A simple model of co-emergence of grid and place fields

Zhaoze Wang[1] zhaoze@seas.upenn.edu
Genela Morris [2,3] genelam@tlvmc.gov.il
Dori Derdikman [4] derdik@technion.ac.il
Pratik Chaudhari [1†] pratikac@seas.upenn.edu
Vijay Balasubramanian [5,6†] vijay@physics.upenn.edu

[1]Dept. of Electrical and Systems Engineering, University of Pennsylvania
[2]Tel Aviv Sourasky Medical Center
[3]Gray Faculty of Medical and Health Sciences, Tel Aviv University
[4]Rappaport Faculty of Medicine, Technion – Israel Institute of Technology
[5]Dept. of Physics, University of Pennsylvania [6]Santa Fe Institute
[†]**Equal contribution**

Abstract

Grid cells in the medial entorhinal cortex and place cells in the hippocampus together support spatial navigation. The two regions are reciprocally connected, and there is a chicken-and-egg problem for how both arise and reinforce each other during development. Current computational accounts either derive one type from the other or use network dynamics to model the emergence of one type in isolation. We introduce a unified recurrent network model that instantiates Dale's Law (every neuron is either excitatory or inhibitory), and is trained to predict the next sensory observation from masked previous sensory observations and egocentric motion. To our knowledge, this is the first single-objective model in which grid and place cells co-emerge without supervision of either type, or reliance on pre-existing spatial-cell representations. We test 1,000 different training configurations and find that grid and place cells robustly co-emerge across a wide range of these settings, with their balance set by recurrent firing-rate noise and sensory masking. The network qualitatively reproduces experimentally observed grid fragmentation in hairpin mazes, grid merging after wall removal, lattice alignment across connected rooms, locally ordered 3D fields observed in freely flying bats, as well as the developmental order in which place cells precede grid cells. We interpret these results in terms of two complementary encoding pressures within a single sensory-prediction objective: (1) correcting errors or reconstructing missing components of sensory observations, and (2) prediction of the next sensory state during navigation. Our results suggest a circuit-level account of the co-emergence of grid and place cells, and experimentally testable predictions for the two kinds of spatial codes. Codes for experiments are available at [1,2].

1 Introduction

Grid cells in the medial entorhinal cortex (MEC) fire periodically across space, forming a triangular lattice that tiles the environment [1, 2]. They have been argued to provide an efficient representation of space [3]. Hippocampal (HC) place cells fire at specific locations [4–6] and remap rapidly between learned environments. While grid cells are thought to emerge from path integration [7, 8], it has been suggested that place cells emerge from encoding sensory experience during navigation [9–11].

[1]Project Page: https://zhaozewang.github.io/grid-and-place
[2]Code: https://github.com/grasp-lyrl/grid_and_place

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

Consistent with these theories, Recurrent Neural Networks (RNNs) trained on path integration develop grid-like activity patterns [8, 12–15], while RNNs trained to auto-encode sensory observations develop place-like activity patterns [10, 11]. In fact, the RNN in [11] trained, using a similar objective, to predict events across time intervals, also develops units displaying the phenomenology of time cells in the hippocampus [16]. Recent theoretical work has emphasized the joint roles of path integration, position encoding, and biological constraints in explaining grid-cell representations [17]. Together, these studies provide well-established theoretical and computational accounts of each cell type in isolation.

In the brain, grid and place cells coexist within a single bidirectionally connected hippocampal-entorhinal circuit [2, 6], and both types provide sparse spatial representations used for self-localization and navigation. In fact, many theories of one cell type rely on the existence of the other. For example, the authors of [18, 19] proposed that grid cells provide a Fourier-like basis from which place cells arise through Hebbian learning [18, 19], while [20, 21] suggested that place cells provide sensory-based error correction that stabilizes grid representations. From the developmental perspective, these potential dependencies create a ***chicken-and-egg problem*** [22, 23]: if each cell type depends on the other, which arises first? An alternative possibility is that the two types co-emerge. We will see that simply adding bi-directional interactions to combine RNNs in which grid cells and place cells emerge individually [8, 10–13] does not lead to co-emergence, perhaps because each type requires distinct recurrent dynamics.

In this paper, we show how grid and place cells can co-emerge in the hidden layers of a single recurrent network that respects Dale's Law (each neuron is constrained to be either excitatory or inhibitory), and is trained to predict the next sensory observation from distorted previous sensory observations and relative rotation/movement signals. The network has no prior knowledge of spatial location or pre-existing spatial cell types. This training objective resembles the next-token prediction goal used to train Large Language Models (LLMs), and also reflects the idea that the hippocampal formation implements a predictive map [9]. We test 1,000 different training configurations by varying the level of recurrent firing-rate noise, the fraction of sensory input masked, and the neuron decay rate, with a different random seed for each configuration. We find that grid and place cells robustly co-exist in our network across a wide range of these settings. The model qualitatively captures experimental observations, including grid-map integration after wall removal, grid-field fragmentation in hairpin mazes, connected-room unification, locally ordered grid patterns in freely flying bats in 3D space, and the emergence of place cells before grid cells during development. By linking co-emergence, circuit constraints, and experimental phenomena, our framework offers a mechanistic account of how grid and place codes can arise together to support spatial navigation. Together, these results suggest a principle for spatial representation in the brain, in which predicting the next sensory observation is sufficient to induce complementary place-cell and grid-cell codes from the implicit spatial structure of sensory experience.

2 From single-cell-type models to co-emergence

Table 1: Comparison with existing theories of grid and place cell emergence.

Model class	**Grid cells**	**Place cells**	**Spatial-cell prior**
Grid-cell circuit and geometric theories [7, 15, 24–28]	Mechanistic / **emergent**	Not modeled	Grid-specific circuit or geometric prior
Path-integration and place-target RNN models [8, 12, 13, 29]	**Emergent**	Predefined input / target	Allocentric coordinates or predefined place-cell targets
Place-cell emergence models [10, 11]	Not modeled	**Emergent**	No grid-cell prior
Successor-representation and predictive-map theories [9]	Derived / spectral	Derived / predictive	Explicit state graph or transition structure
Grid-place interaction and dependency models [18–21]	Predefined	Predefined / mechanistic	Pre-existing grid or place cells
System-level grid-place models [27, 30]	Predefined	Predefined / constructed	Pre-existing spatial-cell representations

Existing models of the entorhinal-hippocampal spatial code fall into distinct families (Table 1): grid-cell circuit and RNN models that rely on spatial supervision or grid-specific priors without modeling place-cell emergence [7, 8, 13, 15, 24–26, 28], place-cell emergence models that obtain localized fields from sensory experiences but do not produce grids [10, 11], and interaction or system-level models that assume that one or both cell types already exist [18, 21, 27]. Our goal in this work is to study how both representations can co-emerge in a single connected circuit.

We will work with a shared recurrent network derived from continuous-time neural dynamics that has also been used in previous work [8, 11–13, 15, 26]. The discrete-time update is given by

$$h_{i,t+1} = (1-\alpha)\, h_{i,t} + \alpha\Big(\textstyle\sum_{j=1}^{N} W_{ij}\, \phi(h_{j,t}) + \sum_{k=1}^{d_{\text{in}}} B_{ik}\, u_{k,t} + b_i \Big), \tag{1}$$

where $h_{i,t}$ is the membrane potential of neuron i at time t, the recurrent weight from neuron j to i is denoted by W_{ij}, the input projection by B, the external input to the network is $u_t \in \mathbb{R}^{d_{\text{in}}}$, the bias is b_i, and the firing-rate nonlinearity (ReLU or softplus) is $\phi(\cdot)$.

2.1 Direct composition of single-cell-type models

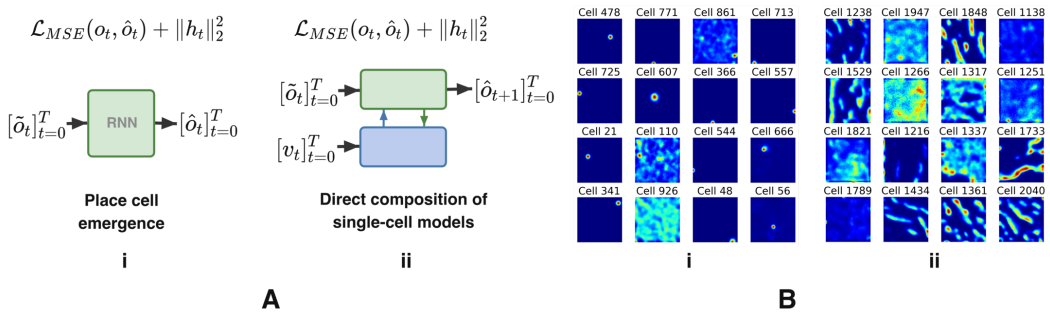


Figure 1: A. (i) An RNN that reconstructs $\hat{o}_t$ from corrupted observations $\tilde{o}_t$ with an ℓ_2 penalty on firing rates leads to place-like cells [11]. (ii) Composition of place and grid cell emergence models within a single network, via bidirectional connections, extends the place cell model in [11] to have an additional input layer, which receives velocity input like the grid cell models in [8, 13]. **B.** Emergent cells in the composed model. (i) Place-like cells in the sensory-driven region. (ii) Irregular stripe- or multi-peaked fields in the "free" region without sensory input. The triangular lattices of grid cell responses are absent.

We first check whether models in which each cell type emerges individually can be combined to get co-emergence of grid and place cells. Sensory denoising or reconstruction during navigation in RNNs (Fig. 1Ai) can produce place-cell-like representations [11]. Similarly, motion-conditioned prediction of spatial location by RNNs can produce grid-cell-like representations [8, 12, 13]. In [13], the network was trained to predict allocentric coordinates, while in [8, 12], location was derived from supervised place cell activity, modeled in [8] as hand-coded Difference of Gaussian (DoG) spatial filters. Since grid-cell models learn by predicting place-cell activity, it is natural to compose the two models by replacing hand-designed place targets with emergent place-like representations.

We start with the emergent place-cell network in [11] (green in Fig. 1Aii), and recurrently connect another network that receives motion input (blue in Fig. 1Aii). Motivated by the successor representation framework for place cells [9], we modify the model of [11] so that it predicts the next sensory input from a noisy and masked current sensory observation. In Fig. 1Aii, we should expect this information to propagate to the blue region via back-propagation and induce units there to predict the place cell activity. This is similar to the direct supervision in [8, 12, 13]. Our trained network successfully reconstructs noisy or masked sensory experiences along simulated trajectories (details in Supp. S.4). It develops place-cell-like representations in the green region (Fig. 1Bi). But the blue region produces neurons with irregular activity stripes or firing fields with multiple peaks (Fig. 1Bii). It does not exhibit triangular lattices characteristic of grid cells.

