-clockwise (orange) directional tuning plotted separately. Positive AV-tuned cells preferentially fire in the counter-clockwise (CCW) direction, whereas negative AV-tuned cells preferentially fire in the clockwise (CW) direction. (D) Directional tuning is stronger in layer 2 than in layer 1, reflected by the directional mean vector length, with Mann-Whitney $U = 6.6 \times 10^3$, $p = 1.2 \times 10^{-53}$. (E) Conjunctive grid-by-direction tuning cells in layer 3. (F) Pure grid cells in layer 4. (G) Distribution of grid scores and direction mean vector lengths for all active cells in RNN layer 3. (H) Same as (G) but in layer 4. (I) Linear speed cells in layer 3, including both positive and negative speed-tuned cells. (J) Linear speed tuning curves for all active cells in layer 3 overlaid in one plot. The blue line denotes the average linear speed tuning curve.

many cells either increased or decreased their firing rates approximately linearly with running speed, although the majority showed positive speed tuning, as reflected in the mean tuning curve (Fig. 3J). Both positive and negative speed tuning have been observed empirically [13]. In layer 4, by contrast, speed tuning was often nonlinear rather than strictly linear, with many cells reaching saturation before intermediate speeds were attained (Fig. S9), again consistent with experimental data [13]. This layer-specific difference in speed tuning yields an experimentally testable prediction.

In summary, the HRNN recapitulates many experimentally identified cell types in the limbic navigation system. Importantly, these cell types are rarely selective for only a single variable such as

direction or location. Instead, they are often conjunctively tuned to multiple features. This supports the view that navigation relies on multiplexed neural codes [12].

4 HRNN captures anatomical motifs

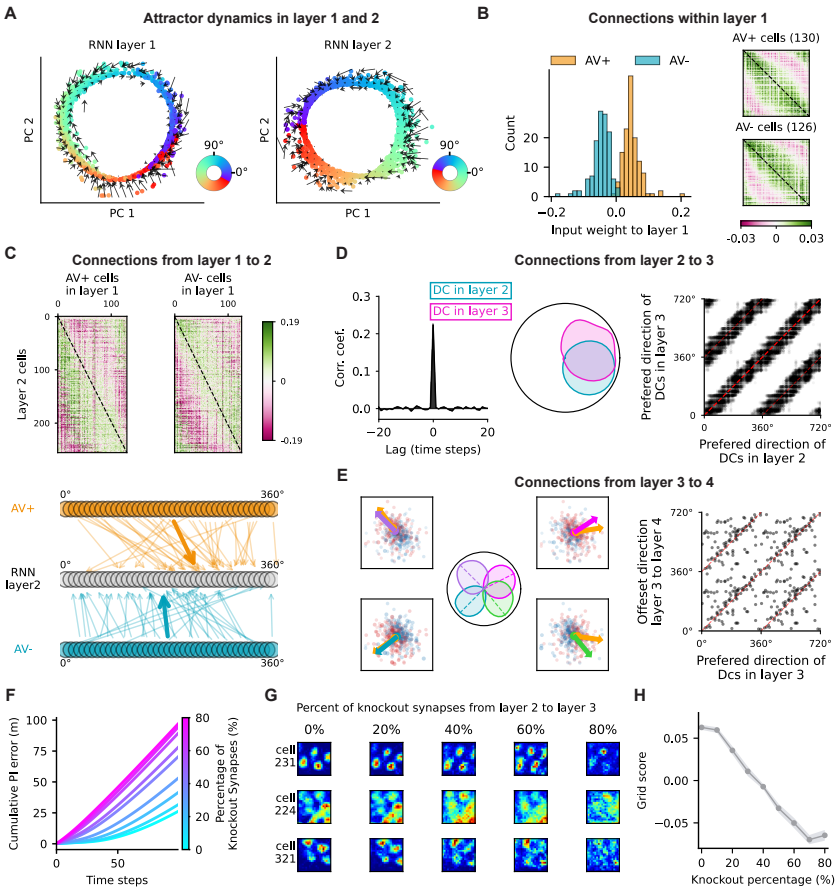

Figure 4: HRNN recapitulates anatomical motifs. (A) Ring-attractor dynamics in layers 1 and 2. Each dot shows the network state projected onto the first two principal components and colored by movement direction. Dark arrows indicate that off-manifold states are driven back toward the attractor manifold. (B) Left: distribution of learned input weights to layer 1, with connections to AV+ and AV- cells shown in orange and blue. Right: asymmetric recurrent connectivity matrices, ordered by preferred direction, for AV+ (top) and AV- (bottom) cells. (C) Top: asymmetric feedforward connectivity from AV+ (left) and AV- (right) cells to layer 2 neurons. Bottom: strongest synaptic connections from AV+/AV- cells to layer 2 cells, highlighting the same asymmetry. (D) Left: cross-correlation of firing sequences for one layer 2 neuron (blue) and one layer 3 neuron (pink); a peak near zero lag indicates co-activation. Middle: directional tuning of the same pair. Right: preferred directions of all functionally connected layer 2-layer 3 pairs, repeated for two cycles for better visualisation. (E) Left: four example layer 3 direction cells. Arrows (not in the orange color) indicate preferred firing direction; red and blue dots mark the spatial firing center of positively and negatively connected layer 4 cells, with dot size proportional to connection strength; the orange dot marks the weighted mean spatial offset. Right: preferred direction of layer 4 cells versus the offset direction of the mean spatial map center in layer 4. (F) Cumulative path-integration error increases as the proportion of pruned synapses from layer 2 to 3 increases. (G) Hexagonal firing fields of three example grid cells in layer 4 progressively degrade with increasing pruning of layer 2-to-layer 3 synapses. (H) Mean grid score across all 4 cells decreases as the pruning percentage increases.

Here we further showed that HRNN recapitulates anatomical motifs reported in the literature [27; 33; 6; 35].

4.1 Attractor dynamics and asymmetric connections for directional coding in layer 1 and 2

Population activity in the head-direction circuit has been shown to lie on a one-dimensional ring-attractor manifold embedded in a high-dimensional neural state space [6]. We found that both layers 1 and 2 emerge ring-attractor dynamics (Fig. 4A). We next examined the flux of network states on and off the manifold using the analysis proposed by Chaudhuri et al. [6] (see Appendix G for more details). This analysis allowed us to test whether states that were temporarily perturbed away from the manifold would return to it, providing evidence for attractor dynamics. We found approximately zero net flux along the manifold, consistent with the isotropic distribution of flow directions along the ring. In contrast, states away from the manifold showed larger net fluxes directed back toward it, consistent with the attractor acting as a low-energy basin (Fig. 4A).

Many previous mechanistic models of angular integration share a common architecture, although they differ in implementation details [27; 26; 41]. This architecture includes (1) cells that encode angular velocity, (2) cells with conjunctive tuning to angular velocity and head direction, and (3) asymmetric synaptic connectivity that shifts the activity bump around the ring. We have already shown the first two features above and here further show that the first two layers also develop asymmetric synaptic connectivity. Specifically, an AV+ cell with a given preferred direction formed its strongest connection with an AV+ cell whose preferred direction was slightly shifted counterclockwise (Fig. 4B). By contrast, an AV- cell connected most strongly to an AV- cell whose preferred direction was slightly shifted clockwise. This learned asymmetry allows angular velocity inputs of opposite signs to shift the activity bump on the ring manifold in opposite directions. Although this pattern is difficult to identify in rodents, it has been elegantly demonstrated in the fly head-direction system [33] and was also captured by a single-layer RNN trained for angular integration [9].

Furthermore, AV+ and AV- cells projected asymmetrically to direction cells in layer 2, but with the opposite shift direction (Fig. 4C). This connectivity pattern has previously been hypothesized in mechanistic models that hard-code this feature to implement angular integration [27; 26; 41]. Here, we show that in a network trained for angular integration, this type of asymmetric connectivity emerges naturally.

4.2 Direction-dependent shift for spatial coding in layer 3 and 4

Similar to angular integration, 2D path integration requires the network to continuously update position relative to a starting point using momentary speed and direction. We next examined how this information is conveyed through the learned motifs in the last two layers. We found that direction cells in layer 2 were functionally coupled to direction cells in layer 3 with a similar preferred direction, some of which were conjunctive grid cells as reported in Vollan et al. [35] (Fig. 4D). To identify co-active pairs, we computed the cross-correlation of their firing sequences; pairs with a peak at zero lag were classified as functionally connected. Among 256 direction cells in layer 2 and 105 direction cells in layer 3 (mean vector length, MVL > 0.5), we identified 7337 functionally connected pairs whose preferred directions were strongly aligned (Fig. 4D).

Because direction cells in layer 3 are also spatially tuned, we next asked how their activity drives position updates in layer 4. We found that a direction cell in layer 3 tends to activate layer 4 cells ahead of the same direction when the agent moves along that direction, and suppress layer 4 cells on the opposite direction (Fig. 4E). We further computed the center of each layer 4 spatial field and weighted it by the absolute value of the feedforward connection from a probe layer 3 cell (see Appendix H for the detailed calculation). Across 103 identified direction cells in layer 3, the resulting mean spatial offset in layer 4 was aligned with the probe cell's preferred firing direction (Fig. 4E).

These results suggest that movement in a given direction activates direction cells in layer 2, which in turn recruit layer 3 direction cells with similar preferred directions. These cells then shift population activity in layer 4 along the same direction, thereby continuously updating position from self-motion cues.

4.3 Disrupting direction signal impairs grid cell firing

Because the HRNN recapitulates cell types and anatomical connectivity patterns found in the brain, it enables in silico targeted perturbation experiments to test how perturbing one cell type affects the activity and function of others.

Here, we investigated how directional signals influence the generation and function of grid cell activity. Previous experiments have shown that disrupting head direction cells in the anterodorsal thalamic nucleus (ADN), either by lesioning the ADN or by transiently inactivating it with lidocaine, impairs grid cell firing patterns [38]. To test whether the HRNN reproduces this phenomenon, we pruned synaptic connections from layer 2 to layer 3 and evaluated both path-integration performance and grid cell activity. Increasing the degree of pruning led to a more rapid increase in path-integration error over time (Fig. 4F) and progressively degraded grid firing patterns (Fig. 4G,H), consistent with the empirical findings [38].

5 Left-right theta sweeps via mutual inhibition between the AV+ and AV- modules in HRNN layer 1

Recent empirical data showed that, during free movement, decoded positions from mEC population activities exhibit rhythmic "theta sweeps," transiently representing locations extending away from the animal's current position at theta frequency and alternating between left and right across successive theta cycles (Fig. 5A) [35]. These left-right sweeps in open space are further driven by bidirectional sweeps from side to side of the head axis in the parasubiculum (Fig. 5B).

