

Path Learning and Path Decoding with Grid Cells and Place Cells

Lijuan Su, Jean-Marc Fellous and Onur Ozan Koyluoglu

April 22, 2016

Abstract

When a rat navigates within an environment, the traversed path which is important to many behaviors, is encoded by firing rates and neural connections of place cells and grid cells. Place cells are located in the Hippocampus which have active firing fields at specific locations in 2 dimensional space. Grid cells on the other hand, are located in the adjacent Medial Entorhinal Cortex(MEC) which is activated whenever the animal's position coincides with any vertex of a regular grid of equilateral triangles spanning the surface of the environment. In our paper, we assume that an animal learning a path through specific locations will form three weight matrices (Grid-Grid, Grid-Place, and Place-Place) that could act as a signature for the learned path. Using different cell models and different neural connectivity learning rules, our models can learn different neural connections between these networks. From these learned weight matrices, it is possible to decode the path. We evaluate the path learning and path decoding performance with different grid cell models and different learning rules.

1 Introduction

Mammalian spatial navigation is central to most behaviors and requires an understanding of the environment at multiple spatial scales (refs in humans and rats). How these scales are established during development and how they are used when an animal forages or learns a specific spatial layout are essentially unknown. The neural substrate of spatial navigation in the rodent has been extensively studied, and several types of neurons have been found that encode specific features potentially useful for spatial navigation.

Place cells are located in the Hippocampus and have active firing fields at specific locations in 2 dimensional space, called 'place fields' (O'Keefe and Nadel, 1978). Grid cells, on the other hand, are located in the adjacent Medial Entorhinal Cortex(MEC) and they show periodic firing as a function of location. Specifically, a grid cell is activated whenever the animal traverses through any vertex of a regular grid of equilateral triangles that span the environment (Hafting et al., 2005). Both grid and place cells are organized across dorsal-ventral axis in increasing spatial field sizes. The size gradient seems to be smooth for place cells (ref), but has been shown to be modular for grid cells (ref). Place and grid cells are functionally reciprocally connected in an ordered fashion, within each level (ref).

Grid cells can be defined by three parameters: spacing, orientation and phase. The spacing of a grid cell can be defined as the distance between the peak points of two neighboring grid fields of a given grid cell. The orientation of a grid cell is the tilt of the grid relative to a reference axis, and the phase is the displacement in the x and y directions relative to an external reference point. In other words, grid cells that belong to the same module share a common spacing and orientation, but their phases differ (Hafting et al., 2005).

Together, grid and place cells form a topographical map for navigational tasks (Moser et al., 2008). This map can be characterized by a weight matrix, where the entries correspond to synaptic connections between grid and place cells (Burak and Fiete, 2009). In other words, if we have a network of M grid and N place cells, we can use $M \times M$ Grid-Grid weight matrix, $M \times N$ Grid-Place weight matrix, $N \times N$ Place-Place weight matrix to keep track of all possible synaptic connections. Using Hebbian plasticity rules, these connections can be strengthened or weakened in accordance with the fields visited during behavior.

A rodent learning a path through specific locations will form three weight matrices (Grid-Grid, Grid-Place, and Place-Place) that could act as a signature for the learned path. In this paper we try to decode the learned path from these learned weight matrices, and compare the decoding results with different grid cell models and different learning rules. With our model, we try to compare the path decoding results from the following configurations: Path decoding with only grid cells, only place cells, or grid-place cells together; Path decoding with linear or modular grid cell model; Path decoding with unsupervised Hebbian or decay Hebbian rules; Path decoding with the same or different learning rules for cells from one module and cells from two different modules.