2.2 Key ingredients for co-emergence

Direct composition of networks with emergent place and grid fields is difficult not only because parameter tuning may be delicate, but also because mechanisms that produce each cell type separately impose distinct constraints on the recurrent circuit as described below. Additionally, in the brain there are constraints imposed by the biophysics of neurons and by the way in which animals experience the world relative to self.

Temporal update constraint. Different spatial codes may require different temporal update dynamics. In our recurrent network, the timescale is controlled by the decay rate α in Eq. 1. Smaller α produces slower, more persistent dynamics and larger α produces faster updates. Place-cell-like denoising benefits from slower dynamics, because partial or corrupted sensory cues must be integrated toward a consistent network state. Grid-cell-like transitions instead require that the network preserves the current state when the animal is stationary, but rapidly updates the state when the animal moves. This cannot be achieved by assigning different fixed decay rates α to different neurons or regions: a fixed rate sets a constant decay timescale independent of behavioral state. What is needed is effectively a gated update, in which the recurrent state is stable in the absence of motion but can be rapidly reconfigured by motion-dependent input. Thus, the difficulty is not only that place and grid codes prefer different timescales, but that grid-like codes require state-dependent switching between persistence and fast update.

Structural constraint. A simple combination like the one in Fig. 1Aii imposes a strong structural constraint: one population is implicitly assigned to the hippocampal role by receiving sensory inputs to support place cells, while the other is implicitly assigned to the MEC role by receiving motion input, and is designated as the putative grid-cell population. This pre-specified division may impede the dynamics required for co-emergence, since each population is strongly shaped by its designated input stream rather than self-organizing under a shared predictive objective. Furthermore, the conventional separation into hippocampus and MEC could simply be a description of functional and structural distinctions that have emerged in a single network that performs a shared task. Thus, pre-assigning separate hippocampal and MEC-like populations may obscure the required self-organization and functional specialization through which grid-like and place-like codes emerge.

Self-motion input constraint. Existing grid-cell RNN models take motion input in terms of allocentric displacement or velocity [8, 12, 13]. This setup assumes that movement is already expressed in an external spatial reference frame. However, during navigation, self-motion cues are more readily available in body-centered forms, such as relative rotation and speed from vestibular, proprioceptive, and motor-related signals, and are presumably converted into allocentric movement signals by network mechanisms that may also exploit external cues and path integration.

Biophysical constraint. In typical RNNs, including the model above, the connection weight matrix is unconstrained. But in the brain neurons satisfy Dale's Law: the outgoing projections of a given neuron are all excitatory (positive weights in an RNN) or inhibitory (negative weights in an RNN)

3 Co-emergence of grid and place cells

These constraints described above have corresponding circuit-level implementations. For the ***temporal update constraint***, recurrent inhibition can support rapid state reconfiguration [31, 32], while the decay term allows activity to decay gradually over time. To address the ***structural constraint***, we do not pre-assign separate hippocampal and MEC populations. Instead, we treat the circuit as a single recurrent population and divide neurons in terms of their external input. *Input-driven neurons* receive direct sensory and motion input. *Recurrently driven neurons* receive no direct external input and are shaped only through recurrent interactions with the rest of the network. This design avoids imposing a predefined grid/place or HC/MEC connectivity structure, allowing potential grid- and place-like subpopulations to self-organize under a shared predictive objective. Finally, to address the ***self-motion input constraint***, we provide motion as egocentric self-motion signals—-rotation relative to previous heading and speed—rather than allocentric displacement or velocity. This forces the network to learn spatial codes from egocentric sensory experience and body-centered motion cues without assuming movement expressed in a global spatial reference frame. To implement the ***biophysical constraint***, we require the recurrent weight matrix to satisfy Dale's Law by assigning a fixed sign to the weights of outgoing synapses of each neuron throughout training. This is a biologically realistic recurrent structure through which inhibitory feedback can shape dynamics.

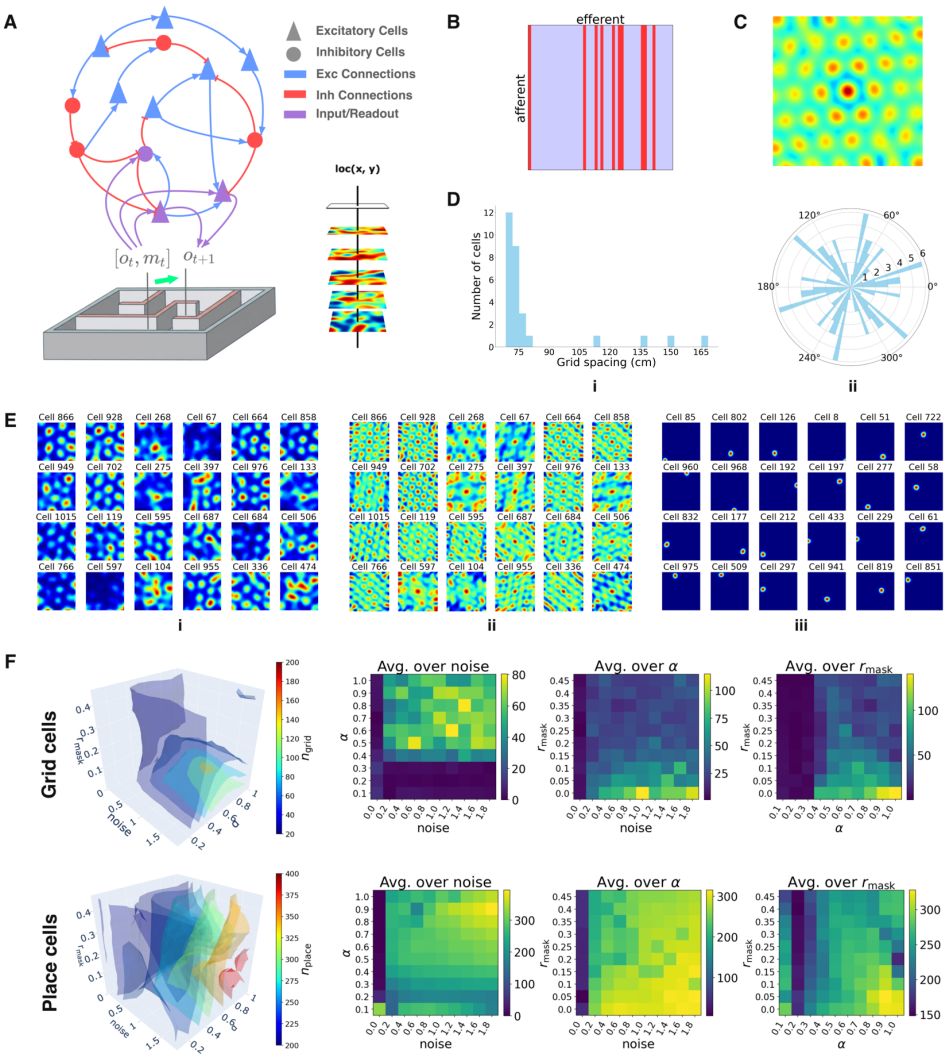


Figure 2: **A.** Architecture of the RNN. Excitatory (triangles) and inhibitory (circles) neurons interact through recurrent connections constrained by Dale's law. The hidden layer is split into input-driven neurons, which receive noisy masked sensory inputs $\tilde{o}_t$ and egocentric motion input m_t, and recurrently driven neurons, which receive no direct external input. The motion signal consists of relative rotation from the previous timestep, represented as a unit vector, and a scalar speed normalized to $[0, 1]$. **B.** Illustration of the recurrent weight matrix. Rows index efferent neurons and columns index afferent neurons; 80% of neurons are excitatory (blue, non-negative outgoing weights) and 20% inhibitory (red, non-positive outgoing weights). **C.** Population-averaged spatial autocorrelogram of cells classified as grid cells. Each autocorrelogram was peak-normalized before averaging, so the panel reflects the shared lattice structure instead of being dominated by high-rate cells. The six satellite peaks around the center summarize the population-level triangular periodicity. **D.** Grid spacing and orientation statistics across emergent grid cells. (i) Distribution of grid spacings in centimeters, measured from the spatial autocorrelogram peaks. (ii) Distribution of grid orientations. Orientation is measured as the angle between the horizontal axis of the rate map and the nearest grid-lattice axis in the spatial autocorrelogram, modulo $60°$ to account for hexagonal symmetry. The polar plot repeats this distribution to extend across the circle visualization purposes. **E.** Example single-cell ratemaps. (i) Grid cell response field from the free region. (ii) Spatial autocorrelograms of these grid cells, showing triangular lattice symmetry. (iii) Place cell response fields from the sensory-driven region. **F.** Parameter sweep over noise level σ_n, decay rate α, and masking fraction r_{mask}. Top row: grid cell counts shown as isosurfaces (left) and marginal heatmaps averaged across different parameters (right). To aid visualization, cell counts were smoothed across adjacent parameter settings before extracting the isosurfaces. Bottom row: place cell counts in the same layout.

We implement these conditions in a recurrent network with $N = 2048$ neurons. Half of the neurons receive direct external input, while the remaining half are only driven through recurrent connections. At initialization, 80% of neurons are deemed excitatory (non-negative outgoing weights), and 20% are deemed inhibitory (non-positive outgoing weights), consistently with typical proportions in hippocampus and cortex, and this constraint is enforced during training (see Supp. S.2). The precise ratio between input-driven and recurrently driven neurons does not qualitatively affect the results presented below (see Supp. S.16.1).

At each step the network receives a masked sensory observation $\tilde{o}_t$ (a fraction r_{mask} dimensions are masked) and an egocentric motion signal m_t consisting of relative rotation and a scalar speed normalized to $[0, 1]$, and predicts the next observation $\hat{o}_{t+1}$. The network is trained to minimize the mean square error between the predictions and ground-truth unmasked observations. Sensory observations are generated over a 220 cm $\times$ 220 cm arena so that nearby locations have similar observations (details in Supp. S.3). After each recurrent update, we inject additive Gaussian noise with amplitude controlled by a parameter σ_n and modulated by the firing rate (Supp. Eq. S4), so that neurons that are more active experience larger perturbations. In the main model, we fix the recurrent bias b_i in Eq. 1 to zero.