We have shown that, AV+ cells in layer 1 preferentially project to downstream directional cells in layer 2 whose tuning fields are slightly shifted to the counter-clockwise side of the AV+ tuning fields, whereas AV- cells preferentially project to downstream directional cells whose tuning fields are slightly shifted to the clockwise side of the AV- tuning fields (Fig. 4C and Fig. 5C). Consistent with this connectivity, knocking out AV+ cells causes the activity bump in layer 2 to drift clockwise, whereas knocking out AV- cells causes it to drift counter-clockwise (Fig. 5D).

Based on this mechanism, we alternately inhibited AV+ and AV- cell populations across successive theta cycles and added cortical sensory feedback to pin the activity bump back to the movement direction at the end of each theta cycle (Fig. 5E). This theta-paced mutual inhibition produced bidirectional theta sweeps in layer 2 (Fig. 5F). Moreover, downstream grid cells in layer 4 exhibited alternating left-right theta sweeps (Fig. 5G&H), as observed in empirical data [35].

In summary, these results show that the HRNN trained on path-integration tasks can capture not only anatomical motifs, but also population-level dynamics that arise from these learned motifs. Notably, these dynamics emerge without imposing explicit biophysical constraints on the HRNN.

6 Model comparison

To distinguish the contributions of modular input-output organisation and hierarchical information routing, we compared the HRNN with three control architectures (Fig. S2A–D). Here, modularity refers to the assignment of angular-velocity and linear-speed inputs to different modules, with direction and position read out from their respective processing stages. Hierarchy refers to the restriction of inter-module connections to sequential feedforward pathways. The flat RNN receives both inputs and produces both outputs from a single recurrent population. The unstructured HRNN retains four serially connected recurrent layers, but receives both inputs in layer 1 and produces both outputs from layer 4. The fully connected multi-module RNN preserves the HRNN's input-output arrangement while allowing recurrent connections between all modules. These controls therefore distinguish the effects of specialised input–output organisation from those of constrained inter-module routing.

We conducted both parameter-matched and unit-number-matched comparisons; architectural details, training results and quantitative analyses are provided in Appendix C. Because parameter matching required smaller populations in the fully connected control, we focus here on the unit-number-matched comparison to assess the contributions of modularity and hierarchy independently of differences in population size.

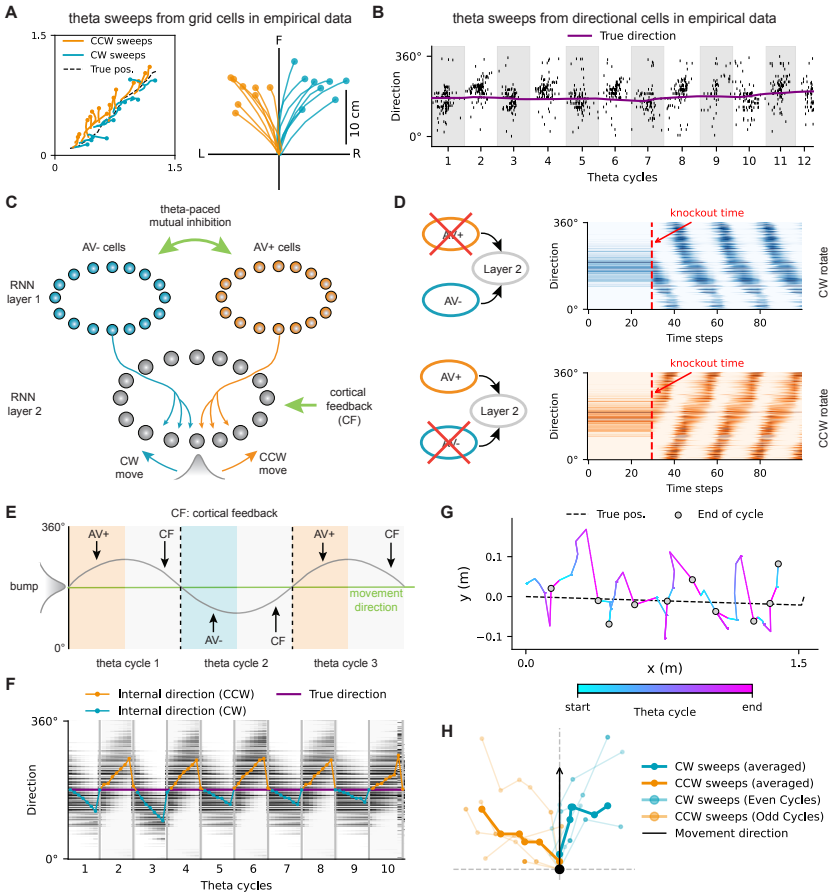

Figure 5: HRNN recapitulates left-right theta sweeps via theta-paced mutual inhibition between the AV+ and AV- cell populations. (A) Left-right theta sweeps decoded from grid cell population activity with data from [35]. Left: left-right sweeps along a straight trajectory. Right: left-right sweeps with initial decoded location recentered at the origin and movement direction pointing to the north. (B) Bidirectional theta sweeps decoded from direction cells in the parasubiculum, with data from [35]. (C) A schematic of the AV+ and AV- populations in layer 1 and the feedforward asymmetric connections from layer 1 to 2. (D) Knocking out AV+ cells (orange) in layer 1 drives the activity bump in layer 2 to drift clockwise, whereas knocking out AV- cells (blue) in layer 1 drives the activity bump in layer 2 to drift counter-clockwise. The red dashed line marks the time of knockout. (E) A schematic of alternatively inhibiting either the AV+ or AV- population across successive theta cycles. At later times in each theta cycle, cortical feedback is introduced to pin the activity bump back to the movement direction. (F) Interleaved inhibition of AV+ and AV- cell populations in layer 1 and the resulting directional sweeps in layer 2 across successive theta cycles. (G) Left-right sweeps in individual theta cycles (from blue to cyan) in layer 4, decoded from the population activity of layer 4 cells. (H) Sweeps in even (blue) and odd (orange) theta cycles after recentering the starting point of each sweep chunk to the origin and reorienting the movement direction to the north, with averaged left and right sweeps plotted as thicker lines.

The unit-number-matched comparison showed that the fully connected multi-module RNN reproduced the principal layer-specific trends observed in the HRNN: stronger directional tuning in layer 2 than in layer 1, a higher proportion of conjunctive grid cells in layer 3 than in layer 4, and a higher proportion of pure grid cells in layer 4 than in layer 3 (Table 4). The unstructured HRNN exhibited some of these trends, but did not show a significant layer-wise difference in conjunctive grid-cell fraction. Thus, the layer-specific organisation of neural responses does not require strictly feedforward hierarchy. Instead, these results support modular input–output organisation as an important contributor to the distribution of directional and spatial representations across populations.

We next examined whether the similarities in layer-specific cell properties extended to population dynamics and connectivity motifs. Across ten independent initialisations, the fully connected multi-module RNN had lower layer-1 ring and asymmetric-connectivity scores than the HRNN, despite having identical population sizes and the same input–output arrangement (Table 4). The unstructured HRNN also had lower scores than the HRNN. However, the contribution of hierarchy should not be interpreted as a general requirement for ring-attractor-like dynamics or path integration. The flat RNN also developed ring-like dynamics and asymmetric connectivity, and perturbing inter-layer connections impaired positional path integration in all three multi-layer architectures (Fig. S4E–F).

These comparisons suggest that modular organisation and hierarchical routing make complementary contributions: the former supports layer-specific representations, whereas their combination supports strong ring-like dynamics and organised connectivity asymmetry.

7 Discussion

Training RNNs on path-integration tasks has become an established approach for reproducing grid-like firing patterns in the entorhinal cortex [1; 8; 37; 21; 28; 7]; see Dorrell and Whittington [10] for a review. One limitation of this approach, however, is that the emergence of grid-like patterns can depend on specific modeling and training choices, including hyperparameter settings and the form of supervised place-cell targets [23]. Even so, these models remain valuable normative frameworks for understanding why grid-like representations can arise from the computational demands of path integration. Most previous models, however, supplied the network with two-dimensional velocity inputs, typically expressed as allocentric components v_x and v_y. While convenient computationally, such a representation is unlikely to be directly available to the brain, because it requires self-motion signals to be transformed into an external coordinate frame, for example relative to the walls of a square environment. In the brain, angular direction and linear speed are represented by distinct cell populations, including head direction cells [31; 36] and linear speed cells [15; 13], without any need for an explicit decomposition of velocity into two orthogonal allocentric components. Therefore, our HRNN separates directional and speed processing, which is more consistent with the biological organization of the navigation circuit.

Angular velocity (AV)-by-direction cells have been reported in the LMN, where they also show hemispheric lateralization [29]. Most direction cells recorded in the left LMN exhibit higher firing rates during counterclockwise (CCW) turns [29], resembling the AV+ cells in layer 1, whereas most direction cells recorded in the right LMN exhibit higher firing rates during clockwise (CW) turns, resembling the AV- cells. This raises the possibility that theta-paced interactions between the two hemispheres could lead to directional sweeps generated in the HRNN. Although the classical head direction circuit is not considered theta-modulated, direction cells in the anteroventral thalamic nucleus and parasubiculum do show theta modulation [32; 16; 35] (the population that encodes movement direction [36] and drives grid cells via conjunctive grid cells [35]). Such theta modulation may support alternating inhibition between lateralized directional populations, providing a possible biological analogue of the artificially imposed theta-paced oscillations between AV+ and AV- cells that generate theta sweeps in the HRNN. Testing this possibility will require future experiments.

Taken together, our results suggest that the HRNN could provide a first step toward digital twins of the brain's spatial navigation system. By linking task demands, neural representations, circuit motifs, and population dynamics in a single trainable model, the HRNN offers an integrative and predictive framework for understanding navigation circuits as dynamical systems. More broadly, we suggest that the next frontier is not simply the accumulation of more data, but the construction of mechanistic models with minimal hand-coded pre-specification that can generate testable predictions and guide targeted perturbations before costly in vivo experiments are performed.

Acknowledgments and Disclosure of Funding

Author contributions. Z.J. and H.Z. contributed equally to this work. K.G. contributed to model simulation. All authors discussed the results, and reviewed the manuscript. Z.J. and N.B. jointly supervised the work.

Funding. This work was supported by a Wellcome Principal Research Fellowship (222457/Z/21/Z) to N.B.

Competing interests. The authors declare no competing interests.

Code availability. The source code used to generate the results presented in this paper is publicly available at `https://github.com/ZilongJi/HRNN4PI.git`.

References

[1] 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.

[2] Joshua P Bassett and Jeffrey S Taube. Neural correlates for angular head velocity in the rat dorsal tegmental nucleus. *Journal of Neuroscience*, 21(15):5740–5751, 2001.

[3] Hugh T Blair, Jeiwon Cho, and Patricia E Sharp. Role of the lateral mammillary nucleus in the rat head direction circuit: a combined single unit recording and lesion study. *Neuron*, 21(6): 1387–1397, 1998.