2 Models and Methods

2.1 Place Cell Model

In our model, we consider N place cells with a continuously increasing field size across the dorsal-ventral axis. Place fields are assumed to have Gaussian tuning curves suggested by (O'Keefe and Burgess, 1996), with the probability density function being

$$f(x, y) = \frac{\exp \left\{ -\frac{1}{2(1-\rho^2)} \left[\left(\frac{x-\mu_x}{\sigma_x} \right)^2 - 2\rho \left(\frac{x-\mu_x}{\sigma_x} \right) \left(\frac{y-\mu_y}{\sigma_y} \right) + \left(\frac{y-\mu_y}{\sigma_y} \right)^2 \right] \right\}}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \quad (1)$$

where (μ_x, μ_y) is the mean vector and the covariance matrix is

$$\begin{pmatrix} Var(X) & Cov(X, Y) \\ Cov(X, Y) & Var(Y) \end{pmatrix} = \begin{pmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{pmatrix} \quad (2)$$

In the model, σ_x^2 and σ_y^2 varies from 50 to 675. σ_x^2 and σ_y^2 defines the field scale and $\rho\sigma_x\sigma_y$ defines the shape of the field. ρ is assumed to be 0, which results in circular place fields. Since the result of a Gaussian function never achieves 0, we have clipped the function and assigned the value 0 where the result of the function is less than 10^{-1} . In addition, since larger variance would result in smaller peak firing rates, to eliminate that effect, we normalize each cell with its peak firing value.

Table 1: Place Cell Model Configuration

	range
μ	Random (0-150)
σ^2	Linear (50-625)
N	100

2.2 Grid Cell Model

Grid cells, on the other hand, are assumed to consist of three 2-D cosine functions as suggested by (Solstad et al., 2006), with their gratings oriented at different angles, $\pi/3$ apart. There are M grid cells with field sizes increasing continuously or modularly across the dorsal-ventral axis. The gaps between grid fields increase as the grid field sizes increase, in accordance with (Brun et al., 2008) and (Hafting et al., 2005). Using (Lyttle et al., 2013), we have derived the grid cell model as follows:

$$G(\mathbf{s}, \lambda, \theta, \mathbf{c}) = \mathbf{g} \left(\sum_{k=1}^3 \cos \left(\frac{4\pi}{\sqrt{3}\lambda} \mathbf{u}(\theta_k - \theta) \cdot (\mathbf{s} - \mathbf{c}) \right) \right) \quad (3)$$

$$\lambda = p1 * sp^2 + p2 * sp + p3$$

where $\mathbf{s} = (\mathbf{x}, \mathbf{y})$ is the location vector (1×2 vector in 2D space), k is the inter-vertex spacing between grid points (in cm), $\mathbf{c} = (\mathbf{x}_0, \mathbf{y}_0)$ is the spatial phase (in cm relative to the origin), $\mathbf{u}(\theta_k) = (\cos(\theta_k), \sin(\theta_k))$ is a unit vector denoting grid orientation in the direction θ_k . (\cdot denotes the inner product). In the grid cell models, we use $\theta_1 = 0$, $\theta_2 = \pi/3$ and $\theta_3 = 2\pi/3$, and the sum of 3 cosine functions are applied to $g(x)$, where $g(x) = \exp(a(x - b)) - 1$, and $a = 0.3$ and $b = -1.5$, in accordance with (de Almeida et al., 2009). We normalize each grid cell with its peak value so that the peak firing rate in each grid cell is 1.

sp is the spacing between two adjacent firing fields of one grid cell, where values of the parameters in Equation (3) are: $p1 = 1.00$, $p2 = 0.34$, and $p3 = -9.06$. In the way, the values of sp are varied between 20 cm and 120 cm, similar to the ones used in (Moser et al., 2014).

In our experiment, we test two model configurations: one is with continuously increasing field sizes across the dorsal-ventral axis; the other is with several discrete field sizes across the dorsal-ventral axis. Grid Cell Model configurations are shown in the Table 2.

2.3 Path Model

Regarding the path that the rat follows, we consider both experimental data and simulation data, each includes random path and reward path. The maze used for collecting experimental data is circular shaped with a radius of 150 cm. For the reward experimental path, within the maze there are three reward locations that the rat is fed when it visits those locations. Hence,

Table 2: Grid Cell Model Configuration

	Linear	Modular
Phase (\mathbf{v})	Random	Random
Spacing (sp)	Linear (20cm-120cm)	Discrete (38cm,48cm,65cm,98cm)
Orientation (θ)	Random between (0-10)	Discrete (0,3,6,9)
M	100	100

once the rat learns the experiment, it does not explore the whole simulation area and tends to traverse the shortest path between reward locations.

