On Path Integration of Grid Cells: Group Representation and Isotropic Scaling

Ruiqi Gao[1*]
ruiqigao@ucla.edu

Jianwen Xie[2]
jianwen@ucla.edu

Xue-Xin Wei[3]
weixx@utexas.edu

Song-Chun Zhu[1,4,5]
sczhu@stat.ucla.edu

Ying Nian Wu[1]
ywu@stat.ucla.edu

[1]Department of Statistics, UCLA [2]Cognitive Computing Lab, Baidu Research
[3]Department of Neuroscience, UT Austin [4]Department of Computer Science, UCLA
[5]Beijing Institute for General Artificial Intelligence (BIGAI)

Abstract

Understanding how grid cells perform path integration calculations remains a fundamental problem. In this paper, we conduct theoretical analysis of a general representation model of path integration by grid cells, where the 2D self-position is encoded as a higher dimensional vector, and the 2D self-motion is represented by a general transformation of the vector. We identify two conditions on the transformation. One is a group representation condition that is necessary for path integration. The other is an isotropic scaling condition that ensures locally conformal embedding, so that the error in the vector representation translates conformally to the error in the 2D self-position. Then we investigate the simplest transformation, i.e., the linear transformation, uncover its explicit algebraic and geometric structure as matrix Lie group of rotation, and explore the connection between the isotropic scaling condition and a special class of hexagon grid patterns. Finally, with our optimization-based approach, we manage to learn hexagon grid patterns that share similar properties of the grid cells in the rodent brain. The learned model is capable of accurate long distance path integration. Code is available at `https://github.com/ruiqigao/grid-cell-path`.

1 Introduction

Imagine walking in the darkness. Purely based on the sense of self-motion, one can gain a sense of self-position by integrating the self motion - a process often referred to as path integration [10, 14, 21, 15, 27]. While the exact neural underpinning of path integration remains unclear, it has been hypothesized that the grid cells [21, 17, 40, 24, 23, 12] in the mammalian medial entorhinal cortex (mEC) may be involved in this process [20, 30, 22]. The grid cells are so named because individual neurons exhibit striking firing patterns that form hexagonal grids when the agent (such as a rat) navigates in a 2D open field [18, 21, 16, 6, 34, 5, 7, 11, 29, 1]. The grid cells also interact with the place cells in the hippocampus [28]. Unlike a grid cell that fires at the vertices of a lattice, a place cell often fires at a single (or a few) locations.

The purpose of this paper is to understand how the grid cells may perform path integration calculations. We study a general optimization-based representational model in which the 2D self-position is

[*] The author is now a Research Scientist at Google Brain team.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

represented by a higher dimensional vector and the 2D self-motion is represented by a transformation of the vector. The vector representation can be considered position encoding or position embedding, where the elements of the vector may be interpreted as activities of a population of grid cells. The transformation can be realized by a recurrent network that acts on the vector. Our focus is to study the properties of the transformation.

Specifically, we identify two conditions for the transformation: a group representation condition and an isotropic scaling condition, under which we demonstrate that the local neighborhood around each self-position in the 2D physical space is embedded conformally as a 2D neighborhood around the vector representation of the self-position in the neural space.

We then investigate the simplest special case of the transformation, i.e., linear transformation, that forms a matrix Lie group of rotation, under which case we show that the isotropic scaling condition is connected to a special class of hexagonal grid patterns. Our numerical experiments demonstrate that our model learns clear hexagon grid patterns of multiple scales which share observed properties of the grid cells in the rodent brain, by optimizing a simple loss function. The learned model is also capable of accurate long distance path integration.

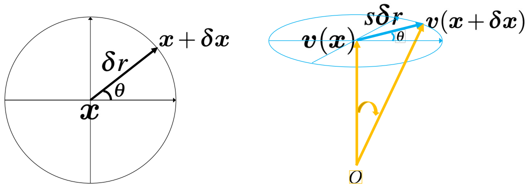
Contributions. Our work contributes to understanding the grid cells from the perspective of representation learning. We conduct theoretical analysis of (1) general transformation for path integration by identifying two key conditions and a local conformal embedding property, (2) linear transformation by revealing the algebraic and geometric structure and connecting the isotropic scaling condition and a special class of hexagon grid patterns, and (3) integration of linear transformation model and linear basis expansion model via unitary group representation theory. Experimentally we learn clear hexagon grid patterns that are consistent with biological observations, and the learned model is capable of accurate path integration.

2 General transformation

2.1 Position embedding

Consider an agent (e.g., a rat) navigating within a 2D open field. Let $\boldsymbol{x} = (x_1, x_2)$ be the self-position of the agent. We assume that the self-position $\boldsymbol{x}$ in the 2D physical space is represented by the response activities of a population of d neurons (e.g., $d = 200$), which form a vector $\boldsymbol{v}(\boldsymbol{x}) = (v_i(\boldsymbol{x}), i = 1, ..., d)^\top$ in the d-dimensional "neural space", with each element $v_i(\boldsymbol{x})$ representing the firing rate of one neuron when the animal is at location $\boldsymbol{x}$.

$\boldsymbol{v}(\boldsymbol{x})$ can be called position encoding or position embedding. Collectively, $(\boldsymbol{v}(\boldsymbol{x}), \forall \boldsymbol{x})$ forms a *codebook* of $\boldsymbol{x} \in \mathbb{R}^2$, and $(\boldsymbol{v}(\boldsymbol{x}), \forall \boldsymbol{x})$ is a *2D manifold* in the d-dimensional neural space, i.e., globally we embed $\mathbb{R}^2$ as a 2D manifold in the neural space. Locally, we identify two conditions under which the 2D local neighborhood around each $\boldsymbol{x}$ is embedded *conformally* as a 2D neighborhood around $\boldsymbol{v}(\boldsymbol{x})$ with a scaling factor. See Fig. 1. As shown in Section 3.3, the conformal embedding is connected to the hexagon grid patterns.


$\boldsymbol{x} + \delta\boldsymbol{x}$
δr
θ
$\boldsymbol{x}$
$s\delta r$
$\boldsymbol{v}(\boldsymbol{x} + \delta\boldsymbol{x})$
$\boldsymbol{v}(\boldsymbol{x})$
θ
O

(a) physical space

(b) neural space

Figure 1: The local 2D polar system around self-position $\boldsymbol{x}$ in the 2D physical space (a) is embedded conformally as a 2D polar system around vector $\boldsymbol{v}(\boldsymbol{x})$ in the d-dimensional neural space (b), with a scaling factor s (so that δr in the physical space becomes $s\delta r$ in the neural space while the angle θ is preserved).

2.2 Transformation and path integration

At self-position $\boldsymbol{x}$, if the agent makes a self-motion $\Delta\boldsymbol{x} = (\Delta x_1, \Delta x_2)$, then it moves to $\boldsymbol{x} + \Delta\boldsymbol{x}$. Correspondingly, the vector representation $\boldsymbol{v}(\boldsymbol{x})$ is transformed to $\boldsymbol{v}(\boldsymbol{x} + \Delta\boldsymbol{x})$. The general form of the transformation can be formulated as:

$$\boldsymbol{v}(\boldsymbol{x} + \Delta\boldsymbol{x}) = F(\boldsymbol{v}(\boldsymbol{x}), \Delta\boldsymbol{x}). \tag{1}$$

The transformation $F(\cdot, \Delta\boldsymbol{x})$ can be considered a representation of $\Delta\boldsymbol{x}$, which forms a 2D additive group. We call Eq. (1) the *transformation model*. It can be implemented by a recurrent network to derive a path integration model: if the agent starts from $\boldsymbol{x}_0$, and makes a sequence of moves $(\Delta\boldsymbol{x}_t, t = 1,...,T)$, then the vector is updated by $\boldsymbol{v}_t = F(\boldsymbol{v}_{t-1}, \Delta\boldsymbol{x}_t)$, where $\boldsymbol{v}_0 = \boldsymbol{v}(\boldsymbol{x}_0)$, and $t = 1,...,T$.