[4] Charlotte N Boccara, Francesca Sargolini, Veslemøy Hult Thoresen, Trygve Solstad, Menno P Witter, Edvard I Moser, and May-Britt Moser. Grid cells in pre-and parasubiculum. *Nature neuroscience*, 13(8):987–994, 2010.

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

[6] Rishidev Chaudhuri, Berk Gerçek, Biraj Pandey, Adrien Peyrache, and Ila Fiete. The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep. *Nature neuroscience*, 22(9):1512–1520, 2019.

[7] Tianhao Chu, Yuling Wu, Neil Burgess, Zilong Ji, and Si Wu. Unfolding the black box of recurrent neural networks for path integration. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[8] 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.

[9] Christopher J Cueva, Peter Y Wang, Matthew Chin, and Xue-Xin Wei. Emergence of functional and structural properties of the head direction system by optimization of recurrent neural networks. *arXiv preprint arXiv:1912.10189*, 2019.

[10] William Dorrell and James CR Whittington. If grid cells are the answer, what is the question? a review of normative grid cell theory. *arXiv preprint arXiv:2601.12424*, 2026.

[11] 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.

[12] Kiah Hardcastle, Niru Maheswaranathan, Surya Ganguli, and Lisa M Giocomo. A multiplexed, heterogeneous, and adaptive code for navigation in medial entorhinal cortex. *Neuron*, 94(2): 375–387, 2017.

[13] James R Hinman, Mark P Brandon, Jason R Climer, G William Chapman, and Michael E Hasselmo. Multiple running speed signals in medial entorhinal cortex. *Neuron*, 91(3):666–679, 2016.

[14] Motosada Iwase, Takuma Kitanishi, and Kenji Mizuseki. Cell type, sub-region, and layer-specific speed representation in the hippocampal–entorhinal circuit. *Scientific reports*, 10(1): 1407, 2020.

[15] 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.

[16] E Lomi, KJ Jeffery, and AS Mitchell. Convergence of location, direction, and theta in the rat anteroventral thalamic nucleus. iscience 26, 106993, 2023.

[17] Valerio Mante, David Sussillo, Krishna V Shenoy, and William T Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *nature*, 503(7474):78–84, 2013.

[18] 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.

[19] Jonathan A Michaels, Stefan Schaffelhofer, Andres Agudelo-Toro, and Hansjörg Scherberger. A goal-driven modular neural network predicts parietofrontal neural dynamics during grasping. *Proceedings of the national academy of sciences*, 117(50):32124–32135, 2020.

[20] Aran Nayebi, Daniel Bear, Jonas Kubilius, Kohitij Kar, Surya Ganguli, David Sussillo, James J DiCarlo, and Daniel L Yamins. Task-driven convolutional recurrent models of the visual system. *Advances in neural information processing systems*, 31, 2018.

[21] Aran Nayebi, Alexander Attinger, Malcolm Campbell, Kiah Hardcastle, Isabel Low, Caitlin S Mallory, Gabriel Mel, Ben Sorscher, Alex H Williams, Surya Ganguli, et al. Explaining heterogeneity in medial entorhinal cortex with task-driven neural networks. *Advances in Neural Information Processing Systems*, 34:12167–12179, 2021.

[22] 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.

[23] Rylan Schaeffer, Mikail Khona, and Ila Fiete. No free lunch from deep learning in neuroscience: A case study through models of the entorhinal-hippocampal circuit. *Advances in neural information processing systems*, 35:16052–16067, 2022.

[24] Johannes D Seelig and Vivek Jayaraman. Neural dynamics for landmark orientation and angular path integration. *Nature*, 521(7551):186–191, 2015.

[25] Patricia E Sharp. Multiple spatial/behavioral correlates for cells in the rat postsubiculum: multiple regression analysis and comparison to other hippocampal areas. *Cerebral Cortex*, 6(2): 238–259, 1996.

[26] Patricia E Sharp, Hugh T Blair, and Michael Brown. Neural network modeling of the hippocampal formation spatial signals and their possible role in navigation: a modular approach. *Hippocampus*, 6(6):720–734, 1996.

[27] William Skaggs, James Knierim, Hemant Kudrimoti, and Bruce McNaughton. A model of the neural basis of the rat's sense of direction. *Advances in neural information processing systems*, 7, 1994.

[28] 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.

[29] Robert W Stackman and Jeffrey S Taube. Firing properties of rat lateral mammillary single units: head direction, head pitch, and angular head velocity. *Journal of Neuroscience*, 18(21): 9020–9037, 1998.

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

[31] 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.

[32] Marian Tsanov, Ehsan Chah, Seralynne D Vann, Richard B Reilly, Jonathan T Erichsen, John P Aggleton, and Shane M O'Mara. Theta-modulated head direction cells in the rat anterior thalamus. *Journal of Neuroscience*, 31(26):9489–9502, 2011.

[33] Daniel Turner-Evans, Stephanie Wegener, Hervé Rouault, Romain Franconville, Tanya Wolff, Johannes D Seelig, Shaul Druckmann, and Vivek Jayaraman. Angular velocity integration in a fly heading circuit. *Elife*, 6:e23496, 2017.

[34] Benigno Uria, Borja Ibarz, Andrea Banino, Vinicius Zambaldi, Dharshan Kumaran, Demis Hassabis, Caswell Barry, and Charles Blundell. The spatial memory pipeline: a model of egocentric to allocentric understanding in mammalian brains. *BioRxiv*, pages 2020–11, 2020.

[35] Abraham Z Vollan, Richard J Gardner, May-Britt Moser, and Edvard I Moser. Left–right-alternating theta sweeps in entorhinal–hippocampal maps of space. *Nature*, 639(8056):995–1005, 2025.

[36] Abraham Z Vollan, Michael F Schellenberger, May-Britt Moser, and Edvard I Moser. Attention-like regulation of theta sweeps in the brain's spatial navigation circuit. *bioRxiv*, pages 2026–01, 2026.

[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.

[38] Shawn S Winter, Benjamin J Clark, and Jeffrey S Taube. Disruption of the head direction cell network impairs the parahippocampal grid cell signal. *Science*, 347(6224):870–874, 2015.

[39] Guangyu Robert Yang and Manuel Molano-Mazón. Towards the next generation of recurrent network models for cognitive neuroscience. *Current opinion in neurobiology*, 70:182–192, 2021.

[40] Guangyu Robert Yang, Madhura R Joglekar, H Francis Song, William T Newsome, and Xiao-Jing Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature neuroscience*, 22(2):297–306, 2019.

[41] Kechen Zhang. Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. *Journal of Neuroscience*, 16(6):2112–2126, 1996.

A Parameter settings

Table 1: HRNN parameter settings

Model Architecture	
n_1	256
n_2	256
n_3	512
n_4	512
$\mathbf{I}^{(1),t}$	$[\omega_t]$, (1×1)
$\mathbf{I}^{(2),t}$	$\mathbf{r}^{(1),t}$, (256×1)
$\mathbf{I}^{(3),t}$	$[(\mathbf{r}^{(2),t})^\top, s_t]^\top$, (257×1)
$\mathbf{I}^{(4),t}$	$\mathbf{r}^{(3),t}$, (512×1)
Readout Neurons	
n_d	128
n_p	512
σ_d	30°
Place Cell Tuning Pattern	DoS
σ_p^1, σ_p^2	$0.12\,\mathrm{m}$, $0.24\,\mathrm{m}$
Task	
Arena size	$2.2\,\mathrm{m} \times 2.2\,\mathrm{m}$
Angular velocity distribution, ω_t	$\omega_t \sim \mathcal{N}(\mu = 0, \sigma = 5.76 \times 2)$
Linear speed distribution, s_t	$s_t \sim \mathrm{Rayleigh}(b = 0.13 \times 2\pi)$
Simulation time step	$dt = 0.02\,\mathrm{s}$
Training	
Path length	20
Batch size	500
Number of epochs	201
Number of batches	1000
Optimizer	Adam
Learning rate	10^{-4}
Regularisation	
L2 regularisation for layers 1 and 2	10^{-4}
L2 regularisation for layers 3 and 4	0

The HRNN was trained with simulated trajectories generated in a two-dimensional arena. All experiments were performed on a single RTX 3080 GPU. At each time step, angular velocity and linear speed were sampled from predefined distributions (Table 1). Each HRNN layer has n_ℓ hidden units and receives $\mathbf{I}^{(\ell),t}$ from the previous layer and/or from external inputs (angular velocity and linear speed). L2 regularisation was applied to the recurrent weights of hidden units in layer 1 and 2. The model architecture, trajectory parameters, readout-neuron parameters, training hyperparameters, and regularisation terms are summarized in Table 1.

The direction readout cells were defined by randomly tiling n_d direction-cell centres across circular angular space from $-\pi$ to π. For the head direction θ_t at time t, the circular distance between θ_t and the preferred direction of cell i, u_i, was computed as

$$\Delta_i(\theta_t) = \min\left(|\theta_t - u_i|,\ 2\pi - |\theta_t - u_i|\right). \tag{5}$$

Direction-cell activity was generated using softmax-normalised Gaussian tuning curves,

$$d_i(\theta_t) = \frac{\exp\left[-\Delta_i^2(\theta_t)/(2\sigma_d^2)\right]}{\sum_{j=1}^{n_d} \exp\left[-\Delta_j^2(\theta_t)/(2\sigma_d^2)\right]}. \tag{6}$$

Place cells (readout) were defined by randomly tiling n_p place-cell centres across the 2.2×2.2 meters two-dimensional environment. For a position $\mathbf{x}_t$ at time t, the squared distance between $\mathbf{x}_t$ and the

centre of place cell i, $\mathbf{u}_i$, was computed as

$$d_i^2(\mathbf{x}_t) = \|\mathbf{x}_t - \mathbf{u}_i\|^2. \tag{7}$$

Place-cell activity was generated using a difference-of-Softmax tuning pattern [28]. First, two softmax-normalised Gaussian population activities were computed as

$$g_i^1(\mathbf{x}_t) = \frac{\exp\left[-d_i^2(\mathbf{x}_t)/(2(\sigma_p^1)^2)\right]}{\sum_{j=1}^{n_p} \exp\left[-d_j^2(\mathbf{x}_t)/(2(\sigma_p^1)^2)\right]}, \qquad g_i^2(\mathbf{x}_t) = \frac{\exp\left[-d_i^2(\mathbf{x}_t)/(2(\sigma_p^2)^2)\right]}{\sum_{j=1}^{n_p} \exp\left[-d_j^2(\mathbf{x}_t)/(2(\sigma_p^2)^2)\right]}. \tag{8}$$

Then, the final place-cell activity for encoding position x_t was obtained from the difference

$$\tilde{p}_i(\mathbf{x}_t) = g_i^1(\mathbf{x}_t) - g_i^2(\mathbf{x}_t), \tag{9}$$

followed by shifting the population activity to be non-negative and normalising it to sum to one across place cells.