For the simulations, we have defined a circular simulation area with radius being 150 cm, and moved our virtual rat within that, in order to be consistent in term of size. Random path algorithm explaining the movement of a foraging rat is derived from (Hasselmo et al., 2007) and can be summarized by the following equations:

$$\begin{aligned}\Delta x(t) &= S(1 - m)p_x + m\Delta x(t - 1) \\ \Delta y(t) &= S(1 - m)p_y + m\Delta y(t - 1)\end{aligned}\quad (4)$$

where $S = 5$, $m = 0.8$, and $p_x, p_y \sim \mathbf{N}(\mathbf{0}, \mathbf{1})$. This way, rat's motion heavily depends on its momentum, i.e., it cannot change its direction drastically. To ensure that rat stays within the boundaries of the simulation area, we use the following formulas

$$\begin{aligned}\Delta x_r(t) &= -\Delta x(t) \\ \Delta y_r(t) &= -\Delta y(t)\end{aligned}\quad (5)$$

whenever rat initiates a motion towards out of the boundaries, and we basically reflect that motion.

The path used in our model are set as Table 3, in which the experimental data included one random path and one 3-rewards path; and the simulation data also included one random path and one 3-rewards path.

Table 3: Path Configurations

	Experiment Path	Simulation Path
Random	ExpRan	SimRan
3 Rewards	ExpRew3	SimRew3

Given one environment, there are two representations of one path: one is the sequences of the traversed locations along time; the other is the frequencies of all the locations in the environment. In our paper, we use five paths to test our model as shown in Figure 1.

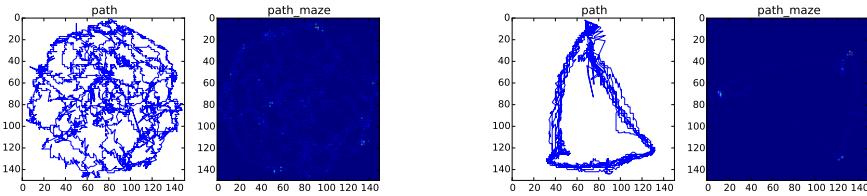
2.4 Average firing rates

In Figure 3(a) the locations at which the grid cell spikes are superimposed on the experimental trajectory are shown in red. Each red dot corresponds to one spike. Figure 3(b) and 3(c) shows the grid cell spikes on the simulated trajectory and place cell spikes on experimental trajectory, respectively.

3 Path Learning Over Navigational Path

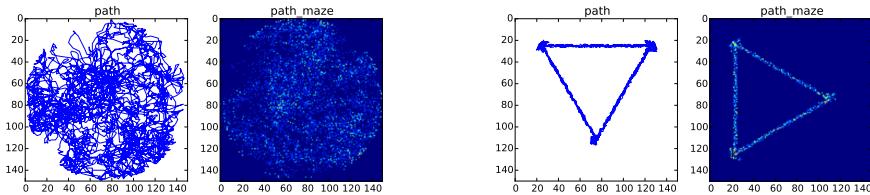
3.1 Overlap of firing fields between grid cells in 1D

Here, we are trying to find out the relationship between the periods of two grid cells in order to maximize the cross-correlation between their firing fields. When space is restricted to one