3.1 Robust co-emergence of grid and place cells

After training, the network develops grid- and place-like representations in the recurrently driven and input-driven neurons, respectively (Fig. 2E.i,iii). In an example trained network, we identified 58 grid-like units using a grid-score threshold of > 0.3, predominantly among recurrently driven neurons, and 307 place-like units across both input-driven and recurrently driven neurons (see Supp. S.8 for classification methods). The emerged grid cells exhibit multiple firing fields arranged in a triangular lattice (Fig. 2D.i). Grid orientation is measured from the spatial autocorrelogram as the angle between the horizontal axis of the rate map and the nearest grid-lattice axis modulo 60° to account for triangular symmetry (Fig. 2D.ii). The identified grid cells show the expected triangular symmetry in their autocorrelograms (Fig. 2E.ii). To further test whether the identified grid cells exhibit characteristic grid-like firing patterns rather than random spatial patterns, we randomly shift each cell's activity in time relative to the agent's position 500 times, recalculating its grid score after each shift. For each cell, we use the 95th percentile of these shuffled scores as a threshold, so that only 5% of shuffled scores exceed it. Across 10 models, the original grid score of every candidate grid cell (GS > 0.1) exceeds its corresponding threshold (see Supp. S.9). Across 10 random-seed replicates, the classification criteria with GS > 0.3 yielded 30.0 ± 19.2 grid-like units and 299.4 ± 4.9 place-like units (mean $\pm$ s.d.). The greater seed-to-seed variability in the grid-like population may partly reflect the sensitivity of existing grid-cell classification metrics. All networks trained with different random seeds robustly developed units with periodic firing fields that qualitatively resembled grid cells.

We vary three parameters to test the robustness of the model: the activity decay rate $\alpha \in [0.1, 1.0]$, noise level $\sigma_n \in [0, 0.45]$ in the recurrent network, and input masking fraction $r_{\text{mask}} \in [0, 0.9]$, over a $10 \times 10 \times 10$ grid (details in Supp. S.11). We train each model for 200,000 time steps with a *different random seed* and identify cell types using the criteria in Supp. S.8. For this sweep, we use the more permissive grid-score threshold of 0.1, which better captures how grid-like periodic structure varies across training conditions. Fig. 2F shows that place cells emerge robustly across nearly all settings. Grid-like periodicity preferentially emerges at higher σ_n, and is modulated more weakly by the decay rate α. Higher masking ratio r_{mask} causes place cells to also appear among recurrently driven neurons, while grid-like periodicity fails to emerge.

To better understand which architectural components support the co-emergence of grid and place cells, we modify individual components while keeping the other settings unchanged. First, we remove the **Dale's Law constraint**, allowing all recurrent weights to be freely learned. This reduces the number of grid cells among recurrently driven neurons from 58 to 3 under the grid-score threshold of 0.3. The remaining cells develop multi-peaked or stripe-like irregular fields without triangular lattice structure, similar to the outcome of the direct composition model in Sec. 2.2. We also test whether grid and place cells still emerge when we train the network to reconstruct the current sensory observation from masked input rather than predict the next sensory observation. Under this condition, we still find 42 grid cells with grid scores above 0.3 and 329 place cells. Finally, we vary the **motion input** by training networks with allocentric motion input, and also with motion input removed completely. Grid- and place-like cells still emerge in both settings, suggesting that explicit motion input is not

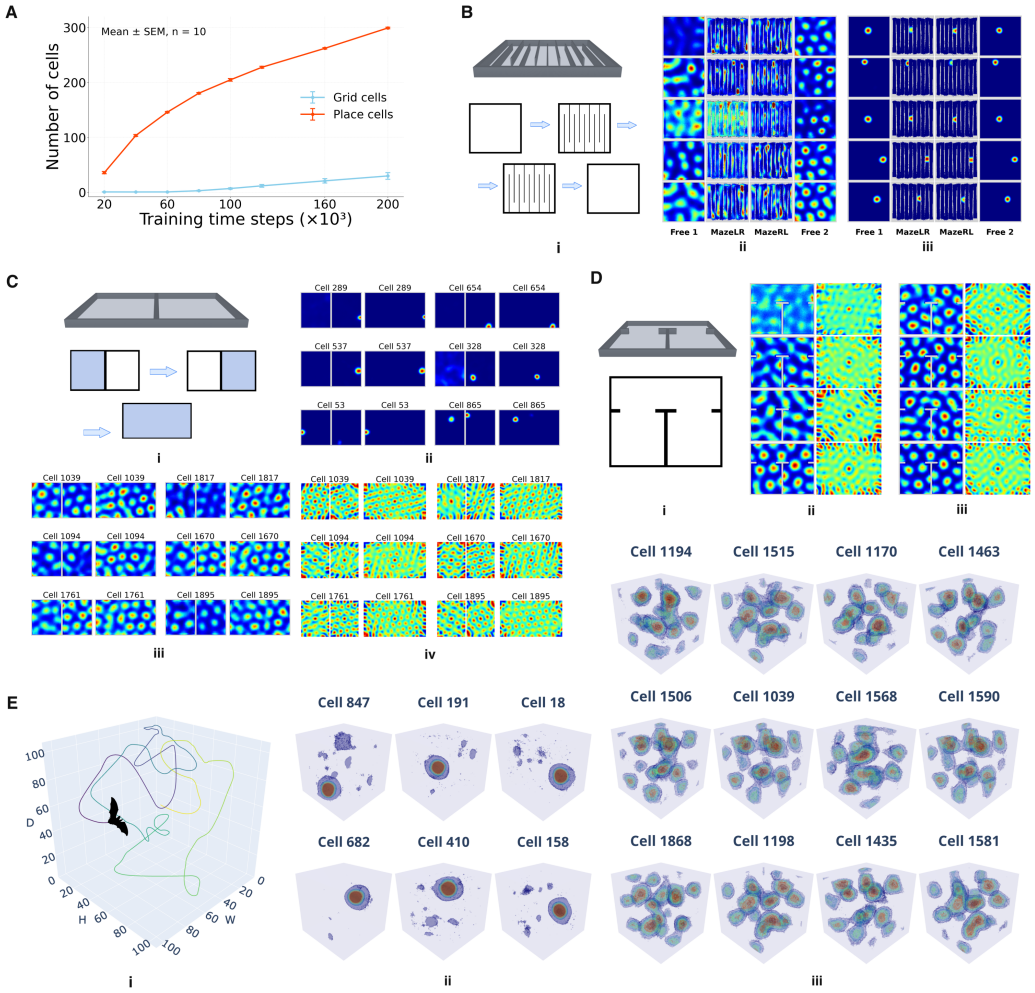


Figure 3: **A.** Developmental order of emerged representations. Grid-cell counts (cyan; grid score > 0.3) and place-cell counts (orange) are shown across training as mean $\pm$ SEM over 10 random-seed replicates. Points show the eight sampled training time steps from 20,000 to 200,000, with lines connecting the means. Place-cell counts rise earlier than grid-cell counts. **B.** Hairpin maze [33]. (i) Schematic of the open-field and hairpin trials. The animal first explores an open arena, then traverses the hairpin maze under constrained trajectories, and finally returns to the open arena. (ii) Grid-cell ratemaps fragment across corridors when movement is constrained, but recover coherent hexagonal patterns in the open arena. (iii) Place cells remain spatially stable across conditions. **C.** Two-room wall removal. (i) Schematic of two rooms separated by a wall and then merged after wall removal. The animal sequentially explores the left room, the right room, and the merged arena. (ii–iv) Ratemaps and autocorrelograms across trials. For compactness, ratemaps from the left- and right-room trials are plotted together with a wall in between, while even columns show responses after wall removal for the same cell. (ii) Place-cell firing centers remain stable before and after wall removal. (iii) Grid cells develop independent hexagonal patterns in each room when the wall is present, and merge into a unified lattice after wall removal. (iv) Autocorrelograms show the same transition from two independent lattices to one unified lattice. **D.** Connected rooms [34]. (i) Schematic of two rooms connected by a corridor. The animal can freely traverse between rooms through the corridor. (ii–iii) Grid-cell ratemaps and autocorrelograms early and late in training. Early in training, grid fields are locally ordered within each room but remain fragmented across the connected environment. Later in training, some cells develop more globally ordered patterns, with grid fields becoming aligned across both rooms and the connecting corridor. **E.** Three-dimensional volumetric traversal. (i) Example 3D trajectory. (ii) Place-like cells develop localized 3D firing fields. (iii) Grid-like cells develop locally ordered volumetric fields, consistent with partially ordered grid responses recorded in flying bats.

required when consecutive sensory observations already contain smooth transition information along trajectories.

4 Properties of the emergent grid and place cells

Next we test whether the co-emergent grid/place cells exhibit properties of their biological counterparts.

Developmental order (Fig. 3A). Place cells emerge before grid cells during training (Fig. 3A), consistently with recordings during rodent development [35–37].

Hairpin maze (Fig. 3B). We replicate the experiment of [33], in which grid cells were recorded as rodents first explored an open field, then traversed an imposed hairpin maze, and then returned to the open field. We model this setting by using the same sensory noise field across environments, slightly modified to respect the maze walls (details in Supp. S.13), so that corresponding spatial locations in the maze and open field provide similar sensory observations while local sensory similarity in the hairpin maze follows the corridor structure. In the open arena, grid cells exhibit coherent triangular patterns. In the hairpin maze, where movement is constrained to narrow corridors, grid patterns fragment across corridors and the spatial autocorrelograms become banded rather than triangular. Upon returning to the open field, triangular firing patterns gradually recover as training continues in the open field. These results recapitulate [33]. However, in our model, place cells fire at corresponding locations in the open arena and hairpin maze, without the repetition across alternating arms reported by [33]. This may be because our simulations provide distinct sensory inputs to different arms, whereas similar running directions and viewing angles may produce similar sensory cues in the animal experiment. We therefore predict that, when distinct visual landmarks or olfactory cues allow the animal to distinguish individual arms, individual place cells would fire selectively in a single arm rather than repeat their firing fields across alternating arms.

Two-room wall removal (Fig. 3C). We train the network for 60,000 steps in room 1 and 60,000 steps in room 2 while the rooms are separated by a wall, followed by 80,000 steps in the combined arena after wall removal. This tests whether the learned spatial code treats the two rooms as separate environments or as one continuous space. When the wall is present, grid cells develop independent triangular lattice patterns in the two compartments, visible in both ratemaps and autocorrelograms. After wall removal, grid patterns merge into a unified lattice spanning both rooms, while place cells maintain localized and spatially stable firing fields. The integration of grid maps after wall removal is consistent with [38], who observed local reorganization near the former partition while grid fields near distal walls remained largely stable. Our model further predicts that place-cell firing centers remain stable after wall removal when sensory cues at corresponding locations remain similar.

Connected rooms (Fig. 3D). Following [34], we connect two rooms by a corridor that the animal can freely traverse. The original experiment tested whether grid cells form local maps within each room or a global map across connected spaces. This setting is related to the wall-removal experiment, but is more challenging because the rooms are linked only indirectly through a corridor, requiring spatial periodicity to align across a longer traversal path. We therefore compare grid-cell responses early and late in training. Early in training, grid cells show fragmented, locally triangular lattice patterns within each room (Fig. 3D.ii). Later in training, some cells develop global patterns, with grid fields and periodicity becoming ordered across the connected environment despite the intervening walls, resembling findings in [34]. Although not all cells form globally ordered lattices, grid cells in both groups preserve locally ordered structure, with local grid orientation gradually drifting across locations. This suggests that the network tends toward global alignment, but errors accumulated over long indirect traversal paths can prevent perfectly coherent global lattices.