B Parameter sweep for HRNN

To assess whether the reported layer-specific representations and circuit motifs depend on a particular choice of model hyperparameters, we trained 45 HRNN configurations, with training parameters (including learning rate, batch size) the same as reported in Table. 1. Each configuration used an independently sampled random seed. The sweep jointly covered recurrent population sizes, numbers and receptive-field widths of direction and place cells, environment size, and layer-specific recurrent weight decay (Table 2). Thus, the robustness analysis spans substantial variation in both model capacity, task statistics and training strategies, rather than reflecting a single trained network.

Despite this variation, all configurations learned the path-integration task (Fig. S1A). The resulting representations also preserved the cell types reported in the main text (Fig. S1B): AHV tuning was prevalent in Layer 1, directional tuning was prevalent in Layer 2, conjunctive grid cells were enriched in Layer 3 relative to Layer 4, and pure grid cells were enriched in Layer 4 relative to Layer 3. Importantly, the functional progression of cell types along the hierarchy was also preserved (Fig. S1C,D): paired comparisons across configurations confirmed a decrease in AHV tuning and an increase in directional tuning from Layer 1 to Layer 2, together with the predicted redistribution of conjunctive and pure grid tuning between Layers 3 and 4. This hierarchy of cell types closely resembles the sequence of functional representations reported in rodent navigation circuits [3; 22].

Finally, the circuit-level signatures were similarly robust. Layer 1 activity formed ring-like manifolds across configurations (Fig. S1E), and ring scores exceeded heading-shuffled controls (Fig. S1F). Moreover, preferred-direction-sorted recurrent connectivity consistently showed direction-dependent asymmetry within AV+ and AV− subpopulations (Fig. S1G), with asymmetry indices exceeding shuffled preferred-direction controls (Fig. S1H). Together, these results indicate that the hierarchical progression of cell types and the associated Layer 1 dynamical and connectivity motifs are robust across the tested hyperparameter range.

Table 2: Hyperparameter robustness sweep for the HRNN. We trained 45 configurations for 100 epochs each, using an independently sampled random seed for each configuration. We evaluated all 16 combinations of recurrent layer sizes, and varied input/output population size, tuning widths, environment size, and each layer-specific recurrent weight decay one at a time from the default configuration.

Hyperparameter	Values tested
$N_{\mathrm{LMN}}, N_{\mathrm{PaS}}$	$\{128, 256\}$
$N_{\mathrm{Conj}}, N_{\mathrm{mEC}}$	$\{256, 512\}$
Number of direction cells, N_d	$\{64, 128, 256\}$
Number of place cells, N_p	$\{256, 384, 512, 640\}$
Direction-cell RF width	$\{10^\circ, 20^\circ, 30^\circ, 40^\circ, 50^\circ\}$
Place-cell RF width	$\{0.09, 0.12, 0.15, 0.18\}$
Environment width and height (m)	$\{1.5, 1.8, 2.0, 2.2, 2.4, 2.6\}$
Recurrent weight decay, each layer	$\{0, 10^{-5}, 10^{-4}, 10^{-3}\}$

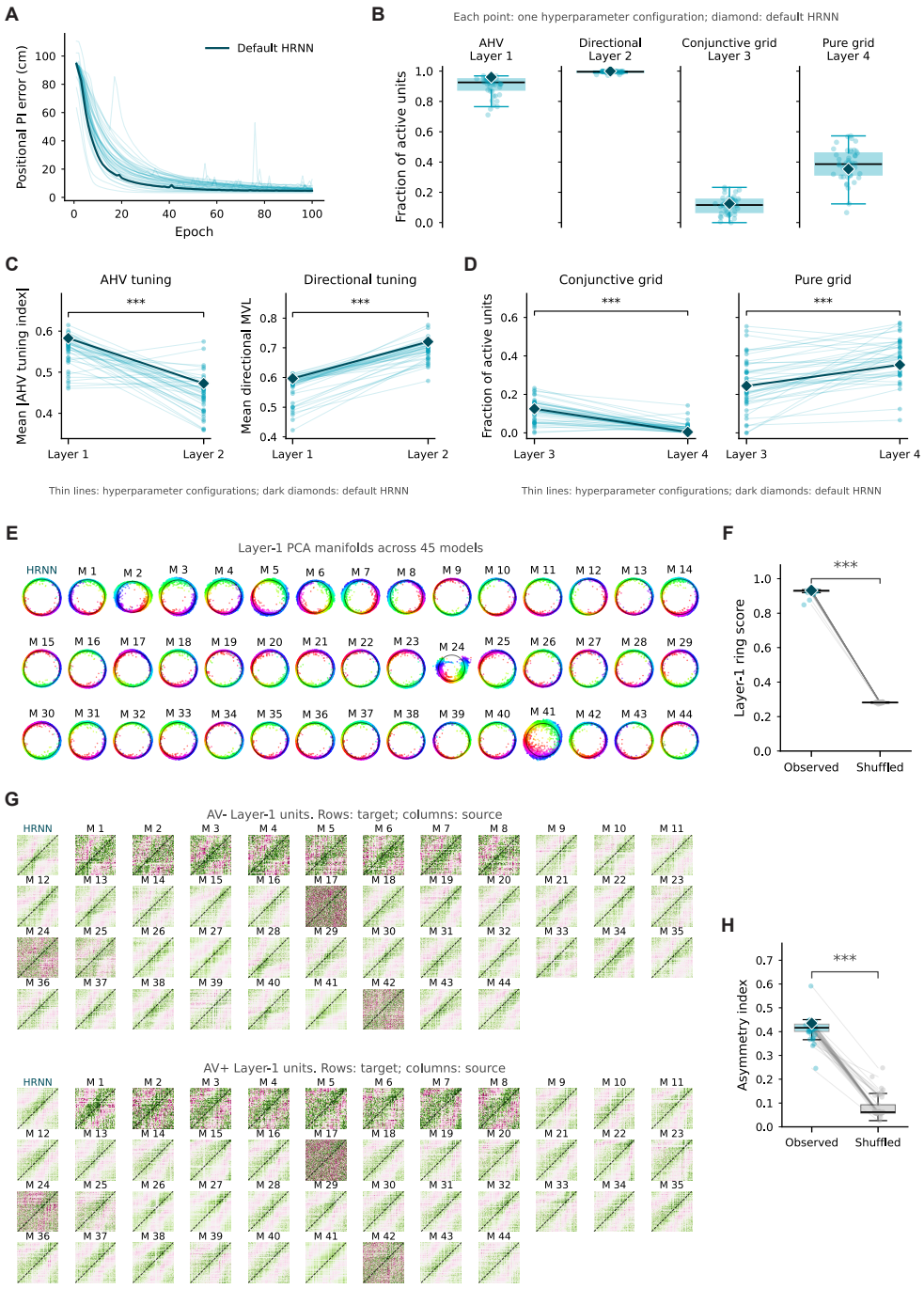

Figure S1: **Robustness of HRNN dynamics, cell types, and connectivity across hyperparameter configurations. A,** Path-integration error during training for 45 HRNN configurations; thin lines denote individual configurations and the dark line denotes the default HRNN. **B,** Fractions of active units exhibiting angular-head-velocity (AHV) tuning in Layer 1, directional tuning in Layer 2, conjunctive grid tuning in Layer 3, and pure grid tuning in Layer 4. **C–D,** Layer-specific tuning transformations across configurations: AHV tuning decreases from Layer 1 to Layer 2, directional tuning increases from Layer 1 to Layer 2, conjunctive grid tuning is enriched in Layer 3 compared to layer 4, and pure grid tuning is enriched in Layer 4 compared to Layer 3. **E,** Two-dimensional PCA projections of Layer 1 activity, reveal ring-like manifolds across configurations. **F,** Layer 1 ring scores exceed heading-shuffled controls. **G,** Recurrent connectivity matrices for AV$-$ and AV$+$ Layer 1 units, show reproducible direction-dependent asymmetric structure. **H,** Preferred-direction asymmetry indices exceed shuffled preferred-direction controls. Diamonds indicate the default HRNN; paired lines link matched observed and shuffled values. $***p < 0.001$, paired one-sided Wilcoxon test across hyperparameter configurations.

C Model comparison

The HRNN is biologically constrained. In the rodent brain, angular velocity is processed by the dorsal tegmental nucleus (DTN) and lateral mammillary nuclei (LMN), with directional information read out in the dorsal presubiculum. Linear speed is processed by speed cells in the entorhinal cortex, likely in medial entorhinal cortex (MEC) layer 3, and conveyed to layer 2, where it serves as input to grid cells. Our HRNN was inspired by this hierarchical routing of information.

For computational reasons, we tested whether the proposed HRNN is necessary for efficient path integration and for the emergence of structured neural representations and anatomically interpretable structures.We compared the HRNN with three control architectures trained on identical angular-velocity and linear-speed inputs and the same direction and position targets.

Together, we considered four recurrent architectures with trainable parameters approximately matched (Fig. S2A-D, Table. 3):

1. **Flat RNN.** A single recurrent population receives both angular velocity and linear speed, and jointly predicts direction and position.
2. **Unstructured HRNN.** Four recurrent layers are connected serially, but both angular-velocity and linear-speed inputs are fed into the first layer, and both direction and location are read out from the fourth layer.
3. **Multi-module RNN.** Four recurrent populations receive modality-specific inputs and outputs, but all populations are recurrently connected to one another.
4. **HRNN.** The proposed biologically constrained hierarchy: angular-velocity information is routed through the first two modules, linear-speed and place-related information enter downstream, and outputs are read out at the corresponding stages.

Table 3: Comparison of the four parameter-matched recurrent architectures. N/A indicates that a layer-wise comparison is not applicable.

Measure	Flat RNN	Unstructured HRNN	Multi-module RNN	HRNN
Parameter count	2.00M	2.43M	1.99M	2.00M
Trainng convergence speed	Slow	Medium	Medium	Fast
Fraction of AV cells	99.7%	99.0%	99.7%	99.6%
Fraction of direction cells	61.1%	51.9%	37.6%	24.1%
Fraction of conjunctive grid cells	4.5%	2.5%	1.0%	8.9%
Fraction of pure grid cells	9.5%	4.1%	3.3%	40.7%
More AV cells in Layer 1 than Layer 2	N/A	No	Yes	Yes
Weaker direction tuning in Layer 1 than Layer 2	N/A	No	Yes	Yes
Significant amount of conjunctive grid cells in Layer 3	N/A	No	No	Yes
Significant amount of pure grid cells in Layer 4	N/A	Yes	Yes	Yes
Ring attractor dynamics	Yes	No	No	Yes
Asymmetric connections	Yes	No	Yes	Yes
Perturbation affects path integration	N/A	Yes	Yes	Yes

C.1 HRNN accelerates path-integration learning compared to control architectures

We first compared training dynamics across ten random initializations per architecture. Although all models ultimately reached low positional path-integration error, the HRNN converged substantially faster and with lower between-seed variability than the flat, serial-unstructured, and fully connected multi-module controls (Fig. S2E).