2.3 Group representation condition

The solution to the transformation model (Eq. (1)) should satisfy the following condition.

Condition 1. *(Group representation condition)* $(\boldsymbol{v}(\boldsymbol{x}), \forall\boldsymbol{x})$ *and* $(F(\cdot,\Delta\boldsymbol{x}), \forall\Delta\boldsymbol{x})$ *form a representation of the 2D additive Euclidean group* $\mathbb{R}^2$ *in the sense that*

$$F(\boldsymbol{v}(\boldsymbol{x}), 0) = \boldsymbol{v}(\boldsymbol{x}),\ \forall\boldsymbol{x}; \tag{2}$$

$$F(\boldsymbol{v}(\boldsymbol{x}), \Delta\boldsymbol{x}_1 + \Delta\boldsymbol{x}_2) = F(F(\boldsymbol{v}(\boldsymbol{x}), \Delta\boldsymbol{x}_1), \Delta\boldsymbol{x}_2),\ \forall\boldsymbol{x}, \Delta\boldsymbol{x}_1, \Delta\boldsymbol{x}_2. \tag{3}$$

$(F(\cdot,\Delta\boldsymbol{x}), \forall\Delta\boldsymbol{x})$ is a Lie group of transformations acting on the codebook manifold $(\boldsymbol{v}(\boldsymbol{x}), \forall\boldsymbol{x})$.

The reason for (2) is that if $\Delta\boldsymbol{x} = 0$, then $F(\cdot, 0)$ should be the identity transformation. Thus the codebook manifold $(\boldsymbol{v}(\boldsymbol{x}), \forall\boldsymbol{x})$ consists of fixed points of the transformation $F(\cdot, 0)$. If $F(\cdot, 0)$ is furthermore a contraction around $(\boldsymbol{v}(\boldsymbol{x}), \forall\boldsymbol{x})$, then $(\boldsymbol{v}(\boldsymbol{x}), \forall\boldsymbol{x})$ are the attractor points.

The reason for (3) is that the agent can move in one step by $\Delta\boldsymbol{x}_1 + \Delta\boldsymbol{x}_2$, or first move by $\Delta\boldsymbol{x}_1$, and then move by $\Delta\boldsymbol{x}_2$. Both paths would end up at the same $\boldsymbol{x} + \Delta\boldsymbol{x}_1 + \Delta\boldsymbol{x}_2$, which is represented by the same $\boldsymbol{v}(\boldsymbol{x} + \Delta\boldsymbol{x}_1 + \Delta\boldsymbol{x}_2)$.

The group representation condition is a necessary self-consistent condition for the transformation model (Eq. (1)).

2.4 Egocentric self-motion

Self-motion $\Delta\boldsymbol{x}$ can also be parametrized egocentrically as $(\Delta r, \theta)$, where Δr is the displacement along the direction $\theta \in [0, 2\pi]$, so that $\Delta\boldsymbol{x} = (\Delta x_1 = \Delta r \cos\theta, \Delta x_2 = \Delta r \sin\theta)$. The egocentric self-motion may be more biologically plausible where θ is encoded by head direction, and Δr can be interpreted as the speed along direction θ. The transformation model then becomes

$$\boldsymbol{v}(\boldsymbol{x} + \Delta\boldsymbol{x}) = F(\boldsymbol{v}(\boldsymbol{x}), \Delta r, \theta), \tag{4}$$

where we continue to use $F(\cdot)$ for the transformation (with slight abuse of notation). $(\Delta r, \theta)$ form a polar coordinate system around $\boldsymbol{x}$.

2.5 Infinitesimal self-motion and directional derivative

In this subsection, we derive the transformation model for infinitesimal self-motion. While we use $\Delta\boldsymbol{x}$ or Δr to denote finite (non-infinitesimal) self-motion, we use $\delta\boldsymbol{x}$ or δr to denote infinitesimal self-motion. At self-position $\boldsymbol{x}$, for an infinitesimal displacement δr along direction θ, $\delta\boldsymbol{x} = (\delta x_1 = \delta r \cos\theta, \delta x_2 = \delta r \sin\theta)$. See Fig. 1 (a) for an illustration. Given that δr is infinitesimal, for any fixed θ, a first order Taylor expansion of $F(\boldsymbol{v}(\boldsymbol{x}), \delta r, \theta)$ with respect to δr gives us

$$\begin{aligned}\boldsymbol{v}(\boldsymbol{x} + \delta\boldsymbol{x}) &= F(\boldsymbol{v}(\boldsymbol{x}), \delta r, \theta) = F(\boldsymbol{v}(\boldsymbol{x}), 0, \theta) + F'(\boldsymbol{v}(\boldsymbol{x}), 0, \theta)\delta r + o(\delta r)\\ &= \boldsymbol{v}(\boldsymbol{x}) + f_\theta(\boldsymbol{v}(\boldsymbol{x}))\delta r + o(\delta r),\end{aligned} \tag{5}$$

where $F(\boldsymbol{v}(\boldsymbol{x}), 0, \theta) = \boldsymbol{v}(\boldsymbol{x})$ according to Condition 1, and $f_\theta(\boldsymbol{v}(\boldsymbol{x})) := F'(\boldsymbol{v}(\boldsymbol{x}), 0, \theta)$ is the first derivative of $F(\boldsymbol{v}(\boldsymbol{x}), \Delta r, \theta)$ with respect to Δr at $\Delta r = 0$. $f_\theta(\boldsymbol{v}(\boldsymbol{x}))$ is the *directional derivative* of $F(\cdot)$ at self-position $\boldsymbol{x}$ and direction θ.

For a fixed θ, $(F(\cdot, \Delta r, \theta), \forall\Delta r)$ forms a one-parameter Lie group of transformations, and $f_\theta(\cdot)$ is the generator of its Lie algebra.

2.6 Isotropic scaling condition

With the directional derivative, we define the second condition as follows, which leads to locally conformal embedding and is connected to hexagon grid pattern.

Condition 2. *(Isotropic scaling condition) For any fixed* $\boldsymbol{x}$, $\|f_\theta(\boldsymbol{v}(\boldsymbol{x}))\|$ *is constant over* θ.

Let $f_0(\boldsymbol{v}(\boldsymbol{x}))$ denote $f_\theta(\boldsymbol{v}(\boldsymbol{x}))$ for $\theta = 0$, and $f_{\pi/2}(\boldsymbol{v}(\boldsymbol{x}))$ denote $f_\theta(\boldsymbol{v}(\boldsymbol{x}))$ for $\theta = \pi/2$. Then we have the following theorem:

Theorem 1. *Assume group representation condition 1 and isotropic scaling condition 2. At any fixed $\boldsymbol{x}$, for the local motion $\delta\boldsymbol{x} = (\delta r \cos\theta, \delta r \sin\theta)$ around $\boldsymbol{x}$, let $\delta\boldsymbol{v} = \boldsymbol{v}(\boldsymbol{x}+\delta\boldsymbol{x}) - \boldsymbol{v}(\boldsymbol{x})$ be the change of vector and $s = \|f_\theta(\boldsymbol{v}(\boldsymbol{x}))\|$, then we have $\|\delta\boldsymbol{v}\| = s\|\delta\boldsymbol{x}\|$. Moreover,*

$$\delta\boldsymbol{v} = f_\theta(\boldsymbol{v}(\boldsymbol{x}))\delta r + o(\delta r) = f_0(\boldsymbol{v}(\boldsymbol{x}))\delta r \cos\theta + f_{\pi/2}(\boldsymbol{v}(\boldsymbol{x}))\delta r \sin\theta + o(\delta r), \tag{6}$$

where $f_0(\boldsymbol{v}(\boldsymbol{x}))$ and $f_{\pi/2}(\boldsymbol{v}(\boldsymbol{x}))$ are two orthogonal basis vectors of equal norm s.