Three-dimensional traversal (Fig. 3E). Finally, we train the network in a 3D environment in which the agent moves freely through the full volume. The trajectory is generated by the same smoothed random-walk procedure with boundary avoidance used in 2D environments (Supp. S.4). We find that place cells develop clear localized 3D firing fields. Grid cells also develop structured volumetric firing fields, but these fields are locally ordered rather than organized as a perfect global lattice, resembling locally ordered grid-like responses observed in freely flying bats [39]. The local ordering that we see may reflect the difficulty of uniformly traversing a 3D volume. As discussed in the next section, grid formation in our network its trajectory-dependent; the difficulty of uniformly traversing and sampling a 3D volume may be leading to the irregularity of the emergent grid fields.

5 Discussion

Our goal in this paper is to build a conceptual framework for understanding the functional logic of the circuits in the animal brain that facilitate spatial navigation. In our model, training the network to predict the next sensory observation from masked sensory observations and egocentric motion is sufficient for grid and place cells to co-emerge. This predictive capability underpins goal-directed planning. We instantiate this objective in an RNN model of the hippocampal formation with a few key biological constraints. Chief among these is Dale's Law—a neuron has either excitatory or inhibitory projections, but not both. We also include firing rate modulated noise, and relative proportions of excitation and inhibition similar to those in the brain. We do not seek here to develop a detailed description of the biological circuits, but rather to reveal the essential constraints and architectural motifs that are required. As such, rather than separating the network into a putative MEC (where grid cells would be found) receiving motion input and a putative hippocampus (where place cells would be found) receiving sensory input, we consider an architecture in which one component receives external input (sensory and motion) and another only has recurrent connections. This allows us to better understand the dynamics that lead to the formation of the two cell types. After learning, the network robustly develops place-cell-like units in the component receiving sensory input and grid-cell-like units in the other component. The emergent place and grid cells reproduce a number of experimentally observed phenomena, including earlier development of place fields, fragmentation of grid fields in hairpin mazes, formation of global grid representations in connected environments, and 3D grid fields with local but not global order. Our model predicts that, when sensory cues distinguish individual arms of a hairpin maze, individual place cells should fire selectively in a single arm rather than across alternating arms. It also predicts that place-field centers remain stable after wall removal when sensory cues at corresponding locations remain similar. The simplicity of our model makes it possible to easily predict the consequences in new experimental regimes.

Why do place and grid representations appear in the trained network? The animal's sensory experience traces a continuous trajectory on a manifold within high-dimensional sensory space, which is constrained to have the same low dimension as the physical space being explored. In this context, predicting the next sensory observation can be decomposed into two distinct problems: (i) ***off-manifold contraction***, and (ii) ***on-manifold transition***. In detail, masking observations perturbs them away from the manifold of valid sensory states. The network pulls the perturbed state back to the manifold. The mechanism for doing so has to be a spatially localized attractor basin – otherwise it might correct to the wrong location. Wang et al. [30] showed that this correction occurs in REMI and is accompanied by place-cell-like responses, although grid-cell responses are explicitly supervised. In our model, both cell types emerge without such supervision. This suggests that place cells can provide coordinates on the manifold of valid sensory experiences. Sensory observations change smoothly as the animal moves along a trajectory. To predict the next observation, its circuit should use the current motion to determine the next location on the sensory manifold. If the spatial location were represented in some local chart, it could be updated using the current velocity by a recurrent circuit. Grid cells can be interpreted as maintaining such a local chart [3, 40, 41]. This update can be fed back to the circuit maintaining the sensory manifold (which contains the place cells). Dorrell and Whittington [17] link grid representations to position encoding and path integration. Our model additionally considers correction from noisy sensory observations, suggesting a complementary role for place cells in anchoring these representations to sensory experience. Altogether, this predictively updates the location on the sensory manifold.

Our model aligns with several existing theories of grid and place cells. ***Continuous-attractor models*** show how recurrent circuit dynamics maintain and update an internal spatial state during movement [7]. Our model similarly treats grid cells as supporting motion-driven transitions, but explains how they can arise in a Dale's Law-constrained recurrent circuit trained by next-observation prediction. Our model also follows existing ***path-integration RNN models*** [8, 12, 13], but addresses two assumptions in these models. First, instead of assuming Difference-of-Gaussian (DoG) place-cell targets [8], our model treats place cells as emergent from an excitatory-inhibitory network. Second, instead of requiring allocentric motion inputs, grid cells emerge in the network with egocentric inputs consistent with the experience of animals, or even without explicit motion signals. In the latter case, the network can infer motion information directly from changes in sensory signals. On the other hand, ***self-supervised grid-cell models*** [15, 26] train grid representations using explicit geometric constraints, such as contrastive similarity between nearby positions [26] or local distance and angle preservation [15]. Our model suggests a possible source for these geometry-preserving constraints:

adjacent locations produce similar sensory experience, so next-observation prediction implicitly encourages nearby states to have related representations. Finally, our theory builds directly on ***episodic-memory theories*** of place cells [10, 11], while connecting them to ***successor-representation*** accounts [9]. In our model, predictive structure is supported not only by place-like state specificity but also by grid-like action-conditioned transitions along the sensory manifold.

5.1 Limitations and Future Directions

Grid cells in the MEC are known to be organized in a hierarchy of discrete modules [42]. Our network only produced a single module in which the grid cells have similar periods. It is possible to encode space with such a distribution of periods [40], but it is known that the discrete scaling hierarchy of the type observed in [42] is more efficient [3]. Perhaps such modules would be obtained if our network were to have multiple layers or some further regularization that incentivizes representational efficiency [25, 28]. Furthermore, experiments show that if an enclosure is suddenly stretched or shrunk, grid fields appear to distort correspondingly [43–45]. There is evidence that this apparent distortion arises from time-averaging phase-shifts in the grid pattern that occur when the animal approaches a boundary in the distorted direction, possibly because of interaction between grid cells and border cells [46, 47]. It would be interesting to test whether our network reproduces such distortion effects, and/or whether it contains emergent border cells in addition to the place and grid cells that we have analyzed. Finally, local predictive-coding rules can produce grid cells from predefined place-cell inputs [29]. Extending such rules to our model could help address the biological limitations of backpropagation through time.

Acknowledgments and Disclosure of Funding

The study was supported by NIH CRCNS grant 1R01MH125544-01 and in part by the NSF and DoD OUSD (R&E) under Agreement PHY-2229929 (The NSF AI Institute for Artificial and Natural Intelligence). Additional support was provided by the United States–Israel Binational Science Foundation (BSF). PC was supported in part by grants from the National Science Foundation (IIS-2145164, CCF-2212519).

References

[1] 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, August 2005. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature03721.

[2] Edvard I. Moser, Emilio Kropff, and May-Britt Moser. Place Cells, Grid Cells, and the Brain's Spatial Representation System. *Annu. Rev. Neurosci.*, 31(1):69–89, July 2008. ISSN 0147-006X, 1545-4126. doi: 10.1146/annurev.neuro.31.061307.090723.

[3] Xue-Xin Wei, Jason Prentice, and Vijay Balasubramanian. A principle of economy predicts the functional architecture of grid cells. *Elife*, 4:e08362, 2015.

[4] J. O'Keefe and J. Dostrovsky. The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, 34(1):171–175, November 1971. ISSN 00068993. doi: 10.1016/0006-8993(71)90358-1.

[5] John O'Keefe. Place units in the hippocampus of the freely moving rat. *Experimental Neurology*, 51(1):78–109, January 1976. ISSN 00144886. doi: 10.1016/0014-4886(76)90055-8.

[6] May-Britt Moser, David C. Rowland, and Edvard I. Moser. Place Cells, Grid Cells, and Memory. *Cold Spring Harb Perspect Biol*, 7(2):a021808, February 2015. ISSN 1943-0264. doi: 10.1101/cshperspect.a021808.

[7] Yoram Burak and Ila R. Fiete. Accurate Path Integration in Continuous Attractor Network Models of Grid Cells. *PLoS Comput Biol*, 5(2):e1000291, February 2009. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1000291.

[8] 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.e13, January 2023. ISSN 08966273. doi: 10.1016/j.neuron.2022.10.003.

[9] Kimberly L Stachenfeld, Matthew M Botvinick, and Samuel J Gershman. The hippocampus as a predictive map. *Nat Neurosci*, 20(11):1643–1653, November 2017. ISSN 1097-6256, 1546-1726. doi: 10.1038/nn.4650.

[10] Marcus K. Benna and Stefano Fusi. Place cells may simply be memory cells: Memory compression leads to spatial tuning and history dependence. *Proc. Natl. Acad. Sci. U.S.A.*, 118(51): e2018422118, December 2021. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.2018422118.

[11] Zhaoze Wang, Ronald W. Di Tullio, Spencer Rooke, and Vijay Balasubramanian. Time Makes Space: Emergence of Place Fields in Networks Encoding Temporally Continuous Sensory Experiences. In *NeurIPS*, 2024. doi: 10.1101/2024.08.11.607484.

[12] Andrea Banino, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, Martin J. Chadwick, Thomas Degris, Joseph Modayil, Greg Wayne, Hubert Soyer, Fabio Viola, Brian Zhang, Ross Goroshin, Neil Rabinowitz, Razvan Pascanu, Charlie Beattie, Stig Petersen, Amir Sadik, Stephen Gaffney, Helen King, Koray Kavukcuoglu, Demis Hassabis, Raia Hadsell, and Dharshan Kumaran. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705):429–433, May 2018. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-018-0102-6.

[13] Christopher J. Cueva and Xue-Xin Wei. Emergence of grid-like representations by training recurrent neural networks to perform spatial localization, March 2018.

[14] Tianhao Chu, Yuling Wu, Neil Burgess, Zilong Ji, and Si Wu. Unfolding the Black Box of Recurrent Neural Networks for Path Integration. In *NeurIPS*, 2025.

[15] Dehong Xu, Ruiqi Gao, Wen-Hao Zhang, Xue-Xin Wei, and Ying Nian Wu. On Conformal Isometry of Grid Cells: Learning Distance-Preserving Position Embedding, February 2025.

[16] Qiaorong S Yu, Zhaoze Wang, and Vijay Balasubramanian. When and where: A model hippocampal network unifies formation of time cells and place cells. *bioRxiv*, pages 2026–03, 2026.