We quantified learning efficiency as the first epoch at which positional error fell below 10 cm. The HRNN reached this criterion consistently within the first few epochs, whereas the other architectures required substantially more training epochs (Fig. S2F). These results show that the advantage is not explained by depth alone or by merely partitioning units into multiple populations; the constrained routing of the HRNN improves optimization efficiency. One potential explanation is that the HRNN first learns directional information in its first two layers (see Fig. 1C). Accurate directional information is then available to layers 3 and 4, allowing them to integrate speed information to estimate position.

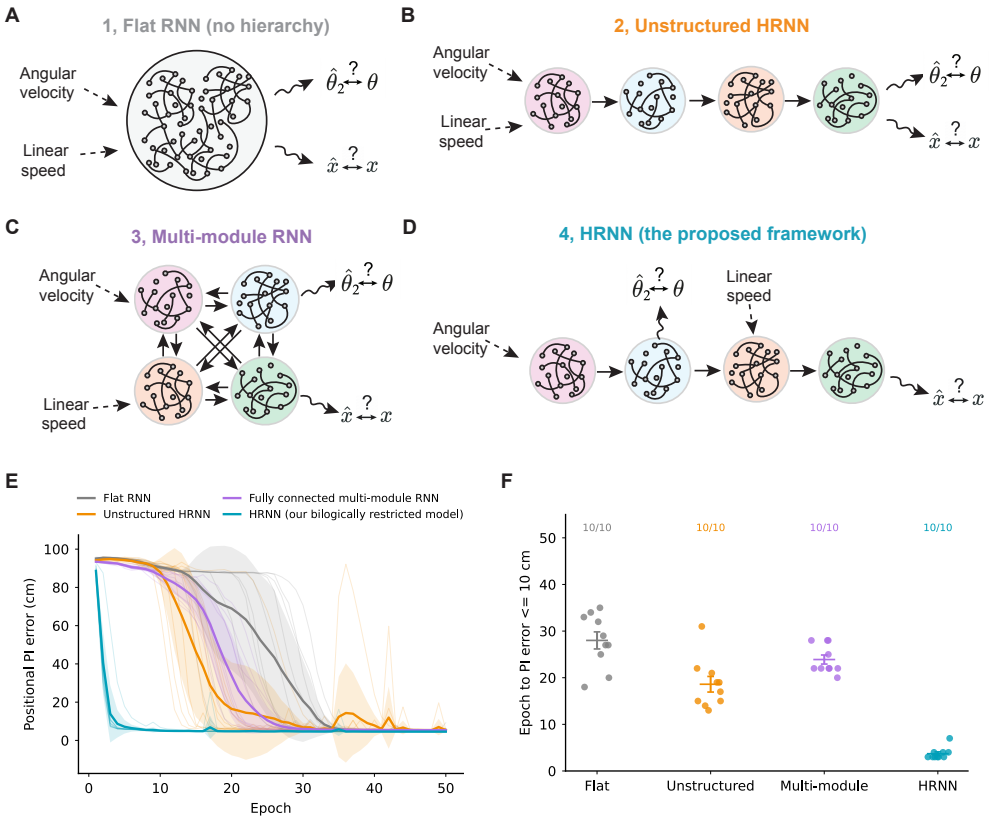

Figure S2: Model comparison reveals that the HRNN accelerates path-integration learning. (A) Flat RNN. (B) Unstructured HRNN. (C) Fully connected multi-module RNN. (D) HRNN. (E) Positional path-integration error across training epochs for ten random initializations of each architecture; faint traces show individual initializations, thick traces show the mean, and shaded regions show variability across initializations. (F) Epoch at which positional path-integration error first reaches 10 cm for each initialization; labels indicate the number of runs (out of ten) reaching this criterion.

C.2 HRNN promotes layer-specific cell types

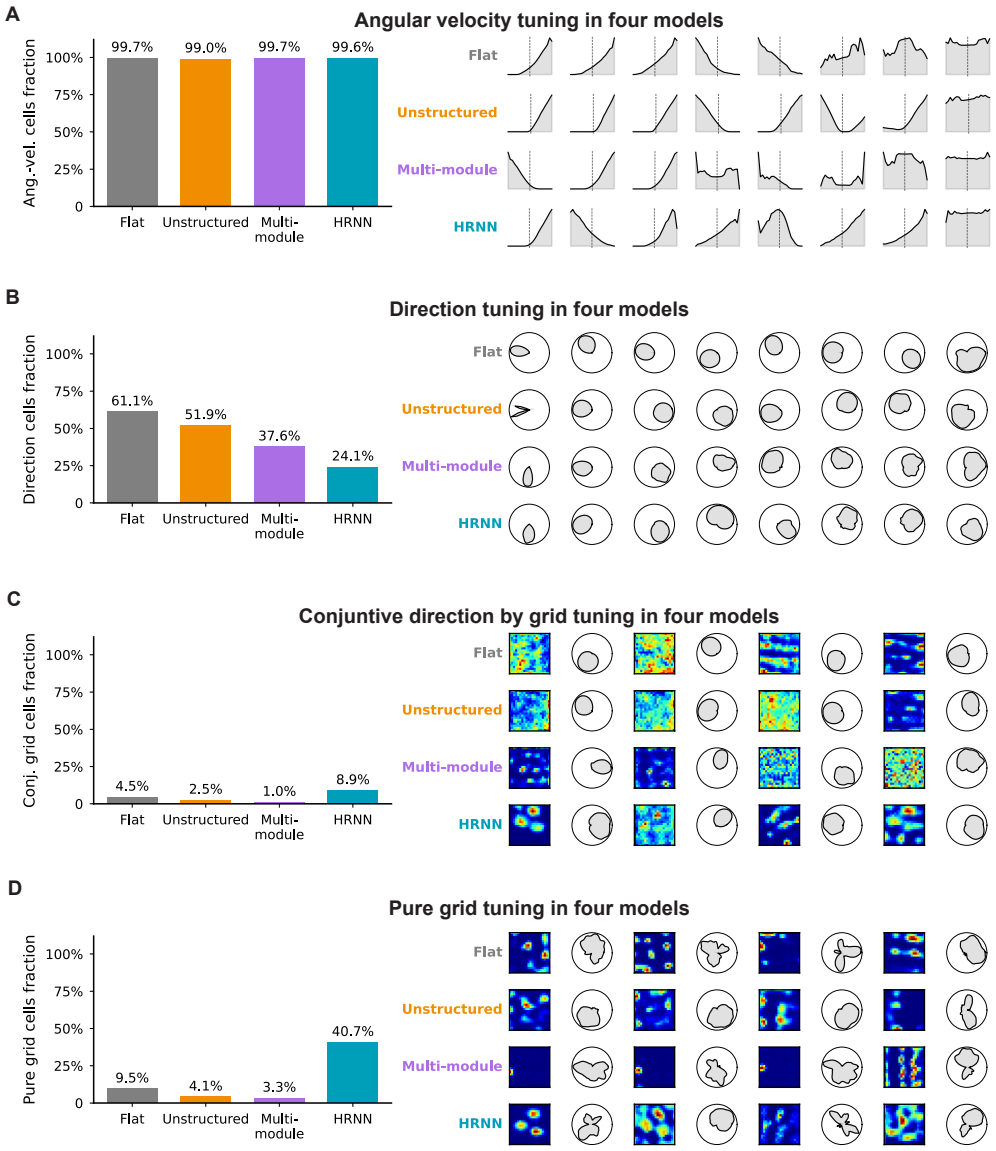

Figure S3: All four recurrent architectures generates biological realistc cell types after training. (A) Fractions of angular-velocity-tuned cells and representative angular-velocity tuning curves for the flat RNN, unstructured HRNN, fully connected multi-module RNN, and HRNN. (B) Fractions of direction-tuned cells and representative directional tuning maps. (C) Fractions of conjunctive direction-by-grid cells and representative spatial firing-rate maps paired with directional tuning maps. (D) Fractions of pure grid cells and representative spatial firing-rate maps paired with directional tuning maps. Bars report the cell fraction in each architecture; colors identify architectures as in Fig. S2.

We next examined whether each architecture developed the cell types expected from its computational organization (Fig. S3). All architectures contained angular-velocity-modulated and directionally tuned units, consistent with the shared path-integration objective. However, their distributions across modules differed markedly. Importantly, only HRNN showed reduced angular velocity tuning and increased diurectional tuning along the hierarchy (Fig. S4A&B), as reported in the biological brain [3].

In the HRNN, early modules were strongly dominated by angular-velocity and directional tuning, whereas downstream modules exhibited conjunctive grid-like and pure grid-like spatial representations. In particular, conjunctive grid cells were enriched in the intermediate spatial module, while pure grid cells were substantially more prevalent in the final module (Fig. S4C&D), as reflected in the bilogical brain [22]. This layer-specific progression was weaker or absent in the flat, unstructured, and fully connected multi-module controls.

These results indicate that task training alone can generate individual tuning motifs, whereas the HRNN promotes their reliable segregation into distinct functional stages, mirroring the organization observed in the biological brain.

C.3 Only the Flat RNN and the HRNN develop ring attractors in the population activity