See Supplementary for a proof and Fig. 1(b) for an illustration. Theorem 1 indicates that the local 2D polar system around self-position $\boldsymbol{x}$ in the 2D physical space is embedded conformally as a 2D polar system around vector $\boldsymbol{v}(\boldsymbol{x})$ in the d-dimensional neural space, with a scaling factor s (our analysis is local for any fixed $\boldsymbol{x}$, and s may depend on $\boldsymbol{x}$). Conformal embedding is a generalization of isometric embedding, where the metric can be changed by a scaling factor s. If s is globally constant for all $\boldsymbol{x}$, then the intrinsic geometry of the codebook manifold $(\boldsymbol{v}(\boldsymbol{x}), \forall\boldsymbol{x})$ remains Euclidean, i.e., flat.

Why isotropic scaling and conformal embedding? The neurons are intrinsically noisy. During path integration, the errors may accumulate in $\boldsymbol{v}$. Moreover, when inferring self-position from visual image, it is possible that $\boldsymbol{v}$ is inferred first with error, and then $\boldsymbol{x}$ is decoded from the inferred $\boldsymbol{v}$. Due to isotropic scaling and conformal embedding, locally we have $\|\delta\boldsymbol{v}\| = s\|\delta\boldsymbol{x}\|$, which guarantees that the ℓ_2 error in $\boldsymbol{v}$ translates proportionally to the ℓ_2 error in $\boldsymbol{x}$, so that there will not be adversarial perturbations in $\boldsymbol{v}(\boldsymbol{x})$ that cause excessively big errors in $\boldsymbol{x}$. Specifically, we have the following theorem.

Theorem 2. *Assume the general transformation model (Eq. (4)) and the isotropic scaling condition. For any fixed $\boldsymbol{x}$, let $s = \|f_\theta(\boldsymbol{v}(\boldsymbol{x}))\|$, which is independent of θ. Suppose the neurons are noisy: $\boldsymbol{v} = \boldsymbol{v}(\boldsymbol{x}) + \varepsilon$, where $\varepsilon \sim \mathscr{N}(0, \tau^2 \boldsymbol{I}_d)$ and d is the dimensionality of $\boldsymbol{v}$. Suppose the agent infers its 2D position $\hat{\boldsymbol{x}}$ from $\boldsymbol{v}$ by $\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}'} \|\boldsymbol{v} - \boldsymbol{v}(\boldsymbol{x}')\|^2$, i.e., $\boldsymbol{v}(\hat{\boldsymbol{x}})$ is the projection of $\boldsymbol{v}$ onto the 2D manifold formed by $(\boldsymbol{v}(\boldsymbol{x}'), \forall\boldsymbol{x}')$. Then we have*

$$\mathbb{E}\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2 = 2\tau^2/s^2. \tag{7}$$

See Supplementary for a proof.

Connection to continuous attractor neural network (CANN) defined on 2D torus. The group representation condition and the isotropic scaling condition appear to be satisfied by the CANN models [2, 6, 7, 29, 1] that are typically hand-designed on a 2D torus. See Supplementary for details.

3 Linear transformation

After studying the general transformation, we now investigate the linear transformation of $\boldsymbol{v}(\boldsymbol{x})$, for the following reasons. (1) It is the simplest transformation for which we can derive explicit algebraic and geometric results. (2) It enables us to connect the isotropic scaling condition to a special class of hexagon grid patterns. (3) In Section 4, we integrate it with the basis expansion model, which is also linear in $\boldsymbol{v}(\boldsymbol{x})$, via unitary group representation theory.

For finite (non-infinitesimal) self-motion, the linear transformation model is:

$$\boldsymbol{v}(\boldsymbol{x}+\Delta\boldsymbol{x}) = F(\boldsymbol{v}(\boldsymbol{x}), \Delta\boldsymbol{x}) = \boldsymbol{M}(\Delta\boldsymbol{x})\boldsymbol{v}(\boldsymbol{x}), \tag{8}$$

where $\boldsymbol{M}(\Delta\boldsymbol{x})$ is a matrix. The group representation condition becomes $\boldsymbol{M}(\Delta\boldsymbol{x}_1 + \Delta\boldsymbol{x}_2)\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{M}(\Delta\boldsymbol{x}_2)\boldsymbol{M}(\Delta\boldsymbol{x}_1)\boldsymbol{v}(\boldsymbol{x})$, i.e., $\boldsymbol{M}(\Delta\boldsymbol{x})$ is a matrix representation of self-motion $\Delta\boldsymbol{x}$, and $\boldsymbol{M}(\Delta\boldsymbol{x})$ acts on the coding manifold $(\boldsymbol{v}(\boldsymbol{x}), \forall\boldsymbol{x})$). For egocentric parametrization of self-motion $(\Delta r, \theta)$, we can further write $\boldsymbol{M}(\Delta\boldsymbol{x}) = \boldsymbol{M}_\theta(\Delta r)$ for $\Delta\boldsymbol{x} = (\Delta r \cos\theta, \Delta r \sin\theta)$, and the linear model becomes $\boldsymbol{v}(\boldsymbol{x}+\Delta\boldsymbol{x}) = F(\boldsymbol{v}(\boldsymbol{x}), \Delta r, \theta) = \boldsymbol{M}_\theta(\Delta r)\boldsymbol{v}(\boldsymbol{x})$.

3.1 Algebraic structure: matrix Lie algebra and Lie group

For the linear model (Eq. (8)), the directional derivative is: $f_\theta(\boldsymbol{v}(\boldsymbol{x})) = F'(\boldsymbol{v}(\boldsymbol{x}), 0, \theta) = \boldsymbol{M}'_\theta(0)\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{B}(\theta)\boldsymbol{v}(\boldsymbol{x})$, where $\boldsymbol{B}(\theta) = \boldsymbol{M}'_\theta(0)$, which is the derivative of $\boldsymbol{M}_\theta(\Delta r)$ with respect to Δr at 0. For infinitesimal self-motion, the transformation model in Eq. (5) becomes

$$\boldsymbol{v}(\boldsymbol{x}+\delta\boldsymbol{x}) = (\boldsymbol{I} + \boldsymbol{B}(\theta)\delta r)\boldsymbol{v}(\boldsymbol{x}) + o(\delta r), \tag{9}$$

where $\boldsymbol{I}$ is the identity matrix. It can be considered a linear recurrent network where $\boldsymbol{B}(\theta)$ is the learnable weight matrix. We have the following theorem for the algebraic structure of the linear transformation.

Theorem 3. *Assume the linear transformation model so that for infinitesimal self-motion* $(\delta r, \theta)$*, the model is in the form of Eq. (9), then for finite displacement* Δr*,*

$$\boldsymbol{v}(\boldsymbol{x}+\Delta\boldsymbol{x}) = \boldsymbol{M}_\theta(\Delta r)\boldsymbol{v}(\boldsymbol{x}) = \exp(\boldsymbol{B}(\theta)\Delta r)\boldsymbol{v}(\boldsymbol{x}). \tag{10}$$

Proof. We can divide Δr into N steps, so that $\delta r = \Delta r/N \to 0$ as $N \to \infty$, and

$$\boldsymbol{v}(\boldsymbol{x}+\Delta\boldsymbol{x}) = (\boldsymbol{I}+\boldsymbol{B}(\theta)(\Delta r/N)+o(1/N))^N\boldsymbol{v}(\boldsymbol{x}) \to \exp(\boldsymbol{B}(\theta)\Delta r)\boldsymbol{v}(\boldsymbol{x}) \tag{11}$$

as $N \to \infty$. The matrix exponential map is defined by $\exp(A) = \sum_{n=0}^{\infty} A^n/n!$. □

The above math underlies the relationship between matrix Lie algebra and matrix Lie group in general [38]. For a fixed θ, the set of $\boldsymbol{M}_\theta(\Delta r) = \exp(\boldsymbol{B}(\theta)\Delta r)$ for $\Delta r \in \mathbb{R}$ forms a *matrix Lie group*, which is both a group and a manifold. The tangent space of $\boldsymbol{M}_\theta(\Delta r)$ at identity $\boldsymbol{I}$ is called *matrix Lie algebra*. $\boldsymbol{B}(\theta)$ is the basis of this tangent space, and is often referred to as the *generator matrix*.