(a) Experimental Random Path

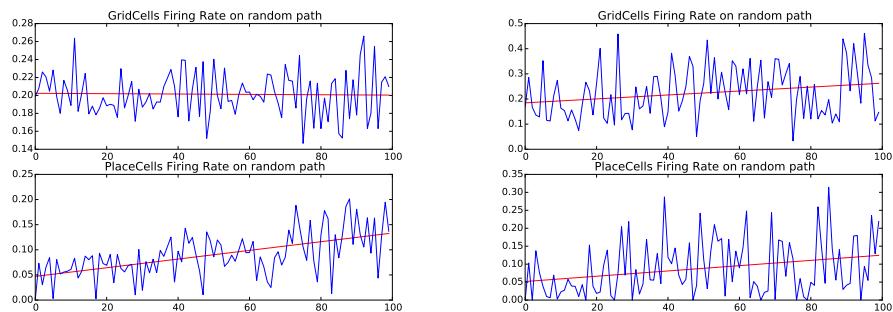
(b) Experimental Reward Path



(c) Simulation Random Path

(d) Simulation Reward Path

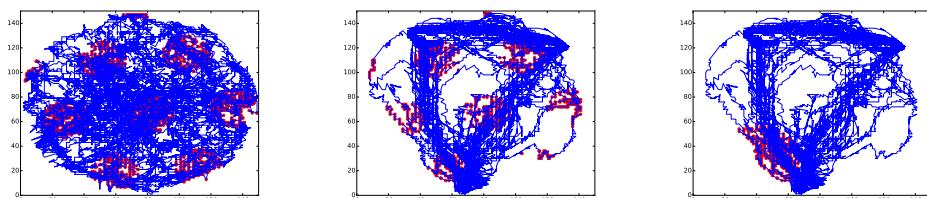
Figure 1: Sequency and Frequency Representation of Path



(a) Experimental Random Path

(b) Experimental Reward Path

Figure 2: Average firing rates



(a) Grid Cell

(b) Grid Cell

(c) Place Cell

Figure 3: Average firing rates

dimension as on a narrow track, the firing rate maps can be represented by von Mises functions, which are periodic generalizations of the Gaussian.

$$G(x, c_i, \lambda_i) = n_i \exp\left\{k_i \cos \frac{2\pi(x - c_i)}{\lambda_i} - 1\right\} \quad (6)$$

where x is the location in 1D space, c_i is the preferred spatial phase of cell i , λ_i is its spatial period, n_i and k_i are the normalized factor to ensure the peak firing rate of grid cells same.

The cross-correlation formula of two grid cells in one dimension can be represented as:

$$\begin{aligned} (G_i * G_j) &= \int_0^R G(x, c_i, \lambda_i) * G(x, c_j, \lambda_j) dx \\ &= \int_0^R n_i \exp\left\{k_i \cos \frac{2\pi(x - c_i)}{\lambda_i} - 1\right\} * n_j \exp\left\{k_j \cos \frac{2\pi(x - c_j)}{\lambda_j} - 1\right\} dx \\ &= \int_0^R n_i n_j * \exp\left\{k_i \cos \frac{2\pi(x - c_i)}{\lambda_i} - 1 + k_j \cos \frac{2\pi(x - c_j)}{\lambda_j} - 1\right\} dx \end{aligned} \quad (7)$$

where R is the length of the narrow track in one dimension area, c_i and c_j are the phases grid cell i and grid cell j , λ_i and λ_j are the periods respectively. There is no simple formula exists for this integral, so here we simpify the fring rate maps of the grid cells in one dimension area as follows:

$$G(x, c_i, \lambda_i) = 1 + \cos\left(\frac{2\pi}{\lambda_i}x + c_i\right) \quad (8)$$

If we assume the phases are equal to 0 and analytically solve the Equation, the cross-correlation formula of the two grid cells in one dimension can be simplified as:

$$\begin{aligned} (G_i * G_j) &= \int_0^R (1 + \cos(\frac{2\pi}{\lambda_i}x + c_i))(1 + \cos(\frac{2\pi}{\lambda_j}x + c_j)) dx \\ &= \int_0^R (1 + \cos(\frac{2\pi x}{\lambda_i}) + \cos(\frac{2\pi x}{\lambda_j}) + \cos(\frac{2\pi x}{\lambda_i}) \cos(\frac{2\pi x}{\lambda_j})) dx \\ &= R + \frac{\lambda_i}{2\pi} \sin \frac{2\pi R}{\lambda_i} + \frac{\lambda_j}{2\pi} \sin \frac{2\pi R}{\lambda_j} + \frac{\lambda_i \lambda_j}{4\pi(\lambda_i - \lambda_j)} \sin \frac{2\pi R(\lambda_i - \lambda_j)}{\lambda_i \lambda_j} \\ &\quad + \frac{\lambda_i \lambda_j}{4\pi(\lambda_i + \lambda_j)} \sin \frac{2\pi R(\lambda_i + \lambda_j)}{\lambda_i \lambda_j} \end{aligned} \quad (9)$$