[17] William Dorrell and James C. R. Whittington. If grid cells are the answer, what is the question? a review of normative grid cell theory. *arXiv preprint arXiv:2601.12424*, 2026. doi: 10.48550/arXiv.2601.12424.

[18] Trygve Solstad, Edvard I. Moser, and Gaute T. Einevoll. From grid cells to place cells: A mathematical model. *Hippocampus*, 16(12):1026–1031, December 2006. ISSN 1050-9631, 1098-1063. doi: 10.1002/hipo.20244.

[19] Edmund T. Rolls, Simon M. Stringer, and Thomas Elliot. Entorhinal cortex grid cells can map to hippocampal place cells by competitive learning. *Network: Computation in Neural Systems*, 17 (4):447–465, January 2006. ISSN 0954-898X, 1361-6536. doi: 10.1080/09548980601064846.

[20] Tora Bonnevie, Benjamin Dunn, Marianne Fyhn, Torkel Hafting, Dori Derdikman, John L Kubie, Yasser Roudi, Edvard I Moser, and May-Britt Moser. Grid cells require excitatory drive from the hippocampus. *Nat Neurosci*, 16(3):309–317, March 2013. ISSN 1097-6256, 1546-1726. doi: 10.1038/nn.3311.

[21] César Rennó-Costa and Adriano B.L. Tort. Place and Grid Cells in a Loop: Implications for Memory Function and Spatial Coding. *J. Neurosci.*, 37(34):8062–8076, August 2017. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.3490-16.2017.

[22] Daniel Bush, Caswell Barry, and Neil Burgess. What do grid cells contribute to place cell firing? *Trends in Neurosciences*, 37(3):136–145, March 2014. ISSN 01662236. doi: 10.1016/j.tins.2013.12.003.

[23] Genela Morris and Dori Derdikman. The chicken and egg problem of grid cells and place cells. *Trends in Cognitive Sciences*, 27(2):125–138, February 2023. ISSN 13646613. doi: 10.1016/j.tics.2022.11.003.

[24] Neil Burgess, Caswell Barry, and John O'Keefe. An oscillatory interference model of grid cell firing. *Hippocampus*, 17(9):801–812, September 2007. ISSN 1050-9631, 1098-1063. doi: 10.1002/hipo.20327.

[25] Louis Kang and Vijay Balasubramanian. A geometric attractor mechanism for self-organization of entorhinal grid modules. *eLife*, 8, August 2019. doi: 10.7554/eLife.46687.

[26] Rylan Schaeffer, Tzuhsuan Ma, Sanmi Koyejo, Mikail Khona, Cristóbal Eyzaguirre, and Ila Rani Fiete. Self-Supervised Learning of Representations for Space Generates Multi-Modular Grid Cells. In *NeurIPS*, September 2023.

[27] Sarthak Chandra, Sugandha Sharma, Rishidev Chaudhuri, and Ila Fiete. Episodic and associative memory from spatial scaffolds in the hippocampus. *Nature*, 638(8051):739–751, February 2025. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-024-08392-y.

[28] Mikail Khona, Sarthak Chandra, and Ila Fiete. Global modules robustly emerge from local interactions and smooth gradients. *Nature*, 640(8057):155–164, April 2025. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-024-08541-3.

[29] Mufeng Tang, Helen Barron, and Rafal Bogacz. Learning grid cells by predictive coding. *arXiv preprint arXiv:2410.01022*, 2024. doi: 10.48550/arXiv.2410.01022.

[30] Zhaoze Wang, Genela Morris, Dori Derdikman, Pratik Chaudhari, and Vijay Balasubramanian. REMI: Reconstructing Episodic Memory During Internally Driven Path Planning. In *NeurIPS*, 2025.

[31] Misha V. Tsodyks, William E. Skaggs, Terrence J. Sejnowski, and Bruce L. McNaughton. Paradoxical Effects of External Modulation of Inhibitory Interneurons. *J. Neurosci.*, 17(11):4382–4388, June 1997. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.17-11-04382.1997.

[32] Carl van Vreeswijk and Haim Sompolinsky. Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science*, 274(5293):1724–1726, 1996. Accessed: 29 Apr 2026.

[33] Dori Derdikman, Jonathan R Whitlock, Albert Tsao, Marianne Fyhn, Torkel Hafting, May-Britt Moser, and Edvard I Moser. Fragmentation of grid cell maps in a multicompartment environment. *Nat Neurosci*, 12(10):1325–1332, October 2009. ISSN 1097-6256, 1546-1726. doi: 10.1038/nn.2396.

[34] Francis Carpenter, Daniel Manson, Kate Jeffery, Neil Burgess, and Caswell Barry. Grid Cells Form a Global Representation of Connected Environments. *Current Biology*, 25(9):1176–1182, May 2015. ISSN 09609822. doi: 10.1016/j.cub.2015.02.037.

[35] Tom J. Wills, Francesca Cacucci, Neil Burgess, and John O'Keefe. Development of the Hippocampal Cognitive Map in Preweanling Rats. *Science*, 328(5985):1573–1576, June 2010. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1188224.

[36] Rosamund F. Langston, James A. Ainge, Jonathan J. Couey, Cathrin B. Canto, Tale L. Bjerknes, Menno P. Witter, Edvard I. Moser, and May-Britt Moser. Development of the Spatial Representation System in the Rat. *Science*, 328(5985):1576–1580, June 2010. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1188210.

[37] Laurenz Muessig, Jonas Hauser, Thomas Joseph Wills, and Francesca Cacucci. A Developmental Switch in Place Cell Accuracy Coincides with Grid Cell Maturation. *Neuron*, 86(5):1167–1173, June 2015. ISSN 08966273. doi: 10.1016/j.neuron.2015.05.011.

[38] Tanja Wernle, Torgeir Waaga, Maria Mørreaunet, Alessandro Treves, May-Britt Moser, and Edvard I. Moser. Integration of grid maps in merged environments. *Nature Neuroscience*, 21(1):92–101, 2018. doi: 10.1038/s41593-017-0036-6. URL https://www.nature.com/articles/s41593-017-0036-6.

[39] Gily Ginosar, Johnatan Aljadeff, Yoram Burak, Haim Sompolinsky, Liora Las, and Nachum Ulanovsky. Locally ordered representation of 3D space in the entorhinal cortex. *Nature*, 596 (7872):404–409, August 2021. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-021-03783-x.

[40] Ila R Fiete, Yoram Burak, and Ted Brookings. What grid cells convey about rat location. *Journal of Neuroscience*, 28(27):6858–6871, 2008.

[41] Daniel Bush, Caswell Barry, Daniel Manson, and Neil Burgess. Using Grid Cells for Navigation. *Neuron*, 87(3):507–520, August 2015. ISSN 08966273. doi: 10.1016/j.neuron.2015.07.006.

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

[43] Caswell Barry, Robin Hayman, Neil Burgess, and Kathryn J Jeffery. Experience-dependent rescaling of entorhinal grids. *Nature neuroscience*, 10(6):682–684, 2007.

[44] Tor Stensola, Hanne Stensola, May-Britt Moser, and Edvard I. Moser. Shearing-induced asymmetry in entorhinal grid cells. *Nature*, 518(7538):207–212, February 2015. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature14151.

[45] Julija Krupic, Marius Bauza, Stephen Burton, Caswell Barry, and John O'Keefe. Grid cell symmetry is shaped by environmental geometry. *Nature*, 518(7538):232–235, February 2015. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature14153.

[46] Kiah Hardcastle, Surya Ganguli, and Lisa M Giocomo. Environmental boundaries as an error correction mechanism for grid cells. *Neuron*, 86(3):827–839, 2015.

[47] Alexandra T Keinath, Russell A Epstein, and Vijay Balasubramanian. Environmental deformations dynamically shift the grid cell spatial metric. *eLife*, 7:e38169, oct 2018. ISSN 2050-084X. doi: 10.7554/eLife.38169. URL https://doi.org/10.7554/eLife.38169.

[48] 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.

[49] Lisa M. Giocomo, Syed A. Hussaini, Fan Zheng, Eric R. Kandel, May-Britt Moser, and Edvard I. Moser. Grid cells use HCN1 channels for spatial scaling. *Cell*, 147(5):1159–1170, 2011. doi: 10.1016/j.cell.2011.08.051.

S Supplementary material

S.1 Continuous-time RNN dynamics

The discrete-time update in Eq. 1 is derived from the following continuous-time neural dynamics:

$$\tau \frac{dh_i(t)}{dt} = -h_i(t) + \sum_{j=1}^{N} W_{ij}\, \phi\big(h_j(t)\big) + \sum_{k=1}^{d_{\text{in}}} B_{ik}\, u_k(t) + b_i, \tag{S1}$$

where τ is the membrane time constant. Applying Euler discretization with step Δt and defining the leak rate $\alpha = \Delta t/\tau$, we obtain the update rule used in the main text. Throughout this work we use $\phi(\cdot) = \mathrm{softplus}(\cdot)$ as the firing-rate nonlinearity, which provides a smooth non-negative activation. In the main model, we fix the recurrent bias b_i to zero.

S.2 Excitatory-inhibitory structure and homeostatic synaptic scaling

Sign assignment. The recurrent matrix $W \in \mathbb{R}^{N\times N}$ acts on a hidden vector of size $N = N_{\text{input-drv}} + N_{\text{rec-drv}}$, with $N_{\text{input-drv}} = N_{\text{rec-drv}} = 1024$ in all main experiments. Each column j of W (the outgoing weights of neuron j) is constrained to a single sign: 80% of neurons are randomly assigned to be excitatory (sign $+1$) and 20% inhibitory (sign -1). The sign vector $s \in \{-1, +1\}^N$ is drawn at initialization and held fixed throughout the training. We parameterize the recurrent matrix as

$$W_{ij} = |M_{ij}|\, s_j, \tag{S2}$$

where $M \in \mathbb{R}^{N\times N}$ is a learnable magnitude matrix. Because only $|M|$ enters Eq. S2, the sign of M_{ij} is irrelevant and the column sign s_j is preserved through every gradient step, enforcing Dale's law without projection.

Initialization. The magnitude matrix M is initialized as in Algorithm 1: sample a Gaussian matrix, apply Dale signs columnwise, normalize the spectral radius to 1, and store the magnitudes into M. For each neuron j, we record its total outgoing synaptic strength $c_j^* = \sum_i M_{ij}$ for the homeostatic scaling (Eq. S3). We describe the motivation for this homeostatic constraint and its role during training in the next section.