Path integration. If the agent starts from $\boldsymbol{x}_0$, and make a sequence of moves $((\Delta r_t, \theta_t), t = 1, ..., T)$, then the vector representation of self-position is updated by

$$\boldsymbol{v}_t = \exp(\boldsymbol{B}(\theta_t)\Delta r_t)\boldsymbol{v}_{t-1}, \tag{12}$$

where $\boldsymbol{v}_0 = \boldsymbol{v}(\boldsymbol{x}_0)$, and $t = 1, ..., T$.

Approximation to exponential map. For a finite but small Δr, $\exp(\boldsymbol{B}(\theta)\Delta r)$ can be approximated by a second-order (or higher-order) Taylor expansion

$$\exp(\boldsymbol{B}(\theta)\Delta r) = \boldsymbol{I}+\boldsymbol{B}(\theta)\Delta r+\boldsymbol{B}(\theta)^2\Delta r^2/2+o(\Delta r^2). \tag{13}$$

3.2 Geometric structure: rotation, periodicity, metic and error correction

If we assume $\boldsymbol{B}(\theta) = -\boldsymbol{B}(\theta)^\top$, i.e., skew-symmetric, then $\boldsymbol{I}+\boldsymbol{B}(\theta)\delta r$ in Eq. (9) is a rotation matrix operating on $\boldsymbol{v}(\boldsymbol{x})$, due to the fact that $(\boldsymbol{I}+\boldsymbol{B}(\theta)\delta r)(\boldsymbol{I}+\boldsymbol{B}(\theta)\delta r)^\top = \boldsymbol{I}+O(\delta r^2)$. For finite Δr, $\exp(\boldsymbol{B}(\theta)\Delta r)$ is also a rotation matrix, as it equals to the product of N matrices $\boldsymbol{I}+\boldsymbol{B}(\theta)(\Delta r/N)$ (Eq. (11)). The geometric interpretation is that, if the agent moves along the direction θ in the physical space, the vector $\boldsymbol{v}(\boldsymbol{x})$ is rotated by the matrix $\boldsymbol{B}(\theta)$ in the neural space, while the ℓ_2 norm $\|\boldsymbol{v}(\boldsymbol{x})\|^2$ remains fixed. We may interpret $\|\boldsymbol{v}(\boldsymbol{x})\|^2 = \sum_{i=1}^d v_i(\boldsymbol{x})^2$ as the total energy of grid cells. See Fig. 1(b).

The angle of rotation is given by $\|\boldsymbol{B}(\theta)\boldsymbol{v}(\boldsymbol{x})\|\delta r/\|\boldsymbol{v}(\boldsymbol{x})\|$, because $\|\boldsymbol{B}(\theta)\boldsymbol{v}(\boldsymbol{x})\|\delta r$ is the arc length and $\|\boldsymbol{v}(\boldsymbol{x})\|$ is the radius. If we further assume the isotropic scaling condition, which becomes that $\|f_\theta(\boldsymbol{v}(\boldsymbol{x}))\| = \|\boldsymbol{B}(\theta)\boldsymbol{v}(\boldsymbol{x})\|$ is constant over θ for the linear model, then the angle of rotation can be written as $\mu\delta r$, where $\mu = \|\boldsymbol{B}(\theta)\boldsymbol{v}(\boldsymbol{x})\|/\|\boldsymbol{v}(\boldsymbol{x})\|$ is independent of θ. Geometrically, μ tells us how fast the vector rotates in the neural space as the agent moves in the physical space. In practice, μ can be much bigger than 1 for the learned model, thus the vector can rotate back to itself in a short distance, causing the periodic patterns in the elements of $\boldsymbol{v}(\boldsymbol{x})$. μ captures the notion of metric.

For $\mu \gg 1$, the conformal embedding in Fig. 1 (b) **magnifies** the local motion in Fig. 1 (a), and this enables error correction [34]. More specifically, we have the following result, which is based on Theorem 2.

Proposition 1. *Assume the linear transformation model (Eq. (9)) and the isotropic scaling condition 2. For any fixed* $\boldsymbol{x}$*, let* $\mu = \|\boldsymbol{B}(\theta)\boldsymbol{v}(\boldsymbol{x})\|/\|\boldsymbol{v}(\boldsymbol{x})\|$*. Suppose* $\boldsymbol{v} = \boldsymbol{v}(\boldsymbol{x})+\varepsilon$*, where* $\varepsilon \sim \mathcal{N}(0, \tau^2\boldsymbol{I}_d)$ *and* $\tau^2 = \alpha^2(\|\boldsymbol{v}(\boldsymbol{x})\|^2/d)$*, so that* α^2 *measures the variance of noise relative to the average magnitude of* $(v_i(\boldsymbol{x})^2, i = 1, ..., d)$*. Suppose the agent infers its 2D position* $\hat{\boldsymbol{x}}$ *from* $\boldsymbol{v}$ *by* $\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}'}\|\boldsymbol{v}-\boldsymbol{v}(\boldsymbol{x}')\|^2$*. Then we have*

$$\mathbb{E}\|\hat{\boldsymbol{x}}-\boldsymbol{x}\|^2 = 2\alpha^2/(\mu^2 d). \tag{14}$$

See Supplementary for a proof. By the above proposition, error correction of grid cells is due to two factors: (1) higher dimensionality d of $\boldsymbol{v}(\boldsymbol{x})$ for encoding 2D positions $\boldsymbol{x}$, and (2) a magnifying $\mu \gg 1$ (our analysis is local for any fixed $\boldsymbol{x}$, and μ may depend on $\boldsymbol{x}$).

3.3 Hexagon grid patterns formed by mixing Fourier waves

In this subsection, we make connection between the isotropic scaling condition 2 and a special class of hexagon grid patterns created by linearly mixing three Fourier plane waves whose directions are $2\pi/3$ apart. We show such linear mixing satisfies the linear transformation model and the isotropic scaling condition.

Theorem 4. *Let $\boldsymbol{e}(\boldsymbol{x}) = (\exp(i\langle \boldsymbol{a}_j, \boldsymbol{x}\rangle), j = 1,2,3)^\top$, where $(\boldsymbol{a}_j, j = 1,2,3)$ are three 2D vectors of equal norm, and the angle between every pair of them is $2\pi/3$. Let $\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{U}\boldsymbol{e}(\boldsymbol{x})$, where $\boldsymbol{U}$ is an arbitrary unitary matrix. Let $\boldsymbol{B}(\theta) = \boldsymbol{U}^*\boldsymbol{D}(\theta)\boldsymbol{U}$, where $\boldsymbol{D}(\theta) = \text{diag}(i\langle \boldsymbol{a}_j, \boldsymbol{q}(\theta)\rangle, j = 1,2,3)$, with $\boldsymbol{q}(\theta) = (\cos\theta, \sin\theta)^\top$. Then $(\boldsymbol{v}(\boldsymbol{x}), \boldsymbol{B}(\theta))$ satisfies the linear transformation model (Eq. (9)) and the isotropic scaling condition 2. Moreover, $\boldsymbol{B}(\theta)$ is skew-symmetric.*

See Supplementary for a proof. We would like to emphasize that the above theorem analyzes a special case solution to our linear transformation model, but our optimization-based learning method **does not assume any superposition of Fourier basis functions** as in the theorem. Our experimental results are learned purely by optimizing a loss function based on the simple assumptions of our model with generic vectors and matrices.