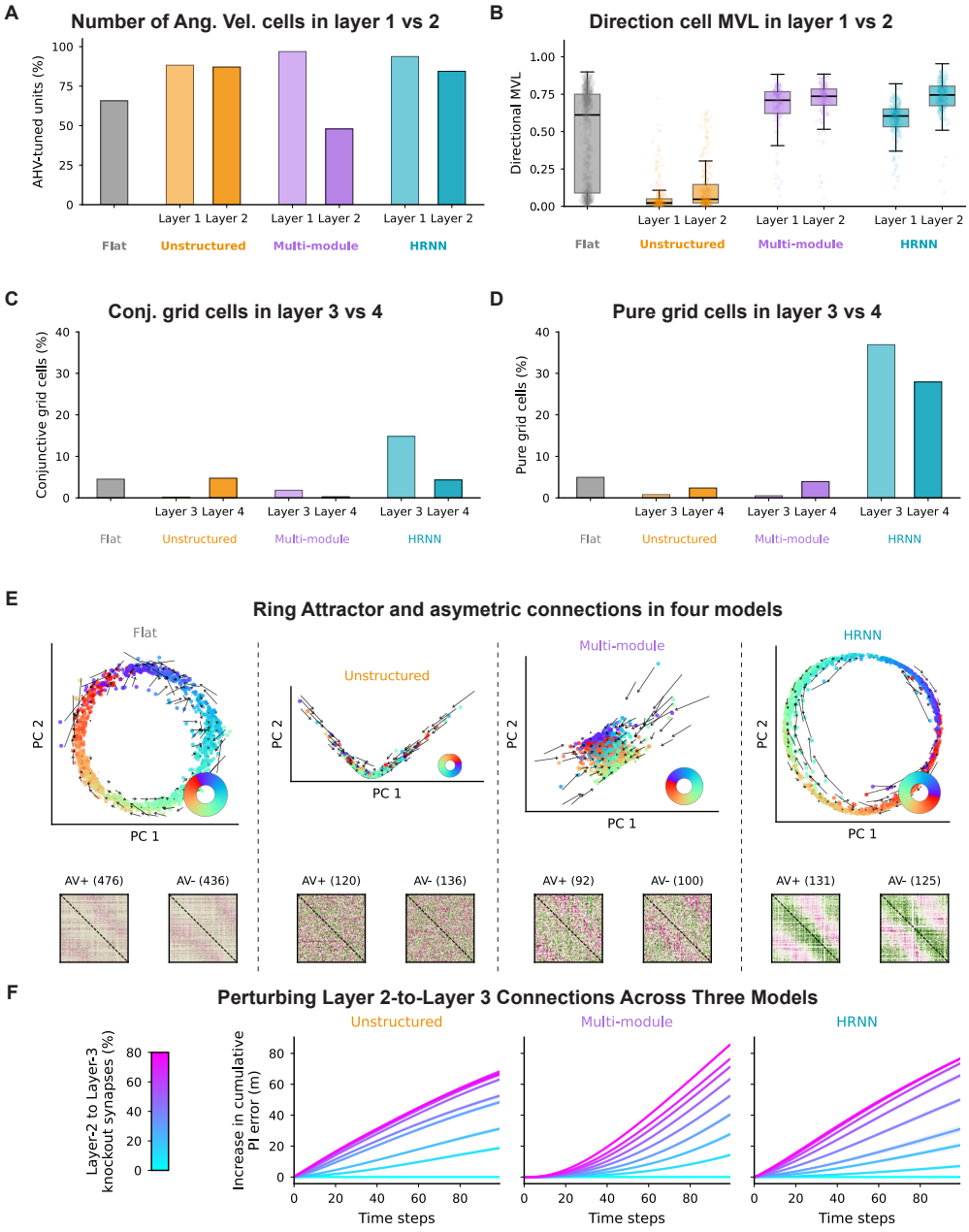

Figure S4: Among the four parameter-matched recurrent architectures, only the HRNN reliably segregates cell types into distinct functional stages, mirroring the organization observed in the biological brain. (A) Proportions of angular-velocity-tuned cells across architectures. (B) Directional tuning strength (mean vector length; MVL) in layers 1 and 2. (C–D) Proportions of conjunctive and pure grid cells in layers 3 and 4. (E) Low-dimensional population dynamics and asymmetric recurrent connectivity between angular-velocity-positive and -negative units. (F) Random knockout of Layer 2-to-Layer 3 synapses increases cumulative positional path-integration error in the unstructured, multi-module, and hierarchical RNNs. Colors indicate the fraction of disrupted synapses.

We then analyzed the geometry and recurrent dynamics of the first recurrent module (Fig. S4). The Flat RNN and HRNN exhibited a circular low-dimensional manifold whose neural trajectories evolved

coherently with head direction, consistent with a ring-attractor-like representation. By contrast, the serial-unstructured HRNN and fully connected multi-module RNN showed less organized low-dimensional geometry and less coherent directional flow. The Flat RNN, fully connected multi-module RNN and HRNN showed asymmetric connections, mimicking shifter cells in the brain ((Fig. S4)E). Finally, randomly silencing Layer 2-to-Layer 3 synapses increased cumulative positional path-integration error in all three multi-layer models (Fig. S4F).

Together with the layer-specific tuning results, these findings suggest that the HRNN's constrained routing does more than accelerate learning: it biases the network toward anatomically interpretable connectivity motifs and a sequential transformation from angular-velocity signals to directional, conjunctive, and spatial representations.

C.4 Unit-matched model comparison

In addition to the parameter-matched model comparison, we also conducted a unit-matched model comparison. The HRNN was compared with the unit-matched unstructured HRNN and the multi-module RNN. All three RNNs contained identical numbers in the four layers (256, 256, 512, and 512). The flat-RNN was excluded for the model comparison because a layer-specific comparison is not applicable to a single-layer architecture.

In summary, the HRNN, with brain-inspired inductive biases, reliably reproduces layer-specific distributions along the navigation hierarchy as well as the relevant ring-like dynamics and asymmetric connectivity motifs. The unit-number-matched multi-module RNN also reproduces aspects of the layer-specific cell-type redistribution, but across 10 independently initialized models it does not reliably reproduce the strong ring-like and asymmetric-connectivity motifs observed in the HRNN. The unit-matched unstructured RNN does not reliably reproduce all layer-specific distributions. These comparisons clarify the contribution of the hierarchical routing and modular input-output structures (Table 4).

Table 4: Comparison of neural representations and ring dynamics across the three recurrent architectures. Values are presented as mean ± SEM. Statistical significance was assessed using FDR-corrected p-values. n.s., not significant; * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$.

Metric	Statistic	Unstructured HRNN	Multi-module RNN	HRNN
AHV-tuned units	Layer 1	$91.8 \pm 0.8\%$	$100.0 \pm 0.0\%$	$99.0 \pm 0.2\%$
	Layer 2	$88.0 \pm 2.3\%$	$84.4 \pm 0.8\%$	$90.2 \pm 0.9\%$
	Difference	L1 > L2	L1 > L2	L1 > L2
	p_{FDR}	0.152 (n.s.)	1.05×10^{-7} (***)	2.39×10^{-5} (***)
Directional MVL	Layer 1	0.388 ± 0.077	0.599 ± 0.003	$\mathbf{0.621 \pm 0.002}$
	Layer 2	0.480 ± 0.072	0.680 ± 0.009	$\mathbf{0.745 \pm 0.004}$
	Difference	L2 > L1	L2 > L1	L2 > L1
	p_{FDR}	0.0283 (*)	2.39×10^{-5} (***)	1.29×10^{-8} (***)
Conjunctive grid cells	Layer 3	$6.7 \pm 1.0\%$	$9.3 \pm 0.6\%$	$\mathbf{16.8 \pm 1.1\%}$
	Layer 4	$4.8 \pm 0.4\%$	$0.5 \pm 0.1\%$	$2.1 \pm 0.6\%$
	Difference	L3 > L4	L3 > L4	L3 > L4
	p_{FDR}	0.152 (n.s.)	2.66×10^{-7} (***)	6.61×10^{-6} (***)
Pure grid cells	Layer 3	$1.0 \pm 0.4\%$	$8.2 \pm 0.5\%$	$23.0 \pm 3.5\%$
	Layer 4	$6.2 \pm 1.4\%$	$13.5 \pm 1.2\%$	$\mathbf{41.8 \pm 2.7\%}$
	Difference	L4 > L3	L4 > L3	L4 > L3
	p_{FDR}	0.00282 (**)	0.00192 (**)	2.59×10^{-4} (***)
Layer-1 ring score		0.468 ± 0.085	0.215 ± 0.011	$\mathbf{0.850 \pm 0.002}$
Layer-1 asymmetry score		0.266 ± 0.052	0.139 ± 0.015	$\mathbf{0.452 \pm 0.003}$

C.5 Model comparison of different readout layer setups and activation functions

As discussed in [28; 23], previous studies have shown that, in single-layer RNNs trained on spatial path integration tasks, the choice of readout configuration (e.g., low-dimensional Cartesian coordinates, Gaussian tuning, or DoS tuning) and activation function (e.g., tanh or ReLU) can substantially influence the emergence of grid cells in hidden units. To investigate how these architectural choices affect grid cell formation in the HRNN, we systematically varied either the readout configuration or the activation function while keeping all other model settings unchanged (Figs. S6–S7).

For the readout configuration, we first examined the effects of varying the number of readout units and their tuning widths (Fig. S5). We then evaluated two alternative readout configurations: (1) a low-dimensional coordinate readout $(\sin\theta, \cos\theta, x, y)$ and (2) Gaussian-tuned direction- and place-cell readout layers (Fig. S6). Finally, to assess the influence of the activation function, we compared the HRNN using tanh activation with the original model (Fig. S7).

In summary, the emergence of direction cells and ring attractor dynamics in Layers 1 and 2 was largely insensitive to variations in readout parameters and readout configurations (coordinate-based or DoS). Similarly, ring attractor dynamics emerged in both layers across all activation function configurations, with the exception of Layer 1 when using tanh activation. In contrast, the emergence of grid cells in Layers 3 and 4 was sensitive to both the number of place-cell readout units and their tuning widths. Moreover, grid cells emerged only when DoS readout and ReLU activation were used, consistent with previous findings [28; 23].

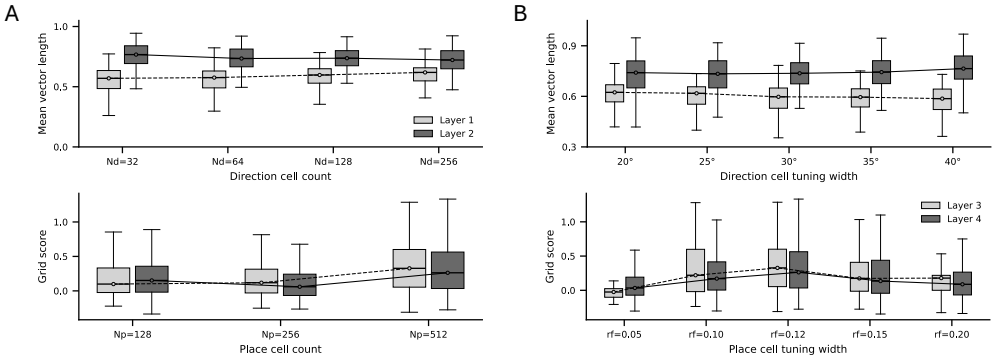

Figure S5: Sensitivity of cell-type properties to readout-neuron tuning parameters. (A) Upper: Mean vector length of directional tuning for different numbers of directional readout cells. Lower: Grid scores for different numbers of place readout cells. (B) Upper: Mean vector length of directional tuning for different tuning widths of directional readout cells. Lower: Grid scores for different tuning widths of place readout cells.