The fourth term in (9) is equal to

$$\frac{R}{2} \text{sinc}\left(\frac{2\pi R(\lambda_i - \lambda_j)}{\lambda_i \lambda_j}\right)$$

which is maximized as $(\lambda_i - \lambda_j) \rightarrow 0$. The fifth term, on the other hand, is equal to

$$\frac{R}{2} \text{sinc}\left(\frac{2\pi R(\lambda_i + \lambda_j)}{\lambda_i \lambda_j}\right)$$

which is maximized as λ_i and λ_j gets larger. Accordingly, second and third terms can also be rewritten as $R \text{sinc}(2\pi x/\lambda_i)$ and $R \text{sinc}(2\pi x/\lambda_j)$, which are also maximized as λ_i and λ_j gets larger. In short, we can say that the integral is maximized when $\lambda_i = \lambda_j$ and they are both large. So if we rewrite (9) with sinc terms, we get

$$\begin{aligned} (G_i * G_j) &= \int_0^R (1 + \cos(\frac{2\pi}{\lambda_i}x + c_i))(1 + \cos(\frac{2\pi}{\lambda_j}x + c_j)) dx \\ &= R(1 + \text{sinc}(\frac{2\pi x}{\lambda_i}) + \text{sinc}(\frac{2\pi x}{\lambda_j}) + \frac{1}{2} \text{sinc}(\frac{2\pi R(\lambda_i - \lambda_j)}{\lambda_i \lambda_j})) \\ &\quad + \frac{1}{2} \text{sinc}(\frac{2\pi R(\lambda_i + \lambda_j)}{\lambda_i \lambda_j})) \end{aligned} \quad (10)$$

In order to compute the integral of equations, we assumed that $R = 200\text{cm}$ and we have used 3 reference λ_i of grid cell i which are 40 cm, 60 cm and 80 cm, and the λ_j of grid cell j is ranged from 20cm to 200cm. The resulting value of the normalized integral of $G_i * G_j$ is shown in Figures 4(a), 4(b) and 4(c) for all 3 reference λ values.

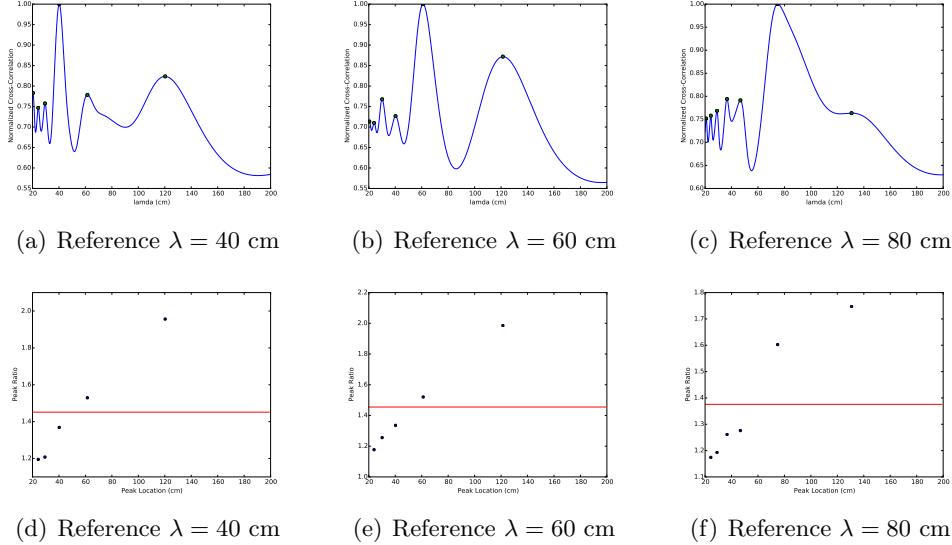


Figure 4: Top row: Normalized Cross-correlation in 1D versus λ for 3 Reference Periods. Bottom row: Ratio of the two consecutive peaks versus peak locations.