We leave it to future work to theoretically prove that the isotropic scaling condition leads to hexagon grid patterns in either the general transformation model or the linear transformation model. The hexagon grid patterns are not limited to superpositions of three plane waves as in the above theorem.

3.4 Modules

Biologically, it is well established that grid cells are organized in discrete modules [4, 37] or blocks. We thus partition the vector $\boldsymbol{v}(\boldsymbol{x})$ into K blocks, $\boldsymbol{v}(\boldsymbol{x}) = (\boldsymbol{v}_k(\boldsymbol{x}), k = 1,...,K)$. Correspondingly the generator matrices $\boldsymbol{B}(\theta) = \text{diag}(\boldsymbol{B}_k(\theta), k = 1,...,K)$ are block diagonal, so that each sub-vector $\boldsymbol{v}_k(\boldsymbol{x})$ is rotated by a sub-matrix $\boldsymbol{B}_k(\theta)$. For the general transformation model, each sub-vector is transformed by a separate sub-network. By the same argument as in Section 3.2, let $\mu_k = \|\boldsymbol{B}_k\boldsymbol{v}_k(\boldsymbol{x})\|/\|\boldsymbol{v}_k(\boldsymbol{x})\|$, then μ_k is the metric of module k.

4 Interaction with place cells

4.1 Place cells

For each $\boldsymbol{v}(\boldsymbol{x})$, we need to uniquely decode $\boldsymbol{x}$ globally. This can be accomplished via interaction with place cells. Specifically, each place cell fires when the agent is at a specific position. Let $A(\boldsymbol{x}, \boldsymbol{x}')$ be the response map of the place cell associated with position $\boldsymbol{x}'$. It measures the adjacency between $\boldsymbol{x}$ and $\boldsymbol{x}'$. A commonly used form of $A(\boldsymbol{x}, \boldsymbol{x}')$ is the Gaussian adjacency kernel $A(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\|\boldsymbol{x} - \boldsymbol{x}'\|^2/(2\sigma^2))$. The set of Gaussian adjacency kernels serve as inputs to our optimization-based method to learn grid cells.

4.2 Basis expansion

A popular model that connects place cells and grid cells is the following basis expansion model (or PCA-based model) [13]:

$$A(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{v}(\boldsymbol{x}), \boldsymbol{u}(\boldsymbol{x}')\rangle = \sum_{i=1}^{d} u_{i,\boldsymbol{x}'} v_i(\boldsymbol{x}), \tag{15}$$

where $\boldsymbol{v}(\boldsymbol{x}) = (v_i(\boldsymbol{x}), i = 1,...,d)^\top$, and $\boldsymbol{u}(\boldsymbol{x}') = (u_{i,\boldsymbol{x}'}, i = 1,...,d)^\top$. Here $(v_i(\boldsymbol{x}), i = 1,...,d)$ forms a set of d basis functions (which are functions of $\boldsymbol{x}$) for expanding $A(\boldsymbol{x}, \boldsymbol{x}')$ (which is a function of $\boldsymbol{x}$ for each place $\boldsymbol{x}'$), while $\boldsymbol{u}(\boldsymbol{x}')$ is the read-out weight vector for place cell at $\boldsymbol{x}'$ and needs to be learned. See Fig. 2 for an illustration. Experimental results on biological brains have shown that the connections from grid cells to place cells are excitatory [42, 31]. We thus assume that $u_{i,\boldsymbol{x}'} \geq 0$ for all i and $\boldsymbol{x}'$.

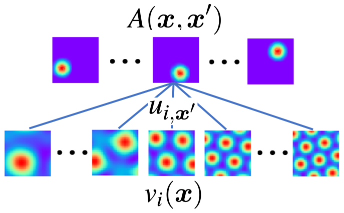


Figure 2: Illustration of basis expansion model $A(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^{d} u_{i,\boldsymbol{x}'} v_i(\boldsymbol{x})$, where $v_i(\boldsymbol{x})$ is the response map of i-th grid cell, shown at the bottom, which shows 5 different i. $A(\boldsymbol{x}, \boldsymbol{x}')$ is the response map of place cell associated with $\boldsymbol{x}'$, shown at the top, which shows 3 different $\boldsymbol{x}'$. $u_{i,\boldsymbol{x}'}$ is the connection weight.

4.3 From group representation to basis functions

The vector representation $\boldsymbol{v}(\boldsymbol{x})$ generated (or constrained) by the linear transformation model (Eq. (8)) can serve as basis functions of the PCA-based basis expansion model (Eq. (15)), due to the fundamental theorems of Schur [41] and Peter-Weyl [38], which reveal the deep root of Fourier analysis and generalize it to general Lie groups. Specifically, if $\boldsymbol{M}(\Delta\boldsymbol{x})$ is an irreducible unitary representation of $\Delta\boldsymbol{x}$ that forms a compact Lie group, then the elements $\{M_{ij}(\Delta\boldsymbol{x})\}$ form a set of orthogonal basis functions of $\Delta\boldsymbol{x}$. Let $\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{M}(\boldsymbol{x})\boldsymbol{v}(0)$ (where we choose the origin 0 as the reference point). The elements of $\boldsymbol{v}(\boldsymbol{x})$, i.e., $(v_i(\boldsymbol{x}), i = 1,...,d)$, are linear mixings of the basis functions $\{M_{ij}(\boldsymbol{x})\}$, so that they themselves form a new set of basis functions that serve to expand $(A(\boldsymbol{x},\boldsymbol{x}'), \forall\boldsymbol{x}')$ that parametrizes the place cells. Thus group representation in our path integration model is a perfect match to the basis expansion model, in the sense that the basis functions are results of group representation.

The basis expansion model (or PCA-based model) (Eq. (15)) assumes that the basis functions are orthogonal, whereas in our work, **we do not make the orthogonality assumption**. Interestingly, the learned transformation model generates basis functions that are close to being orthogonal automatically. See Supplementary for more detailed explanation and experimental results.

4.4 Decoding and re-encoding

For a neural response vector $\boldsymbol{v}$, such as $\boldsymbol{v}_t$ in Eq. (12), the response of the place cell associated with location $\boldsymbol{x}'$ is $\langle \boldsymbol{v}, \boldsymbol{u}(\boldsymbol{x}')\rangle$. We can decode the position $\hat{\boldsymbol{x}}$ by examining which place cell has the maximal response, i.e.,

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}'} \langle \boldsymbol{v}, \boldsymbol{u}(\boldsymbol{x}')\rangle. \tag{16}$$

After decoding $\hat{\boldsymbol{x}}$, we can re-encode $\boldsymbol{v} \leftarrow \boldsymbol{v}(\hat{\boldsymbol{x}})$ for error correction. Decoding and re-encoding can also be done by directly projecting $\boldsymbol{v}$ onto the manifold $(\boldsymbol{v}(\boldsymbol{x}), \forall\boldsymbol{x})$, which gives similar results. See Supplementary for more analysis and experimental results.