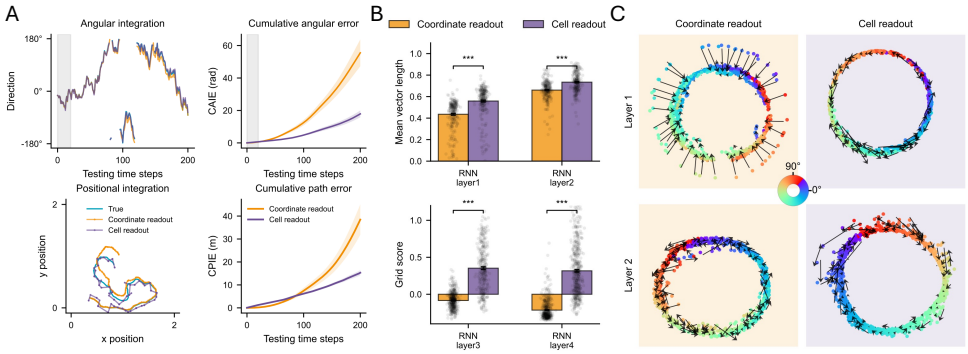

Figure S6: Performance of models with different readout setups. (A) Path-integration performance in models with (1) coordinate readout $(\sin\theta, \cos\theta, x, y)$ and (2) direction- and place-cell readout. (B) Upper: Mean vector length of directional tuning in layers 1 and 2 in the two models. Lower: Grid scores in layers 3 and 4 in the two models. (C) Ring-attractor manifolds in layer 1 (upper) and layer 2 (lower) in the two models.

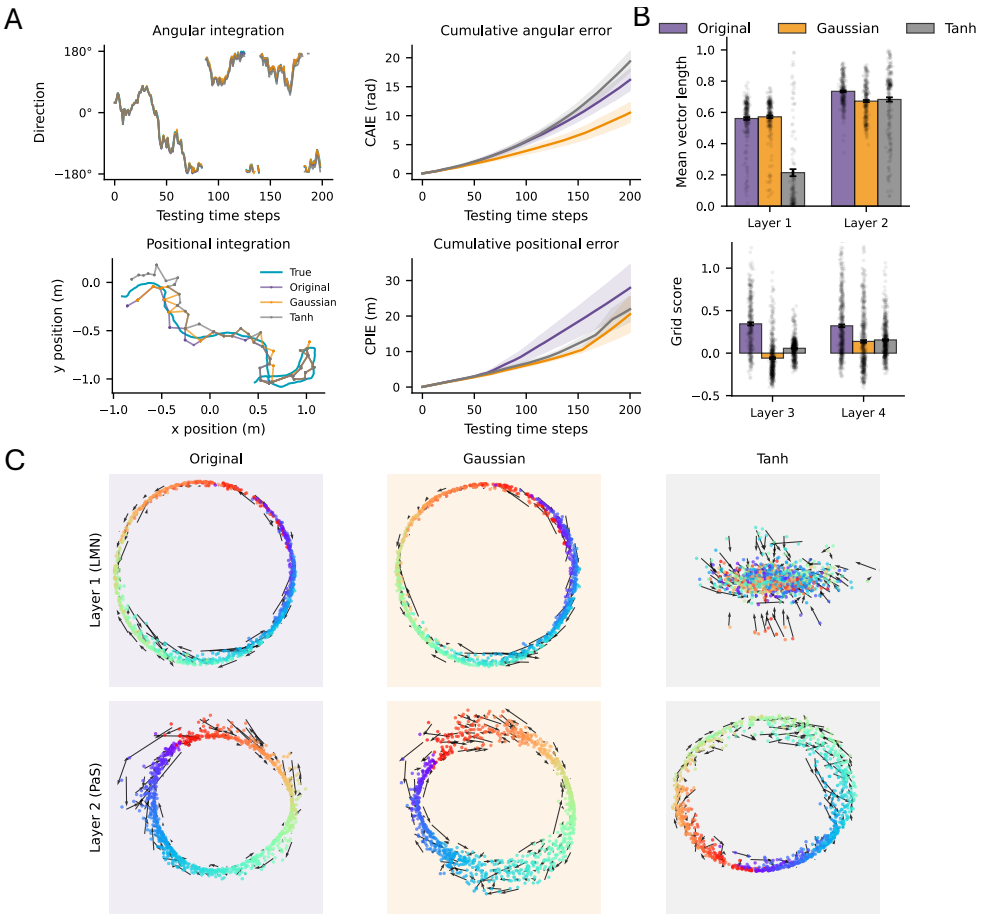

Figure S7: Performance of models with different place-cell readout patterns and activation functions. (A) Path-integration performance in models with (1) DoS readout and ReLU activation, (2) Gaussian readout and ReLU activation, and (3) DoS readout and tanh activation. (B) Upper: Mean vector length of directional tuning in layers 1 and 2 in the three models. Lower: Grid scores in layers 3 and 4 in the two models. (C) Ring-attractor manifolds in layer 1 (upper) and layer 2 (lower) in the three models.

D Criteria for identifying angular velocity cells

For each unit in layer 1 and 2, we first computed an AV tuning curve by binning signed angular velocity and averaging the unit activity within each bin. Let r_i be the mean activity in AV bin i, and let v_i be the corresponding AV bin center. Activity values were rectified as

$$\tilde{r}_i = \max(r_i, 0). \tag{10}$$

The AV tuning index was defined as a signed modulation depth:

$$\mathrm{TI}_{\mathrm{AV}} = \mathrm{sign}(\beta)\frac{\tilde{r}_{\max} - \tilde{r}_{\min}}{\tilde{r}_{\max} + \tilde{r}_{\min} + \epsilon}, \tag{11}$$

where

$$\tilde{r}_{\max} = \max_i \tilde{r}_i, \qquad \tilde{r}_{\min} = \min_i \tilde{r}_i, \tag{12}$$

and $\epsilon = 10^{-8}$. The sign was determined by the slope β of a linear fit to the AV tuning curve:

$$\tilde{r}_i = \alpha + \beta v_i + \eta_i. \tag{13}$$

To quantify how well the tuning curve was described by a linear relationship, we computed

$$R^2 = 1 - \frac{\sum_i (\tilde{r}_i - \hat{r}_i)^2}{\sum_i (\tilde{r}_i - \bar{r})^2 + \epsilon}, \tag{14}$$

where $\hat{r}_i = \alpha + \beta v_i$ is the fitted response and $\bar{r}$ is the mean of $\tilde{r}_i$ across AV bins.

A unit was classified as AV-tuned if

$$|\mathrm{TI}_{\mathrm{AV}}| > 0.2 \qquad \text{and} \qquad R^2 > 0.5. \tag{15}$$

E Calculating directional mean vector length

Directional tuning was quantified by calculating the mean vector length (MVL). Ground-truth angles were divided into 60 equally spaced bins between $-\pi$ and π, and the mean activity of each cell was calculated within each direction bin to obtain a direction-tuning curve. The tuning curves were circularly smoothed using a Gaussian filter with wrap-around boundary conditions.

Only non-silent units were included in the analysis, defined as cells whose peak activity exceeded 10^{-8}. Each direction-tuning curve was normalized by its total activity across direction bins before computing the MVL. For each cell, MVL was calculated as the magnitude of the activity-weighted circular mean vector:

$$\mathrm{MVL} = \sqrt{\left(\frac{\sum_b r_b \cos\theta_b}{\sum_b r_b}\right)^2 + \left(\frac{\sum_b r_b \sin\theta_b}{\sum_b r_b}\right)^2}, \tag{16}$$

where r_b denotes the normalised mean activity of the cell in direction bin b after setting negative values to zero, and θ_b is the centre of that direction bin. Larger MVL values indicate stronger directional modulation. Hidden units with MVL > 0.3 were classified as direction-tuned cells, unless otherwise stated.

F Calculating the grid score

Grid score was calculated following Banino et al. [1]. Only non-silent units with maximum activation exceeding 10^{-8} were included in the calculation.

The two-dimensional trajectory arena was first divided into 20×20 spatial bins. At each time step, the ground-truth position was assigned to the corresponding spatial bin, and activity was averaged across all visits to each bin to obtain a spatial rate map for each unit. For each non-silent cell, the spatial autocorrelogram (SAC) of its spatial rate map was computed. To quantify six-fold rotational symmetry, the SAC was rotated by $30°$, $45°$, $60°$, $90°$, $120°$, $135°$, and $150°$.

Correlations between the original and rotated SACs were calculated within annular masks, excluding the central region of the SAC. For each mask, the grid score compared correlations at grid-aligned rotations ($60°$ and $120°$) with those at non-grid-aligned rotations ($30°$, $90°$, and $150°$):

$$\text{Grid score}_{60} = \frac{C_{60°} + C_{120°}}{2} - \frac{C_{30°} + C_{90°} + C_{150°}}{3}, \tag{17}$$

where C_α denotes the correlation between the SAC and the SAC rotated by angle α within the annular mask. The final grid score of each unit was defined as the maximum Grid score_{60} across all tested annular masks. Larger values indicate stronger hexagonal grid-like spatial periodicity. Cells with grid score > 0.3 were classified as grid cells, unless otherwise stated. Among grid cells, cells with MVL > 0.3 were further classified as conjunctive grid cells, whereas those with HD MVL ≤ 0.3 were classified as pure grid cells.

G Calculating flux in ring attractor dynamics

To visualise the low-dimensional structure in layer 1 and 2, principal component analysis (PCA) was first applied to the population activity in layer 1 and layer 2, separately. The first two principal components were used to visualize the population activity in the low dimensional space.

To further assess whether the observed manifold exhibited attractor-like dynamics, flux analysis was then performed in the PCA-projected state space. This analysis follows the population-manifold approach used by Chaudhuri et al. [6], where local flow fields were examined to determine whether neural states away from the manifold showed biased dynamics back toward the manifold. Here, flux was estimated from temporal transitions in the low-dimensional PCA trajectories of the HRNN hidden states.

To estimate the local flow field, for each time step, the position in PCA subspace was defined as the current hidden state, and the displacement vector was defined as the difference between the next and current PCA states:

$$\Delta \mathbf{x}_t = \mathbf{x}_{t+1} - \mathbf{x}_t. \tag{18}$$

The two-dimensional PCA subspace was divided into 20×20 bins. Each transition was assigned to a bin according to its starting position $\mathbf{x}_t$, and the displacement vectors within each bin were averaged. The resulting mean displacement vector in each occupied bin was defined as the local flux vector:

$$\mathbf{F}_{ij} = \frac{1}{N_{ij}} \sum_{\mathbf{x}_t \in B_{ij}} (\mathbf{x}_{t+1} - \mathbf{x}_t), \tag{19}$$

where B_{ij} denotes the (i, j)-th bin in PCA space, and N_{ij} is the number of transitions assigned to that bin. Only bins containing at least one transition were included. The flux vectors therefore summarize the average direction and magnitude of local hidden-state dynamics in the PCA-projected state space.

H Calculating offset from layer 3 to layer 4

The spatial offset from layer 3 to layer 4 was quantified using a signed, weight-based centre-of-mass (CoM) analysis. For each cell in layer 3 and layer 4, a spatial CoM was first computed from its two-dimensional spatial rate map.

For the j_{th} direction cell in layer 3 (MVL > 0.5), its outgoing weights to the i_{th} cell in layer 4 were extracted: W_{ij}.