Algorithm 1 Recurrent weight initialization

Sample $\tilde{W}_{ij} \sim \mathcal{N}(0, 1/N)$
$\tilde{W}_{ij} \leftarrow s_j\, |\tilde{W}_{ij}|$ # apply Dale signs
$\tilde{W} \leftarrow \tilde{W}/\rho(\tilde{W})$ # normalize spectral radius to 1
$M_{ij} \leftarrow |\tilde{W}_{ij}|$ # store magnitudes; signs reapplied via Eq. S2
$c_j^* \leftarrow \sum_i M_{ij}$ # total outgoing synaptic strength of neuron j

Homeostatic synaptic scaling. Back-propagation in the RNN may update synaptic weights much faster than biological plasticity would allow, which can destabilize the training dynamics. To mitigate this, after each weight update, we apply a normalization that softly pulls each neuron's total outgoing synaptic strength toward its initial value:

$$M_{ij} \leftarrow M_{ij} \left(\frac{c_j^*}{\sum_k M_{kj}} \right)^{\eta}, \tag{S3}$$

with $\eta = 10^{-3}$. This regularization does not prevent co-emergence, but without it the network often fails to learn new environments and instead overfits to the first room.

Multiplicative firing-rate noise. Previous studies have suggested that place cells may emerge from denoising corrupted sensory input [11]. We consider two sources of such corruption. The first kind is partial occlusion of sensory cues during navigation. We model this by randomly masking sensory input channels independently across time and sensory dimensions, replacing masked values with the mean value of the corresponding sensory channel. The second is stochasticity in neural activity, this models cortical firing which is approximately Poisson-like with rate-dependent variance. Since our

model operates on firing rates rather than explicit spikes, we approximate this variability by adding Gaussian noise whose magnitude depends on the firing rate:

$$\tilde{h}_i \leftarrow \max\Big(0,\ h_i + \sqrt{\sigma_n h_i + \epsilon}\,\xi_i\Big), \quad \xi_i \sim \mathcal{N}(0,1), \tag{S4}$$

where σ_n is the noise level. Clipping enforces non-negativity of firing rates in our model, since actual spike counts are strictly non-negative.

Input and output projections. In our network, we split the hidden layer into two regions: (i) where neurons are directly driven by the sensory and motion input signal to predict the next sensory observation, and (ii) where neurons receive input only indirectly from the first region. We refer to the first region as the sensory-driven region, and the second as recurrent-driven region. The sensory-driven population receives input $u_t = [\tilde{o}_t;\, m_t] \in \mathbb{R}^{d_{\text{in}}}$ with $d_{\text{in}} = d_{\text{obs}} + d_{\text{motion}}$ through an input projection matrix $B \in \mathbb{R}^{N_{\text{input-drv}} \times d_{\text{in}}}$. The free population receives no external input. The hidden state evolves according to Eq. 1:

$$h_{i,t+1} = (1-\alpha)\, h_{i,t} + \alpha\Big(\textstyle\sum_{j=1}^{N} W_{ij}\, \phi(h_{j,t}) + \sum_{k=1}^{d_{\text{in}}} B_{ik}\, u_{k,t} + b_i\Big), \tag{S5}$$

After this deterministic recurrent update, we apply the firing-rate-dependent noise in Eq. S4 to each hidden unit. Predictions are read out from the sensory-driven population using a linear readout projection matrix: $\hat{o}_t = C\, h_t^{\text{sens-drv}}$. The network is trained to minimize the error of predicting the next observation: $\sum_t \|\hat{o}_t - o_{t+1}\|_2^2$.

S.3 Simulating sensory observations

Boundary-aware sensory input. We simulate sensory observations using a method similar to [11]. As they suggest, animals receive sensory experiences that change smoothly with location. To model this, we use random Gaussian fields for each sensory channel. Along each sensory channel, the signal varies smoothly across spatial locations, generating a smooth sensory response map of dimension $W \times H$, where W and H are the dimensions of the environment in pixels, respectively. We suppose that an animal has multiple such smooth spatial channels; together, these channels construct a sensory response map of dimension $d_{\text{sens}} \times W \times H$, where the sensory input at each location is defined by its location-specific sensory vector. The authors of [11] first initialize each sensory channel's spatial response map as a random Gaussian field, and then convolve the field with a 2D Gaussian smoothing kernel where the kernel width is σ. However, this method does not respect environmental boundaries. Near a boundary, if σ is larger than the spatial width of a wall, then the 2D Gaussian smoothing kernel can blend sensory observations across the wall.

To resolve the above issue, we replace Gaussian smoothing with a boundary-aware diffusion process that runs for n_{iter} steps. At each step, we apply a 3×3 box filter at each location that averages the sensory responses at neighboring locations that lie in free space. This prevents smoothing across walls and respects environmental boundaries. We set $n_{\text{iter}} = \lceil 1.5\sigma^2 \rceil$ because since one step of a 3×3 box filter has variance $2/3$ along each spatial dimension; so after n steps the effective Gaussian width is $\sqrt{2n/3}$. This gives $n \approx 1.5\sigma^2$ for a smoothing width σ.

Populations. We use four sensory input populations with different smoothing widths $\sigma \in \{6, 8, 10, 12\}$ cm, each containing $N_{\text{cell}} = 256$ cells, giving a total sensory dimension of $d_{\text{obs}} = 1024$. Using multiple smoothing widths allows the sensory observations to contain both fine- and coarse-scale spatial features.

S.4 Trajectory generation

Trajectory generation. We simulate animal trajectories as a smooth random walk. At each timestep t, the agent maintains a current speed v_t and unit movement direction $\mathbf{d}_t$, together with a target speed v_t^* and target direction $\mathbf{d}_t^*$ unit vector. The target variables represent the speed and direction that the agent is currently trying to move toward. Instead of choosing a completely new velocity at every timestep, the agent only occasionally resamples these targets: v_t^* is redrawn with probability p_v and $\mathbf{d}_t^*$ is redrawn with probability p_d. This way, the agent tends to continue moving in a similar way for several timesteps before changing its intended motion. Speed targets are sampled from a log-normal distribution with mean μ_{spd} and standard deviation σ_{spd} (speed is positive). Directions are sampled as random unit vectors in $\mathbb{R}^2$ or $\mathbb{R}^3$ depending on the simulation environment. To

avoid abrupt changes when a new target is sampled, the current speed and direction are gradually updated toward their targets using exponential moving averages: $v_{t+1} = (1 - \alpha_v)v_t + \alpha_v v_t^*$, and $\mathbf{d}_{t+1} = \text{normalize}((1 - \alpha_d)\mathbf{d}_t + \alpha_d \mathbf{d}_t^*)$. Here, α_v and α_d control how quickly the simulated agent adapts to the newly sampled target speed and direction.

Boundary avoidance. To prevent the agent from running into walls or getting stuck near boundaries, we apply a soft boundary avoidance near walls. Before trajectory generation, we compute a distance-to-wall map $d(x)$, which gives the distance from each location x to the nearest wall and a local wall-normal direction $\mathbf{n}(x)$ which points away from the wall. We also define an avoidance strength $c(x)$ which is large near walls and decays as the agent moves farther away from them. When the agent is far from any wall, its movement direction $\mathbf{d}_t$ is unchanged. When the agent is close to a wall and moving towards it, we compute the wall-parallel direction by removing the component of $\mathbf{d}_t$ that points into the wall. We then blend the current direction with this wall-parallel direction, with a blending weight β_t that increases when the agent is closer to the wall and when its speed is larger. This softly turns the agent along the boundary instead of abruptly reflecting it, producing smooth wall-following trajectories in both 2D and 3D environments.

Training along long trajectories. Previous RNN studies [8, 11, 13] train their models with backpropagation through time (BPTT) on short trajectory segments, typically around 20–40 simulation steps or approximately 1–2 s. In their setup, the network is updated using inputs sampled from short trajectory fragments; the simulated agent is relocated to a new position after each update. The authors in [8] initialize the network hidden state with a small multilayer perceptron after each location reset, while the authors in [11] initialize the hidden state to zero. Our model is also trained with BPTT. But we realized that such reset-based training schemes do not allow studying place-cell and grid-cell behavior over longer continuous trajectories. This is especially true in experiments such as the hairpin maze. We resolve this issue by detaching the last hidden state from the previous training segment and use it to initialize the first hidden state of the current training segment. This is the appropriate way to implement truncated BPTT and it allows the model to be trained on short segments while preserving continuous trajectory dynamics across updates.

S.5 Sensory masking

As discussed in Section S.1, we simulate noisy sensory inputs by masking parts of it by the mean sensory response. Masking is applied independently across the batch, timesteps and sensory dimensions. At each timestep, each sensory dimension is independently masked with probability r_{mask}, where r_{mask} denotes the masking ratio. Masked entries are replaced by the corresponding cell's mean firing rate, computed over the arena, while unmasked entries keep their original sensory value. Using the per-cell mean rather than zero prevents the network from treating masked entries as an artificial zero-valued cue.

S.6 Collecting ratemaps during training

Experimentally measured ratemaps are obtained while the animal explores actively. Thus, instead of freezing the model and evaluating it in a separate testing phase, we continuously compute the firing rate of different cells using the trajectories sampled during training. At each optimization step, we collect hidden states from all batches and timesteps, together with the corresponding agent coordinates. The coordinates are rounded to the nearest simulation bin ($1\,\text{cm} \times 1\,\text{cm}$). For each visited bin, we compute each hidden unit's mean firing rate over all visits to that bin. This gives a ratemap estimate over the subset of spatial bins visited during the current optimization step. To aggregate ratemaps over training, we maintain one running ratemap for each hidden unit. At each update step, only the spatial bins visited by the current trajectories are updated. For those bins, the ratemap values from previous updates are combined with the average firing rates computed at the current update step using an exponential moving average with decay $\gamma = 0.995$. Spatial bins that are not visited keep their previous values and bins that have never been visited are marked as undefined. This allows ratemaps to be accumulated smoothly throughout training while giving slightly higher weight to more recent activity. With $\gamma = 0.995$, the ratemap retains an effective memory of approximately 200 update steps for regularly visited bins. When saving the ratemaps, we average each 3×3 block of simulation bins, so each ratemap bin spans $3\,\text{cm}$ along each axis.

Table S1: Default training and architectural parameters.

Parameter	Value	Description
$N_{\text{input-drv}}$	1024	Sensory-driven population size
$N_{\text{rec-drv}}$	1024	Free population size
$\phi(\cdot)$	softplus	Firing-rate nonlinearity
EI ratio	0.8 / 0.2	Excitatory / inhibitory fraction
α	0.5	Default leak rate (sigmoid-gated)
σ_n	0.8	Default firing-rate noise level
b_i	0 (fixed)	Bias term in Eq. 1
η	10^{-3}	Homeostatic scaling exponent
optimizer	AdamW	
learning_rate	$5 \cdot 10^{-4}$	
batch_size	256	Each batch is one batch of trajectories
traj_duration	10	Backpropagation window
gradient_clip	1.0	L2 norm cap on parameter gradients
Δt	1/20 s	Simulation timestep
$\mu_{\text{spd}}, \sigma_{\text{spd}}$	40, 20 cm/s	Log-normal speed mean and std
α_v, α_d	0.8, 0.2	EMA smoothing for speed and direction
p_v, p_d	0.2, 0.1	Per-step target switch probabilities
σ_{cell}	$\{6, 8, 10, 12\}$	Sensory cells smoothing widths
N_{cell}	256 per population	Cells per sensory population
r_{mask}	0.2	Default sensory mask ratio

S.7 Default training parameters

Unless otherwise stated, all reported runs use the parameters in Table S1. Per-experiment overrides are listed in their respective sections.