5 Learning

We learn the model by optimizing a loss function defined based on three model assumptions discussed above: (1) the basis expansion model (Eq. (15)), (2) the linear transformation model (Eq. (10)) and (3) the isotropic scaling condition 2. The input is the set of adjacency kernels $A(\boldsymbol{x},\boldsymbol{x}'), \forall\boldsymbol{x},\boldsymbol{x}'$. The unknown parameters to be learned are (1) $(\boldsymbol{v}(\boldsymbol{x}) = (\boldsymbol{v}_k(\boldsymbol{x}), k = 1,...,K), \forall\boldsymbol{x})$, (2) $(\boldsymbol{u}(\boldsymbol{x}'), \forall\boldsymbol{x}')$ and (3) $(\boldsymbol{B}(\theta), \forall\theta)$. We assume that there are K modules or blocks and $\boldsymbol{B}(\theta)$ is skew-symmetric, so that $\boldsymbol{B}(\theta)$ are parametrized as block-diagonal matrices $(\boldsymbol{B}_k(\theta), k = 1,...,K), \forall\theta)$ and only the lower triangle parts of the matrices need to be learned. The loss function is defined as a weighted sum of simple ℓ_2 loss terms constraining the three model assumptions: $L = L_0 + \lambda_1 L_1 + \lambda_2 L_2$, where

$$L_0 = \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'}[A(\boldsymbol{x},\boldsymbol{x}') - \langle \boldsymbol{v}(\boldsymbol{x}), \boldsymbol{u}(\boldsymbol{x}')\rangle]^2, \text{ (basis expansion)} \tag{17}$$

$$L_1 = \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{x},\Delta\boldsymbol{x}} \|\boldsymbol{v}_k(\boldsymbol{x}+\Delta\boldsymbol{x}) - \exp(\boldsymbol{B}_k(\theta)\Delta r)\boldsymbol{v}_k(\boldsymbol{x})\|^2, \text{ (transformation)} \tag{18}$$

$$L_2 = \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{x},\theta,\Delta\theta}[\|\boldsymbol{B}_k(\theta+\Delta\theta)\boldsymbol{v}_k(\boldsymbol{x})\| - \|\boldsymbol{B}_k(\theta)\boldsymbol{v}_k(\boldsymbol{x})\|]^2. \text{ (isotropic scaling)} \tag{19}$$

In L_1, $\Delta\boldsymbol{x} = (\Delta r\cos\theta, \Delta r\sin\theta)$. λ_1 and λ_2 are chosen so that the three loss terms are of similar magnitudes. $A(\boldsymbol{x},\boldsymbol{x}')$ are given as Gaussian adjacency kernels. For regularization, we add a penalty on $\|\boldsymbol{u}(\boldsymbol{x}')\|^2$, and further assume $\boldsymbol{u}(\boldsymbol{x}') \geq 0$ so that the connections from grid cells to place cells are excitatory [42, 31]. However, note that $\boldsymbol{u}(\boldsymbol{x}') \geq 0$ is not necessary for the emergence of hexagon grid patterns as shown in the ablation studies.

Expectations in L_0, L_1 and L_2 are approximated by Monte Carlo samples. L is minimized by *Adam* [25] optimizer. See Supplementary for implementation details.

It is worth noting that, consistent with the experimental observations, we assume individual place field $A(\boldsymbol{x},\boldsymbol{x}')$ to exhibit a Gaussian shape, rather than a Mexican-hat pattern (with balanced excitatory center and inhibitory surround) as assumed in previous basis expansion models [13, 33] of grid cells.

ReLU non-linearity. We also experiment with a non-linear transformation model where a ReLU activation is added. See Supplementary for details.

6 Experiments

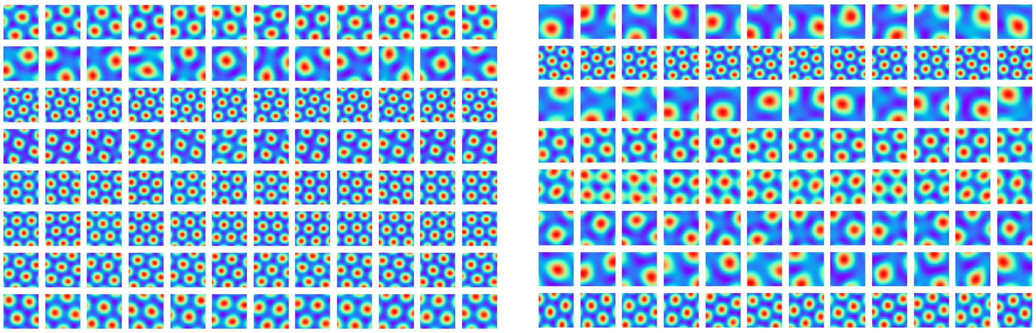


Figure 3: Hexagonal grid firing patterns emerge in the learned network. Every response map shows the firing pattern of one neuron (i.e, one element of $\boldsymbol{v}$) in the 2D environment. Every row shows the firing patterns of the neurons within the same block or module.

We conduct numerical experiments to learn the representations as described in Section 5. Specifically, we use a square environment with size 1m × 1m, which is discretized into a 40×40 lattice. For direction, we discretize the circle $[0, 2\pi]$ into 144 directions and use nearest neighbor linear interpolations for values in between. We use the second-order Taylor expansion (Eq. (13)) to approximate the exponential map $\exp(\boldsymbol{B}(\theta)\Delta r)$. The displacement Δr are sampled within a small range, i.e., Δr is smaller than 3 grids on the lattice. For $A(\boldsymbol{x}, \boldsymbol{x}')$, we use a Gaussian adjacency kernel with $\sigma = 0.07$. $\boldsymbol{v}(\boldsymbol{x})$ is of $d = 192$ dimensions, which is partitioned into $K = 16$ modules, each of which has 12 cells.

6.1 Hexagon grid patterns

Fig. 3 shows the learned firing patterns of $\boldsymbol{v}(\boldsymbol{x}) = (v_i(\boldsymbol{x}), i = 1, ..., d)$ over the 40×40 lattice of $\boldsymbol{x}$. Every row shows the learned units belonging to the same block or module. Regular hexagon grid patterns emerge. Within each block or module, the scales and orientations are roughly the same, but with different phases or spatial shifts. For the learned $\boldsymbol{B}(\theta)$, each element shows regular sine/cosine tuning over θ. See Supplementary for more learned patterns.

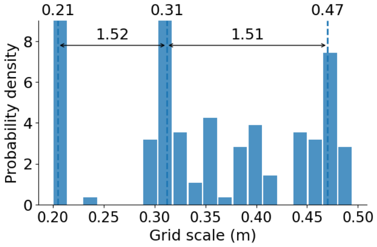


Figure 4: Multi-modal distribution of grid scales of the learned model grid cells. The scale ratios closely match the real data [37].

Table 1: Summary of gridness scores of the patterns learned from different models. To determine valid grid cells, we apply the same threshold of gridness score as in [3], i.e., gridness score > 0.37. For our model, we run 5 trials and report the average and standard deviation.

Model	Gridness score (↑)	% of grid cells
[3] (LSTM)	0.18	25.20
[33] (RNN)	0.48	56.10
Ours	**0.90** ± 0.044	**73.10** ± 1.33

We further investigate the characteristics of the learned firing patterns of $\boldsymbol{v}(\boldsymbol{x})$ using measures adopted from the literature of grid cells. Specifically, the hexagonal regularity, scale and orientation of grid-like patterns are quantified using the gridness score, grid scale and grid orientation [26, 32], which are determined by taking a circular sample of the autocorrelogram of the response map. Table 1 summarizes the results of gridness scores and comparisons with other optimization-based approaches [3, 33]. We apply the same threshold to determine whether a learned neuron can be considered a grid cell as in [3] (i.e., gridness score > 0.37). For our model, 73.10% of the learned neurons exhibit significant hexagonal periodicity in terms of the gridness score. Fig. 4 shows the histogram of grid scales of the learned grid cell neurons (mean 0.33, range 0.21 to 0.49), which follows a multi-modal distribution. The ratio between neighboring modes are roughly 1.52 and 1.51, which closely matches the theoretical predictions [39, 36] and also the empirical results from rodent grid cells [37]. Collectively, these results reveal striking, quantitative correspondence between the properties of our model neurons and those of the grid cells in the brain.