We also investigate the ratio of the two consecutive peaks in Figures 4(d), 4(e) and 4(f). The average peak ratio shown by the red line is 1.4513 in Figure 4(d), 1.4549 in Figure 4(e) and 1.3756 in Figure 4(f). The findings are compatible with Moser's paper.

3.2 Overlap of firing fields between grid cells in 2D

Here we did the same analysis for 2D analyisi as we did 1D. As shown in the results, the average peak ratio is 1.3801 in Figure 5(d), 1.4742 in Figure 5(e) and 1.3529 in Figure 5(f).

$$\begin{aligned}
 (G_i * G_j) &= \int_0^R \int_0^R G(\mathbf{s}, \lambda_i, \theta_i, \mathbf{c}_i) * G(\mathbf{s}, \lambda_j, \theta_j, \mathbf{c}_j) d\mathbf{x} d\mathbf{y} \\
 &= \int_0^R \int_0^R g \left(\sum_{k=1}^3 \cos \left(\frac{4\pi}{\sqrt{3}\lambda_i} \mathbf{u}(\theta_k - \theta_i) \cdot (\mathbf{s} - \mathbf{c}_i) \right) \right) \\
 &\quad * g \left(\sum_{k=1}^3 \cos \left(\frac{4\pi}{\sqrt{3}\lambda_j} \mathbf{u}(\theta_k - \theta_j) \cdot (\mathbf{s} - \mathbf{c}_j) \right) \right) d\mathbf{x} d\mathbf{y}
 \end{aligned} \tag{11}$$

Where R is the maze size, $\mathbf{s} = (\mathbf{x}, \mathbf{y})$ is the location vector (1×2 vector in 2D space), k is the inter-vertex spacing between grid points (in cm), $\mathbf{c} = (\mathbf{x}_0, \mathbf{y}_0)$ is the spatial phase (in cm relative to the origin), $\mathbf{u}(\theta_k) = (\cos(\theta_k), \sin(\theta_k))$ is a unit vector denoting grid orientation in the direction θ_k (\cdot denotes the inner product), and $g(x) = \exp(a(x - b)) - 1$ where $a = 0.3$ and $b = -1.5$.

In addition, we wanted to see the effect for all the cells with diffferent periods and orientations. The firing fields of different cells maybe overlapped, which is related to the weight connections beween these cells. Here we compute the overlap of cell i and cell j in 2 dimension environment. To compare the overlaps for different cell pairs, given the firing fields of G^i and G^j , and the maze size of m , we use three methods to represent the overlap of two cells over 2 dimension environment.