S.8 Cell classification

We classify cells into three types: *place*, *grid* or *other spatial*, based on per-cell scores computed from each ratemap.

Classifying activated spatial cells. Since both grid cells and place cells are strongly modulated by an animal's spatial locations, we first filter out cells that are active and have spatial information content above 0.2. We define a cell as active if it has a mean firing rate $\bar{r}_i \leq 0.1$. The spatial information content is

$$\text{SIC}_i = \sum_{\mathbf{x}} p(\mathbf{x}) \frac{R_i(\mathbf{x})}{\bar{r}_i} \log_2 \frac{R_i(\mathbf{x})}{\bar{r}_i}, \tag{S6}$$

where $p(\mathbf{x})$ is the empirical occupancy of spatial bin $\mathbf{x}$. Cells with $\text{SIC}_i > 0.2$ are treated as spatial cells.

Place-cell classification. We next test whether selected spatial cells are place cells. Previous studies often use SIC alone to identify place cells. However, because our model contains place cells, grid cells, and other spatially tuned cells within the same hidden layer, SIC alone may not be sufficient for robust classification. We therefore add two firing-field locality criteria for place-cell classification. Specifically, a place cell should contain only a small number of spatially localized firing fields, rather than many repeated peaks. In addition, its firing should be concentrated to a small spatial region. To measure this, we first threshold each ratemap at 0.3 of its maximum firing rate. After thresholding, each connected group of spatial bins above the threshold is identified as a firing field. We classify a cell as a place cell if it has fewer than 3 such firing fields and if its largest field contains at least 35% of the cell's total firing mass.

Spatial autocorrelogram. To classify grid cells, previous studies have used autocorrelograms to reveal the characteristic spatial periodicity of grid-cell firing [1]. In this study, we use the same spatial autocorrelogram computation method as that of [1]. Specifically, for each ratemap $R_i(\mathbf{x})$, we compute a 2D spatial autocorrelogram to measure whether the firing fields form a regular periodic structure. For each spatial lag $\boldsymbol{\tau} = (\tau_x, \tau_y)$, we compare the ratemap with a shifted copy of itself, $R_i(\mathbf{x} - \boldsymbol{\tau})$. The correlation is computed only over spatial bins where both $R_i(\mathbf{x})$ and $R_i(\mathbf{x} - \boldsymbol{\tau})$ are valid, which corrects for edge effects and unvisited locations. Following this standard sample-correlation formulation, the autocorrelation at lag $\boldsymbol{\tau} = (\tau_x, \tau_y)$ for cell i, with ratemap $R \equiv R_i$, is

$$A(\boldsymbol{\tau}) = \frac{n \sum R(\mathbf{x})R(\mathbf{x}-\boldsymbol{\tau}) - \sum R(\mathbf{x}) \sum R(\mathbf{x}-\boldsymbol{\tau})}{\sqrt{n \sum R(\mathbf{x})^2 - \left(\sum R(\mathbf{x})\right)^2}\sqrt{n \sum R(\mathbf{x}-\boldsymbol{\tau})^2 - \left(\sum R(\mathbf{x}-\boldsymbol{\tau})\right)^2}}, \tag{S7}$$

where all sums are taken over the n spatial bins for which both $R(\mathbf{x})$ and $R(\mathbf{x} - \boldsymbol{\tau})$ are valid. Lags with too few overlapping valid bins are not evaluated. The resulting autocorrelogram is shifted so that the zero-lag term is centered, producing a $(2H - 1) \times (2W - 1)$ map for an $H \times W$ ratemap.

Grid score. The "grid score" measures whether a cell has the six-fold rotational symmetry characteristic of grid cells. Following the standard grid-score procedure [48], we take a centered crop of the autocorrelogram computed above with radius $r = 60$ bins. We exclude the small central peak around zero lag, since this peak appears for all cells, and also exclude a narrow edge band to avoid boundary artifacts. We then compute the Pearson correlation between the autocorrelogram and a rotated copy of itself. Let ρ_θ denote this correlation after rotation by θ degrees. The grid score is defined as

$$\mathrm{GS}_i = \min(\rho_{60}, \rho_{120}) - \max(\rho_{30}, \rho_{90}, \rho_{150}). \tag{S8}$$

A hexagonal grid should have high correlation after 60° and 120° rotations, but lower correlation after 30°, 90°, and 150° rotations.

Note that we classify grid cells after place-cell classification. This ordering is unlikely to remove valid grid cells, because our place-cell criterion requires fewer than 3 firing fields, whereas a valid grid cell should contain multiple periodically repeated fields. Among the remaining spatially tuned cells, we classify a cell as a grid cell if it passes the grid-score threshold. We use $\mathrm{GS}_i > 0.3$ for all analyses, except for the parameter-sweep result in Figure 2F. For the parameter sweep, we use a more lenient threshold of $\mathrm{GS}_i > 0.1$, so that the summary better reflects how grid-like periodic patterns change across the swept parameters. We also require $\min(\rho_{60}, \rho_{120}) > 0$, ensuring that a positive grid score is supported by positive correlations at the grid-symmetric rotations. Finally, we require at least 4 firing fields, which prevents single-peaked or weakly localized spatial cells from being classified as grid cells.

Other spatial cells. Spatial cells that do not satisfy either the place or grid criteria are labelled *other spatial*. These are cells with structured but non-localized and non-hexagonal spatial tuning (e.g., stripe-like, multi-peaked irregular fields, or border-like).

S.9 Shuffle control for grid-cell classification

We classify grid cells using their grid scores, but irregular spatial firing patterns may also receive positive scores. We therefore test whether the identified cells retain high grid scores when their activity is no longer matched to the positions where it was recorded. We evaluate the $1{,}024$ recurrently driven cells in each of 10 independently trained models using 16 independent trajectories of 300 seconds each. For each candidate grid cell, we shift its recorded activity by a random time interval relative to the agent's position. Activity shifted past the end of the recording is placed at the beginning. This preserves the order of the activity while changing which activity corresponds to each position. We then reconstruct the ratemap and recalculate the grid score, repeating this procedure 500 times per cell.

For each cell, we use the 95th percentile of its 500 shuffled scores as a threshold, so that only 5% of shuffled scores exceed it. Across the 10 models, all $1{,}825$ candidate grid cells with $\mathrm{GS} > 0.1$ have original scores above their corresponding thresholds, including all 516 cells with $\mathrm{GS} > 0.3$. These thresholds range from 0.020 to 0.066, and every original score exceeds its threshold by at least 0.0487.

S.10 Comparison of grid spacing and field size with experimental recordings

We use the same analysis pipeline to compare 516 grid cells from 10 independently trained models with 13 grid cells from the publicly available recordings of Hafting et al. [1]. We measure grid spacing from the six surrounding peaks in each spatial autocorrelogram and define field area as the central connected region with correlation values above 0.2. We calculate field radius as the radius of a circle with the same area, and compute spacing / field radius separately for each cell. Values in Table S2 are medians with interquartile ranges.

Table S2: Grid spacing and field size in the model and experimental recordings.

Our Model (516 cells from 10 models)	**Hafting et al. (13 cells)**
Grid spacing: 70.31 cm (68.65–71.48)	**Grid spacing:** 42.82 cm (41.53–49.54)
Field area: 1,233 cm^2 (1,179–1,269)	**Field area:** 477 cm^2 (423–549)
Spacing / field radius: 3.55 (3.48–3.62)	**Spacing / field radius:** 3.75 (3.59–3.77)

For reference, Giocomo et al. [49] reported spacing / field radius ratios of 3.26 ± 0.07 in control mice and 3.32 ± 0.06 in HCN1 knockout mice (mean $\pm$ SEM).

S.11 Parameter sweep details

Sweep grid. The parameter sweep in the main text (Figure 2F) covers a $10 \times 10 \times 10$ grid over the leak rate, firing-rate noise level and sensory mask ratio:

$$\alpha \in \{0.1, 0.2, \ldots, 1.0\}, \quad \sigma_n \in \{0, 0.05, \ldots, 0.45\}, \quad r_{\text{mask}} \in \{0, 0.1, \ldots, 0.9\}. \tag{S9}$$

For each configuration, we train the network for 200,000 time steps in the $220\,\text{cm} \times 220\,\text{cm}$ square arena while holding all other parameters at their default values (Table S1). We use a different random seed for each training condition to test robustness across stochastic components of the setup, including the sensory response fields, trajectories, and network initialization. We use only one random seed for each configuration (different for each configuration). When summarizing results along each marginal axis, each value aggregates over 100 sweep conditions, providing implicit averaging over the other parameters and random seeds.

S.12 Two-room wall removal experiment

In this experiment, two $180 \times 150\,\text{cm}^2$ rectangular rooms are placed side-by-side and separated by a thin internal wall of thickness equal to twice the simulator's 5-cm border, giving a merged arena of size $180 \times 310\,\text{cm}^2$. We train the network for 60,000 time steps in the left room, 60,000 in the right room, and 80,000 in the merged arena after wall removal.

1. Room 0 (left sub-room). The agent is spawned and confined to the left compartment.
2. Room 1 (right sub-room). The agent is respawned in the right compartment.
3. Merged arena. The internal wall is removed, and the agent is allowed to move freely through the full arena.

At initialization, we first sample a random Gaussian noise field. We then generate the sensory response maps for the two rooms using the boundary-aware diffusion smoothing method described in Section S.3. Because this diffusion process respects environmental boundaries, sensory values are not smoothed across the wall separating the two rooms. As a result, the sensory response maps on the two sides of the internal wall can differ sharply. During the first two phases, the agent is spawned in the corresponding room, and its trajectory is constrained to be within that room because the two rooms are not connected.

After the agent has explored both rooms, we remove the internal wall. Starting from the same initial Gaussian noise field, we re-apply the diffusion-based smoothing process on the merged arena. After this re-smoothing, sensory responses near the original wall location become smoothly connected across the two formerly separated rooms. Since the diffusion process only acts locally, sensory

responses at locations far from the removed wall are expected to remain similar to their values before wall removal.