Connection to continuous attractor neural network (CANN) defined on 2D torus. The fact that the learned response maps of each module are shifted versions of a common hexagon periodic pattern implies that the learned codebook manifold forms a 2D torus, and as the agent moves, the responses of the grid cells undergo a cyclic permutation. This is consistent with the CANN models hand-crafted on 2D torus. See Supplementary for a detailed discussion.

Ablation studies. We conduct ablation studies to examine whether certain model assumptions are empirically important for the emergence of hexagon grid patterns. The conclusions are highlighted as follows: (1) The loss term L_2 (Eq. (19)) constraining the isotropic scaling condition is necessary for learning hexagon grid patterns. (2) The constraint $\boldsymbol{u}(\boldsymbol{x}') \geq 0$ is not necessary for learning hexagon patterns, but the activations can be either excitatory or inhibitory without the constraint. (3) The skew-symmetric assumption on $\boldsymbol{B}(\boldsymbol{\theta})$ is not important for learning hexagon grid pattern. (4) Hexagon patterns always emerge regardless of the choice of block size and number of blocks. (5) Multiple blocks or modules are necessary for the emergence of hexagon grid patterns of multiple scales. See Fig. 5 for several learned patterns and Supplementary for the full studies.

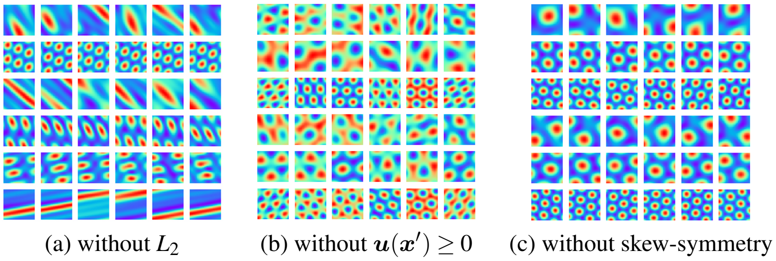

(a) without L_2 (b) without $\boldsymbol{u}(\boldsymbol{x}') \geq 0$ (c) without skew-symmetry

Figure 5: Learned response maps in ablation studies where a certain model assumption is removed. (a) Remove the loss term L_2. (b) Remove the assumption $\boldsymbol{u}(\boldsymbol{x}') \geq 0$. (c) Remove the skew-symmetric assumption on $\boldsymbol{B}(\boldsymbol{\theta})$.

6.2 Path integration

We then examine the ability of the learned model on performing multi-step path integration, which can be accomplished by recurrently updating $\boldsymbol{v}_t$ (Eq. (12)) and decoding $\boldsymbol{v}_t$ to $\boldsymbol{x}_t$ for $t = 1, ..., T$ (Eq. (16)). Re-encoding $\boldsymbol{v}_t \leftarrow \boldsymbol{v}(\boldsymbol{x}_t)$ after decoding is adopted. Fig. 6(a) shows an example trajectory of accurate path integration for number of time steps $T = 30$. As shown in Fig. 6(b), with re-encoding, the path integration error remains close to zero over a duration of 500 time steps (< 0.01 cm, averaged over 1,000 episodes), even if the model is trained with the single-time-step transformation model (Eq. (18)). Without re-encoding, the error goes slight higher but still remains small (ranging from 0.0 to 4.2 cm, mean 1.9 cm in the 1m × 1m environment). Fig. 6(c) summarizes the path integration performance by fixing the number of blocks and altering the block size. The performance of path integration would be improved as the block size becomes larger, i.e., with more neurons in each module. When block size is larger than 16, path integration is very accurate for the time steps tested.

Error correction. See Supplementary for numerical experiments on error correction, which show that the learn model is still capable of path integration when we apply Gaussian white noise errors or Bernoulli drop-out errors to $\boldsymbol{v}_t$.

6.3 Additional experiments on path planning and egocentric vision

We also conduct additional experiments on path planning and egocentric vision with our model. Path planning can be accomplished by steepest ascent on the adjacency to the target position. For egocentric vision, we learn an extra generator network that generates the visual image given the position encoding formed by the grid cells. See Supplementary for details.

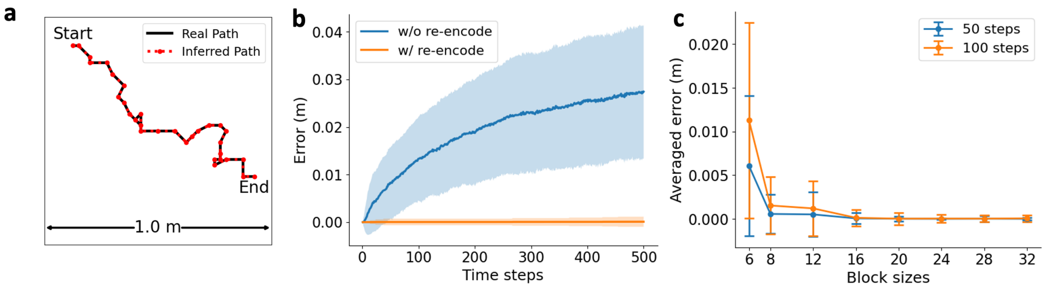


Figure 6: The learned model can perform accurate path integration. (a) Black: example trajectory. Red: inferred trajectory. (b) Path integration error over number of time steps, for procedures with re-encoding and without re-encoding. (c) Path integration error with fixed number of blocks and different block sizes, for 50 and 100 time steps. The error band in (b) and error bar in (c) are standard deviations computed over 1,000 episodes.

7 Related work

Our work is related to several lines of previous research on modeling grid cells. First, RNN models have been used to model grid cells and path integration. The traditional approach uses simulation-based models with hand-crafted connectivity, known as continuous attractor neural network (CANN) [2, 6, 7, 29, 1]. On the other hand, more recently two pioneering papers [9, 3] developed optimization-based RNN approaches to learn the path integration model and discovered that grid-like response patterns can emerge in the optimized networks. These results are further substantiated in [33, 8]. Our work analyzes the properties of the general recurrent model for path integration, and these properties seem to be satisfied by the hand-crafted CANN models. Our method belongs to the scheme of optimization-based approaches, and the learned response maps share similar properties as assumed by the CANN models.

Second, our work differs from the PCA-based basis expansion models [13, 33, 35] in that, unlike PCA, we make no assumption about the orthogonality between the basis functions, and the basis functions are generated by the transformation model. Furthermore, in previous basis expansion models [13, 33], place fields with Mexican-hat patterns (with balanced excitatory center and inhibitory surround) had to be assumed in order to obtain hexagonal grid firing patterns. However, experimentally measured place fields in biological brains were instead well characterized by Gaussian functions. Crucially, in our model, hexagonal grids emerge from learning with Gaussian place fields, and there is no need to assume any additional surround mechanisms or difference of Gaussians kernels.

In another related paper, [19] proposed matrix representation of 2D self-motion, while our work analyzes general transformations. Our investigation of the special case of linear transformation model reveals the matrix Lie group and the matrix Lie algebra of rotation group. Our work also connects the linear transformation model to the basis expansion model via unitary group representation theory.

8 Conclusion

This paper analyzes the recurrent model for path integration calculations by grid cells. We identify a group representation condition and an isotropic scaling condition that give rise to locally conformal embedding of the self-motion. We study a linear prototype model that reveals the matrix Lie group of rotation, and explore the connection between the isotropic scaling condition and hexagon grid patterns. In addition to these theoretical investigations, our numerical experiments demonstrate that our model can learn hexagon grid patterns for the response maps of grid cells, and the learned model is capable of accurate long distance path integration.