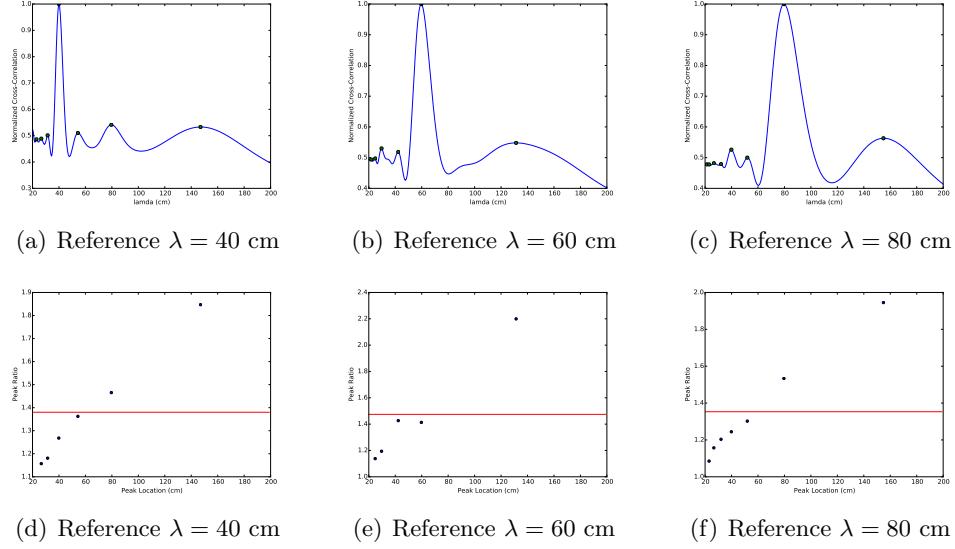


Figure 5: Top row: Normalized Cross-correlation in 2D versus λ for 3 Reference Periods. Bottom row: Ratio of the two consecutive peaks versus peak locations.

Method1 (Pearson Product-moment Correlation Coefficient, PCC): Pearson product-moment correlation coefficients of the firing fields of G^i and G^j .

$$\rho^{ij} = \frac{\text{cov}(G^i, G^j)}{\sigma_i \sigma_j} \quad (12)$$

Where cov is the covariance between G_i and G_j , the σ_i and σ_j are the standard deviations of the G^i and G^j .

Method2 (Mean Squared Error, MSE): Average of the sum of the squared errors between the firing fields of G^i and G^j .

$$MSE^{ij} = \begin{bmatrix} (G_{11}^i - G_{11}^j)^2 & (G_{12}^i - G_{12}^j)^2 & \dots & (G_{1m}^i - G_{1m}^j)^2 \\ (G_{21}^i - G_{21}^j)^2 & (G_{22}^i - G_{22}^j)^2 & \dots & (G_{2m}^i - G_{2m}^j)^2 \\ \vdots & \vdots & \vdots & \ddots \\ (G_{m1}^i - G_{m1}^j)^2 & (G_{m2}^i - G_{m2}^j)^2 & \dots & (G_{mm}^i - G_{mm}^j)^2 \end{bmatrix} \quad (13)$$

Method3 (Cross Correlation, CCR): Average of the sum of the products between the firing fields of G^i and G^j .

$$CCR^{ij} = \begin{bmatrix} G_{11}^i * G_{11}^j & G_{12}^i * G_{12}^j & \dots & G_{1m}^i * G_{1m}^j \\ G_{21}^i * G_{21}^j & G_{22}^i * G_{22}^j & \dots & G_{2m}^i * G_{2m}^j \\ \vdots & \vdots & \vdots & \ddots \\ G_{m1}^i * G_{m1}^j & G_{m2}^i * G_{m2}^j & \dots & G_{mm}^i * G_{mm}^j \end{bmatrix} \quad (14)$$

Where G_{xy}^i means the firing rate of cell i at position (x, y) ; and G_{xy}^j means the firing rate of cell j at position (x, y) .

3.3 Neural Connectivity Learning Over Navigational Path

To acquire some knowledge about the synaptic connectivity of the neurons, we try to obtain the weight matrix between grid and place cells. For that purpose, we started with unsupervised Hebbian rule where the matrix entries w_{ij} s are updated as follows