The signed weighted population CoM in layer 4, relative to cell j in layer 3, was then computed as:

$$\mathbf{c}_j^{\text{weighted}} = \mathbf{c}_j^{\text{Layer 3}} + \frac{\sum_i W_{ij} \left(\mathbf{c}_i^{\text{Layer 4}} - \mathbf{c}_j^{\text{Layer 3}} \right)}{\sum_i |W_{ij}|}, \tag{20}$$

where $\mathbf{c}_j^{\text{Layer 3}}$ is the spatial CoM of cell j in layer 3, $\mathbf{c}_i^{\text{Layer 4}}$ is the spatial CoM of cell i in layer 4. Positive and negative feedforward weights shift the weighted population CoM in opposite directions, while normalization by $\sum_i |W_{ij}|$ preserved the signed contribution of each projection.

The spatial offset from layer 3 to layer 4 was defined as the vector from the layer 3 CoM to the signed weighted layer 4 CoM:

$$\Delta \mathbf{c}_j = \mathbf{c}_j^{\text{weighted}} - \mathbf{c}_j^{\text{Layer 3}}. \tag{21}$$

I Tuning maps of all RNN units

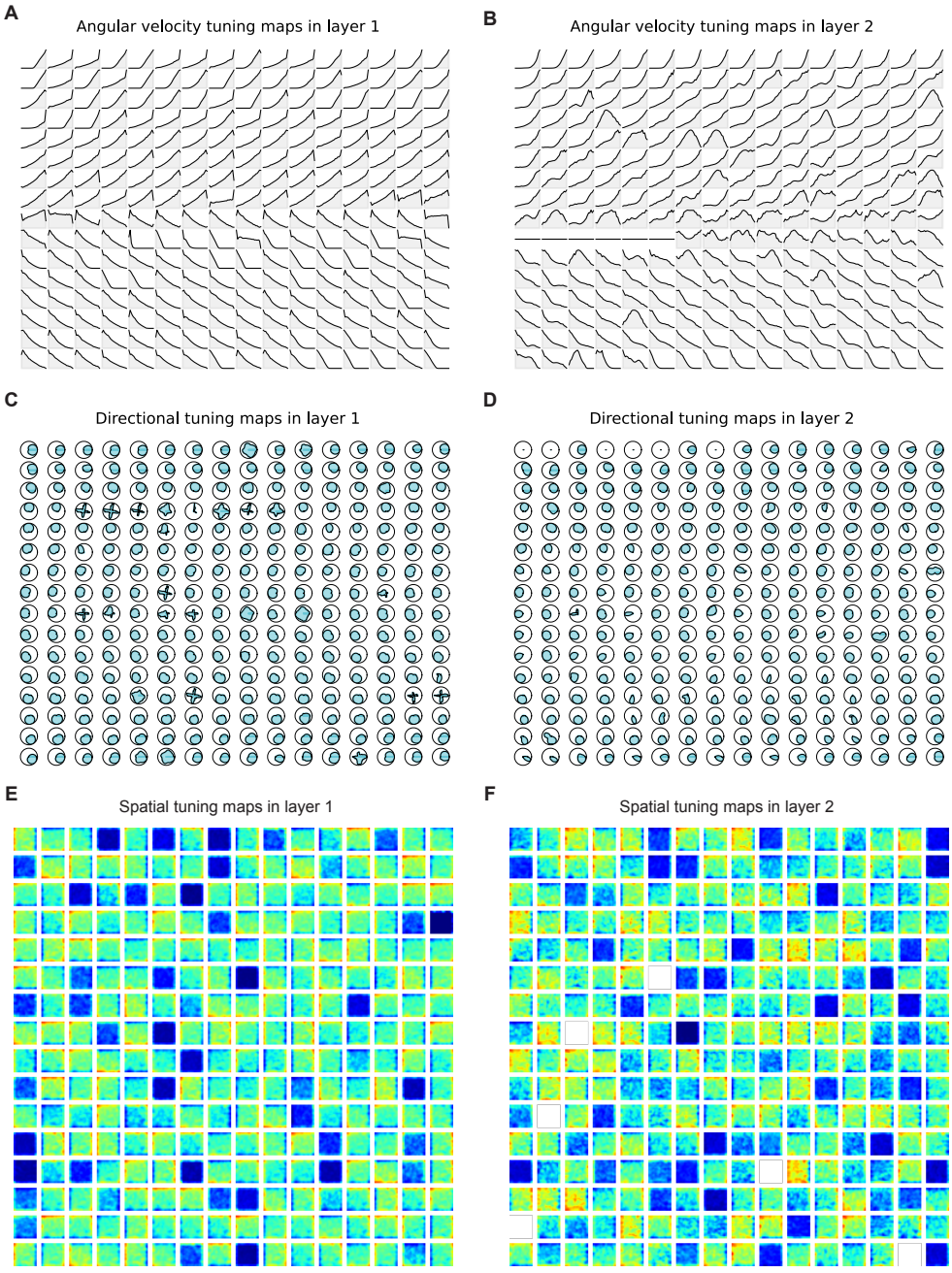

Figure S8: All tuning maps in RNN layers 1 and 2. (A-B) Angular velocity tuning maps in layer 1 and layer 2. (C-D) Directional tuning maps. (E-F) Spatial tuning maps. Note that there is no clear spatial tuning in the first two layers.

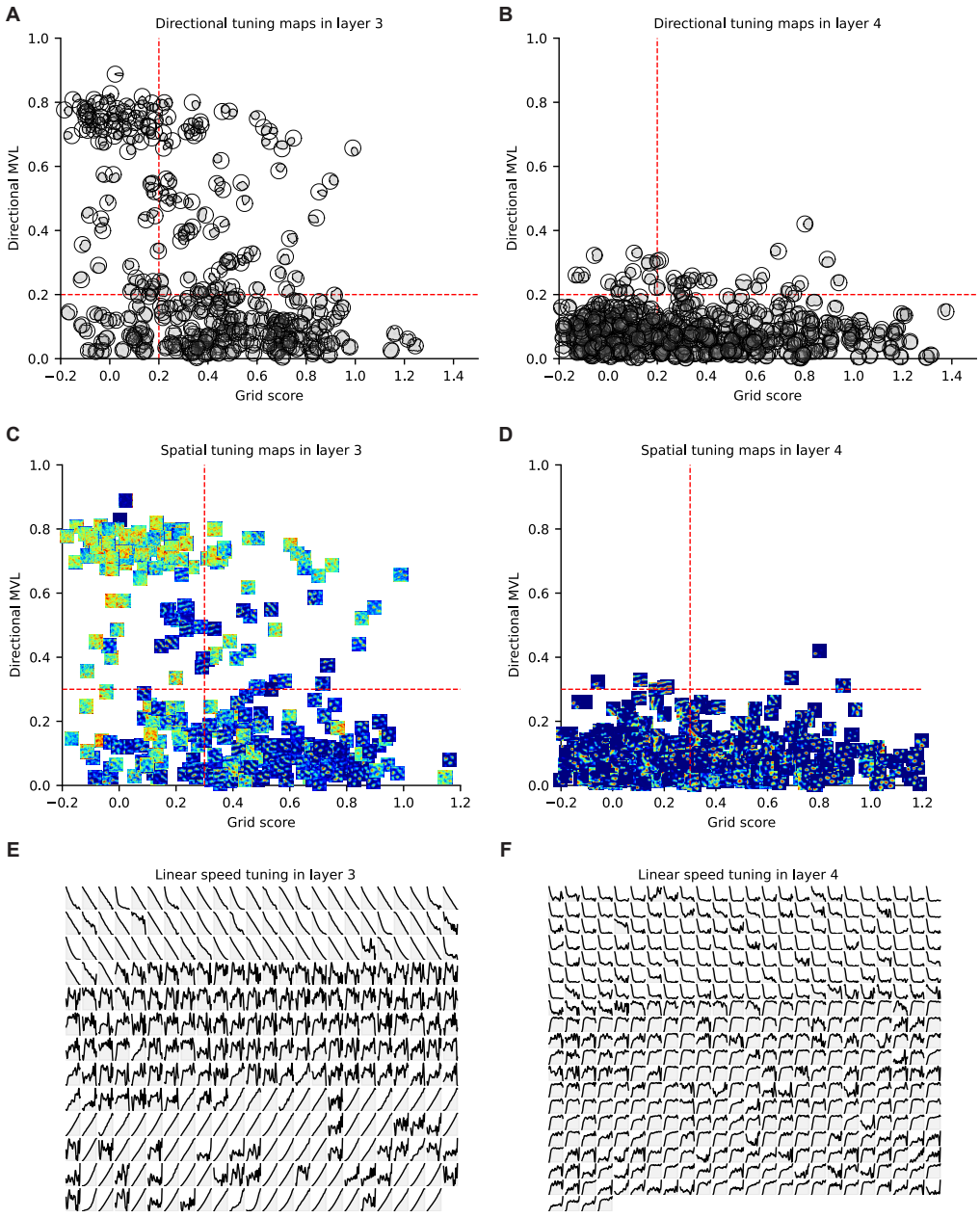

Figure S9: All tuning maps in RNN layer 3 and 4. (A-B) Directional tuning maps in layer 3 and 4, arranged according to each cell's directional mean vector length (y-axis) and grid score (x-axis). (C-D) Spatial tuning maps in layer 3 and 4, arranged in the same way. (E-F) Linear speed tuning maps in layer 3 and 4, arranged according to the slope of speed tuning from negative to positive. Note that linear speed tuning is more linear in layer 3 compared to layer 4.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

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 abstract and introduction accurately summarize the paper's main contributions, namely that the HRNN reproduces key cell types, anatomical motifs, and population dynamics of the spatial navigation system, while also making clear the modeling assumptions and scope.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: Limitations were discussed in the first two paragraphs of the Discussion.

Guidelines:

- The answer [N/A] 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: [N/A]

   Justification: The paper does not include theoretical results

   Guidelines:

   - The answer [N/A] 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: All results presented in the figures are fully reproducible from the information provided in the paper.

Guidelines:

- The answer [N/A] 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: The source code used to generate the results presented in this paper is publicly available at `https://github.com/ZilongJi/HRNN4PI.git`

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: All the details have been included in Appendix A.

   Guidelines:

   - The answer [N/A] 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: Confidence intervals in Figure 3D has been defined in main text.

   Guidelines:

   - The answer [N/A] 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: [Yes]

   Justification: Computer resources were provided in Appendix A.

   Guidelines:

   - The answer [N/A] 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 authors have reviewed the NeurIPS Code of Ethics and believe that the research conforms to it.

   Guidelines:

   - The answer [N/A] 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: [N/A]

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

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

Guidelines:

- The answer [N/A] 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: All external assets used in the paper are properly credited through citations, and their use complies with the applicable terms and conditions.

    Guidelines:

    - The answer [N/A] 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: [No]

    Justification: The paper introduces new code, which will be documented and released upon acceptance, but it is not included in the current submission.

    Guidelines:

    - The answer [N/A] 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: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] 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: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were not used as an important, original, or non-standard component of the core methods in this research.

Guidelines:

- The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.