In this work, the numerical experiments are mostly limited to the linear transformation model, with the exception of an experiment with ReLU non-linearity. We will conduct experiments on the other non-linear transformation models, especially the forms assumed by the hand-crafted continuous attractor neural networks. Moreover, we assume that the agent navigates within a square open-field environment without obstacles or rewards. It is worthwhile to explore more complicated environments, including 3D environment.

Acknowledgments and Disclosure of Funding

The work was supported by NSF DMS-2015577, ONR MURI project N00014-16-1-2007, DARPA XAI project N66001-17-2-4029, and XSEDE grant ASC170063. We thank Yaxuan Zhu from UCLA Department of Statistics for his help with experiments on egocentric vision. We thank Dr. Wenhao Zhang for sharing his knowledge and insights on continuous attractor neural networks. We thank Sirui Xie for discussions. We thank the three reviewers for their constructive comments.

References

[1] Haggai Agmon and Yoram Burak. A theory of joint attractor dynamics in the hippocampus and the entorhinal cortex accounts for artificial remapping and grid cell field-to-field variability. *eLife*, 9:e56894, 2020.

[2] Daniel J Amit. *Modeling brain function: The world of attractor neural networks*. Cambridge university press, 1992.

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

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

[5] Hugh T Blair, Adam C Welday, and Kechen Zhang. Scale-invariant memory representations emerge from moire interference between grid fields that produce theta oscillations: a computational model. *Journal of Neuroscience*, 27(12):3211–3229, 2007.

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

[7] Jonathan J Couey, Aree Witoelar, Sheng-Jia Zhang, Kang Zheng, Jing Ye, Benjamin Dunn, Rafal Czajkowski, May-Britt Moser, Edvard I Moser, Yasser Roudi, et al. Recurrent inhibitory circuitry as a mechanism for grid formation. *Nature neuroscience*, 16(3):318–324, 2013.

[8] Christopher J Cueva, Peter Y Wang, Matthew Chin, and Xue-Xin Wei. Emergence of functional and structural properties of the head direction system by optimization of recurrent neural networks. *International Conferences on Learning Representations (ICLR)*, 2020.

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

[10] Charles Darwin. Origin of certain instincts, 1873.

[11] Licurgo de Almeida, Marco Idiart, and John E Lisman. The input–output transformation of the hippocampal granule cells: from grid cells to place fields. *Journal of Neuroscience*, 29(23):7504–7512, 2009.

[12] Christian F Doeller, Caswell Barry, and Neil Burgess. Evidence for grid cells in a human memory network. *Nature*, 463(7281):657, 2010.

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

[14] Ariane S Etienne and Kathryn J Jeffery. Path integration in mammals. *Hippocampus*, 14(2):180–192, 2004.

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

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

[17] Marianne Fyhn, Torkel Hafting, Menno P Witter, Edvard I Moser, and May-Britt Moser. Grid cells in mice. *Hippocampus*, 18(12):1230–1238, 2008.

[18] Marianne Fyhn, Sturla Molden, Menno P Witter, Edvard I Moser, and May-Britt Moser. Spatial representation in the entorhinal cortex. *Science*, 305(5688):1258–1264, 2004.

[19] Ruiqi Gao, Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. Learning grid cells as vector representation of self-position coupled with matrix representation of self-motion. *arXiv preprint arXiv:1810.05597*, 2018.

[20] Mariana Gil, Mihai Ancau, Magdalene I Schlesiger, Angela Neitz, Kevin Allen, Rodrigo J De Marco, and Hannah Monyer. Impaired path integration in mice with disrupted grid cell firing. *Nature neuroscience*, 21(1):81–91, 2018.

[21] 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, 2005.

[22] Aidan J Horner, James A Bisby, Ewa Zotow, Daniel Bush, and Neil Burgess. Grid-like processing of imagined navigation. *Current Biology*, 26(6):842–847, 2016.

[23] Joshua Jacobs, Christoph T Weidemann, Jonathan F Miller, Alec Solway, John F Burke, Xue-Xin Wei, Nanthia Suthana, Michael R Sperling, Ashwini D Sharan, Itzhak Fried, et al. Direct recordings of grid-like neuronal activity in human spatial navigation. *Nature neuroscience*, 16(9):1188, 2013.

[24] Nathaniel J Killian, Michael J Jutras, and Elizabeth A Buffalo. A map of visual space in the primate entorhinal cortex. *Nature*, 491(7426):761, 2012.

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[26] 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, 2010.

[27] Bruce L McNaughton, Francesco P Battaglia, Ole Jensen, Edvard I Moser, and May-Britt Moser. Path integration and the neural basis of the'cognitive map'. *Nature Reviews Neuroscience*, 7(8):663, 2006.

[28] John O'Keefe. A review of the hippocampal place cells. *Progress in neurobiology*, 13(4):419–439, 1979.

[29] Hugh Pastoll, Lukas Solanka, Mark CW van Rossum, and Matthew F Nolan. Feedback inhibition enables theta-nested gamma oscillations and grid firing fields. *Neuron*, 77(1):141–154, 2013.

[30] Thomas Ridler, Jonathan Witton, Keith G Phillips, Andrew D Randall, and Jonathan T Brown. Impaired speed encoding is associated with reduced grid cell periodicity in a mouse model of tauopathy. *bioRxiv*, page 595652, 2019.

[31] David C Rowland, Horst A Obenhaus, Emilie R Skytøen, Qiangwei Zhang, Cliff G Kentros, Edvard I Moser, and May-Britt Moser. Functional properties of stellate cells in medial entorhinal cortex layer ii. *Elife*, 7:e36664, 2018.

[32] 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.

[33] Ben Sorscher, Gabriel Mel, Surya Ganguli, and Samuel A Ocko. A unified theory for the origin of grid cells through the lens of pattern formation. 2019.

[34] Sameet Sreenivasan and Ila Fiete. Grid cells generate an analog error-correcting code for singularly precise neural computation. *Nature neuroscience*, 14(10):1330, 2011.

[35] Kimberly L Stachenfeld, Matthew M Botvinick, and Samuel J Gershman. The hippocampus as a predictive map. *Nature neuroscience*, 20(11):1643, 2017.

[36] Martin Stemmler, Alexander Mathis, and Andreas VM Herz. Connecting multiple spatial scales to decode the population activity of grid cells. *Science Advances*, 1(11):e1500816, 2015.

[37] 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, 2012.

[38] Michael Taylor. Lectures on lie groups. *Lecture Notes, available at http://www. unc. edu/math/Faculty/met/lieg. html*, 2002.

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

[40] Michael M Yartsev, Menno P Witter, and Nachum Ulanovsky. Grid cells without theta oscillations in the entorhinal cortex of bats. *Nature*, 479(7371):103, 2011.

[41] Anthony Zee. *Group theory in a nutshell for physicists*. Princeton University Press, 2016.

[42] Sheng-Jia Zhang, Jing Ye, Chenglin Miao, Albert Tsao, Ignas Cerniauskas, Debora Ledergerber, May-Britt Moser, and Edvard I Moser. Optogenetic dissection of entorhinal-hippocampal functional connectivity. *Science*, 340(6128), 2013.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See the abstract and introduction sections.
   (b) Did you describe the limitations of your work? [Yes] See the last paragraph of Section 8.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See supplementary.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Theorem 1, 2, 3 and 4 and Proposition 1.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Supplementary for proofs of Theorem 1, 2, 4 and Proposition 1. See Section 3.1 for proof of Theorem 3.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Supplementary for the code.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See the beginning of Section 6 and Supplementary "implementation details".
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Table 1 and Figure 6.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Supplementary "implementation details".
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Table 1 for citations and Supplementary "implementation details".
   (b) Did you mention the license of the assets? [Yes] See Supplementary "implementation details".
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See Supplementary for the code.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See Supplementary "implementation details".
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See Supplementary "implementation details".
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We don't conduct research with human subjects
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] We don't conduct research with human subjects
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] We don't conduct research with human subjects