Across all three phases, we use the parameters in Table S1, with $\alpha = 0.5$, $\sigma_n = 0.8$, and $r_{\text{mask}} = 0$. We maintain a ratemap aggregator over the full merged arena. During phases 0 and 1, only the bins visited in the active room are updated, while bins in the other room remain unchanged. At the beginning of phase 2, we reset the ratemap aggregation so that the merged-arena ratemaps reflect only the post-wall-removal experience.

In a separate time-course analysis, we compare intermediate ratemaps with a pre-removal reference formed by joining the two room-specific maps and a final merged-arena reference. For cells with GS > 0.1 in both rooms, we compute spatial correlations within each room, excluding the former wall region, and average their Fisher z-transformed values across rooms and cells. At $2{,}000$ time steps after wall removal, similarity to the pre-removal reference is higher than to the merged reference (0.912 versus 0.853). By $5{,}000$ time steps, this ordering reverses (0.824 versus 0.916), indicating a shift toward the merged representation between these sampled time points.

S.13 Hairpin maze experiment

The hairpin arena contains 10 parallel corridors connected by alternating turn gaps, forming a continuous serpentine path. Its bounding box is matched to the open arena, so the same raw sensory field can be evaluated in both environments after applying boundary-aware diffusion smoothing. To construct the sensory inputs, we first generate one raw Gaussian noise field for each sensory population. We then apply the boundary-aware diffusion process separately in the open arena and in the hairpin maze. Because the raw field is shared, corresponding locations have similar sensory values away from the inserted walls. In the hairpin maze, however, diffusion is blocked by corridor walls, so sensory similarity follows the corridor structure rather than passing through walls.

In the open-arena phases, trajectories are generated by the default smooth random-walk generator. In the hairpin phases, we use a scripted serpentine controller that drives the agent through the corridors one lap at a time: left-to-right in the second phase (Figure 3B, MazeLR) and right-to-left in the third phase (Figure 3B, MazeRL). The hidden state is reset to zero at each lap boundary, so successive laps form independent trials. At every step, speed is sampled from the same log-normal distribution used by the open-arena controller. We also add small random lateral offsets within each corridor, together with a perpendicular drift, so that the agent samples the full corridor width rather than only following the centerline.

Following [33], we train the model across four phases: open arena, hairpin forward, hairpin backward, and open arena again. During each phase, we update the network for $100,000$ steps. We reset the accumulated ratemap whenever we shift the training phase, so each phase's ratemaps reflect only cell activity collected during that phase.

S.14 Connected-rooms experiment

We use two $120 \times 120\,\text{cm}^2$ compartments separated by an $8\,\text{cm}$ dividing wall, with a $248 \times 70\,\text{cm}^2$ return corridor along the north side. Each compartment connects to the corridor through a single $80\,\text{cm}$ opening centered on its north wall, so the corridor and the two compartments form one continuous walkable region. The fully connected arena is treated as a single environment during both sensory-field generation and training. Therefore, the sensory field changes smoothly across the corridor. We train the network with $800,000$ update steps, allowing grid-cell-like representations to stabilize while also tracking their drift over training.

S.15 Three-dimensional traversal experiments

For the 3D experiments, we use a cubic arena of size $97 \times 97 \times 97$ bins, with a 5-voxel border giving a 107^3 ambient map. The four sensory populations use 3D smoothing widths $\sigma \in \{8, 10, 12, 14\}$, each containing 256 cells. The boundary-aware diffusion process described in Section S.3 is applied with a volumetric $3 \times 3 \times 3$ kernel. The motion input is 4-dimensional, consisting of a 3D unit movement direction and a scalar speed.

In the random-fly traversal experiment, the agent moves freely through the volume using the same smooth trajectory generator and boundary-avoidance scheme as in 2D. We use $\mu_{\text{spd}} = 30$, $\sigma_{\text{spd}} = 10$,

and $\alpha_v = 0.5$, and train the model for 800,000 steps with firing-rate noise $\sigma_n = 0.8$. Under this traversal, place-like cells develop localized 3D firing fields, while grid-like cells develop locally ordered volumetric firing fields, consistent with the partially ordered 3D grid patterns reported in flying bats.

S.16 Robustness of co-emergence

S.16.1 Region ratio

The default network uses $N_{\text{sens-drv}} = N_{\text{rec-drv}} = 1024$. To examine whether this exact split affects co-emergence, we trained four additional models with different sensory-driven/recurrent-driven ratios: $(N_{\text{sens-drv}}, N_{\text{rec-drv}}) \in \{(512, 1536), (768, 1280), (1280, 768), (1536, 512)\}$, while holding the total number of hidden units fixed at $N = 2048$.

Across all four models, both place cells and grid cells emerged consistently. The corresponding numbers of classified place/grid cells were $263/68$, $302/43$, $307/94$, and $313/12$, respectively. Although our main experiments were conducted in the $1024/1024$ setting for purposes of clarity, this experiment shows that co-emergence is not specific to that case. The reduced grid-cell count in the $1536/512$ split suggests that having too few recurrent-driven units can limit grid-cell emergence, but the qualitative presence of both cell types remains consistent across the tested partitions.

S.16.2 Excitatory–inhibitory ratio

We also test how the proportion of excitatory and inhibitory neurons affects the co-emergence of grid and place cells. With excitatory-to-inhibitory ratios of 70:30, 80:20, and 90:10, we identify 44, 48, and 5 grid cells with grid scores above 0.3, respectively, together with 327, 302, and 302 place cells. Both cell types emerge under all three ratios, although grid cells are substantially less numerous at 90:10, while place-cell counts remain similar.

S.16.3 Trajectory and speed controls

We also test how movement speed and trajectory structure affect grid- and place-cell emergence. Grid cells emerge across mean speeds of 10–60 cm/s, with slower movement producing more closely spaced grid fields. When trajectory smoothing is removed, movement direction changes abruptly at each turning event, and grid cells no longer emerge in the tested condition. In contrast, randomly repositioning the agent every 10 s still produces 118 grid cells with grid scores above 0.1 and 294 place cells. These results suggest that frequent abrupt turns and occasional interruptions of the trajectory have different effects on grid-cell emergence.

S.16.4 Homeostatic synaptic scaling

Setting $\eta = 0$ in Eq. S3 removes the column-wise outgoing-budget rescaling. Over short training horizons ($\leq$ 50,000 steps), grid- and place-like responses still emerge qualitatively. However, over longer training horizons, the recurrent dynamics gradually drift: units that initially exhibit hexagonally arranged firing fields lose contrast over time, with firing-field centers becoming progressively weaker until the periodic structure is no longer visually apparent. We also found that, without homeostatic scaling, switching to a new room can produce infinite losses, possibly because the network has overfit to the preceding room. Homeostatic scaling therefore acts as a soft stabilizer that preserves the co-emergent operating regime over extended training, rather than being strictly required for the initial formation of grid- or place-like responses.

We also vary the scaling strength, using $\eta = 10^{-4}, 10^{-3}, 10^{-2}$ in otherwise identical training runs with 200,000 update steps. These settings yield 17, 43, and 0 grid cells with grid scores above 0.3, respectively, together with 336, 302, and 262 place cells. Reducing the scaling strength from the default $\eta = 10^{-3}$ preserves both cell types, although with fewer grid cells. At $\eta = 10^{-2}$, no cells meet the grid-cell criterion, while place cells remain, suggesting that grid-cell formation is sensitive to strong synaptic scaling.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Yes, in our paper, we introduce a unified recurrent network model that instantiates Dale's Law, and is trained to predict the next sensory observation from masked previous sensory observations and egocentric motion. We found that training this model successfully reproduces several key phenomenologies that were observed on biological place and grid cells. Our paper, as well as our abstract and introduction, has accurately reflected these contributions.

   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: We have included a detailed discussion of limitations in the discussion section.

   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: This 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: We have included a detailed discussion of the precise setup of our model, including how environments, trajectories, motion inputs, and sensory observations are simulated. We also report the network architecture, E/I composition, nonlinearities, leak and gating settings, noise model, sensory masking procedure, optimizer, learning rate, batch size, gradient truncation horizon, gradient clipping, simulation timestep, trajectory statistics, and all default training hyperparameters in Table S1. For the main experiments, we describe the parameter sweeps, random-seed protocol, cell-type classification metrics, spatial-information and grid-score analyses, and the environment-specific evaluation procedures. Together, these details specify the experimental pipeline sufficiently for an independent researcher to reproduce the main results and assess the claims made 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 code is available at https://github.com/grasp-lyrl/grid_and_place. The repository includes the simulation and training scripts, experiment configurations, and instructions for running the experiments. The training data are generated procedurally. The supplement describes how we generate trajectories and sensory observations, train the network, and evaluate the resulting spatial representations. An overview of the experiments is available on the project page at https://zhaozewang.github.io/grid-and-place/.

   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: We specify the experimental setup in the main text and supplement, including the simulated environments, trajectory generation, sensory observations, model architecture, Dale's constraints, optimizer, learning rate, batch size, training horizon, gradient clipping, noise and masking settings, parameter sweeps, random seeds, and evaluation metrics.

   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: Our main results are supported by a sweep over 1,000 training conditions with different random seeds, showing that the observed co-emergence patterns are not seed-specific. We also report the mean number of emerged grid cells and place cells, together with their standard deviations, in the main text and supplement where appropriate.

   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: We report the compute resources used for the experiments in the supplemental materials. We trained all experiments in Sections 3 and 4 on a single NVIDIA RTX 5090 GPU, and ran the 1,000-condition sweep on a workstation with dual NVIDIA RTX 4090 GPUs. We also specify the batch size, training horizon, model size, and simulation settings needed to estimate memory and runtime requirements.

   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: We have reviewed the NeurIPS Code of Ethics and the research conforms to it. The paper uses procedurally generated simulation data, does not involve human subjects, crowdsourcing, private or sensitive data, or high-risk model release, and does not require any deviation from the Code of Ethics.

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: [Yes]

Justification: The paper discusses the broader impacts of the work as foundational research on spatial representation learning and neural computation. Potential positive impacts include improving mechanistic understanding of biological navigation and informing more structured, data-efficient representations in artificial systems. Potential negative impacts are indirect, since the work is not a deployed system and uses only procedurally generated simulation data.

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 does not release data or models with high risk for misuse.

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: [N/A]

    Justification: All experimental data are generated procedurally from our own simulation setup, and the model is trained from scratch for the experiments reported in the paper.

    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: [Yes]

    Justification: We release the simulation and training code at `https://github.com/grasp-lyrl/grid_and_place`. The repository README provides installation instructions and commands for running the experiments.

    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 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 human subjects.

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 important, original, or non-standard components of the core method. They were used only for text polishing and standard analysis or visualization code assistance, with all outputs manually reviewed by the authors.

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.