$$w_{ij}(t) = w_{ij}(t-1) + \alpha x_i(s(t))x_j(s(t)) \quad (15)$$

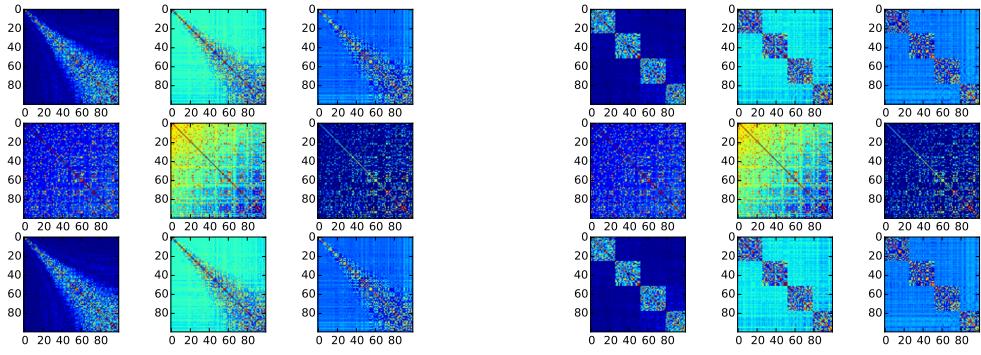


Figure 6: Overlap between Cells: Row1 is GG, Row2 is PP, Row3 is GP; Column1 is PCC 12, Column2 is MSE 13, Column3 is CCR 14.

Here, $s(t)$ is the stimuli at time t , α is the learning rate, x_i is the pre-synaptic neuron and x_j is the post-synaptic neuron. w_{ij} s corresponds to synapses, and as a result, when two cells 'fire together they wire together'. As expressed in the Equation 15, when both neurons fire with stimuli $s(t)$, we update both w_{ij} and w_{ji} , regardless of which neuron fires first. Thus, the resulting weight matrices are symmetric. We start with initializing the matrix with zeros and update the entries at each location on the path.

However, one problem with unsupervised Hebbian rule is that the entries can grow without bounds. To overcome this, we decided to switch to the Hebbian rule with decay and modified our update equation by adding a decay element as follows

$$w_{ij}(t) = \begin{cases} w_{ij}(t-1) + \alpha_{ij} x_i(s(t)) x_j(s(t)) & \text{if } x_i(s(t)) \cdot x_j(s(t)) > 0 \\ w_{ij}(t-1) - \gamma_{ij} x_i(s(t)) & \text{if } x_j(s(t)) = 0, x_i(s(t)) \geq 0 \end{cases} \quad (16)$$

where $\gamma_{ij} = \gamma$ is the decay rate. Here, α_{ij} depends on i and j in order to ensure modularity. Because, we found out that when α is homogeneous across all i, j pairs, we cannot achieve modularity.

So here, we test two learning rules of the Equation 15 and Equation 16. Each with three types: continuously increasing grid field size, discrete modular grid field size but the connectivity learning rule is same; and discrete modular grid field size but the connectivity learning rule is different for two cells from the same modular (α_1, γ_1) and from two different modules (α_2, γ_2), as shown in Table 4.

Table 4: Unsupervised Hebbian Learning (U-HL) and Decayed Hebbian Learning (D-HL).

	Linear	Module1	Module2
U-HL	$\alpha = 0.3$	$\alpha = 0.3$	$\alpha_1 = 0.3; \alpha_2 = 0.03$
D-HL	$\alpha = 0.3, \gamma = 0.6$	$\alpha = 0.3, \gamma = 0.6$	$\alpha_1 = 0.3, \gamma_1 = 0.6; \alpha_2 = 0.03, \gamma_2 = 0.06$

4 Path Decoding and Evaluation

4.1 Path Decoding from Learned Neural Connectivity

Given the learned weight matrix W from a navigational path, we define the weight-maze to represent the path decoding from the weight matrix.

$$W = \begin{bmatrix} w_{11} & w_{12} & w_{13} & \dots & w_{1n} \\ w_{21} & w_{22} & w_{23} & \dots & w_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & w_{n3} & \dots & w_{nn} \end{bmatrix} \quad (17)$$

$$o^{ij} = \begin{bmatrix} v_{11}^i * v_{11}^j & v_{12}^i * v_{12}^j & \dots & v_{1m}^i * v_{1m}^j \\ v_{21}^i * v_{21}^j & v_{22}^i * v_{22}^j & \dots & v_{2m}^i * v_{2m}^j \\ \vdots & \vdots & \vdots & \ddots \\ v_{m1}^i * v_{m1}^j & v_{m2}^i * v_{m2}^j & \dots & v_{mm}^i * v_{mm}^j \end{bmatrix} \quad (18)$$

$$O = \sum w_{ij} * o^{ij} \quad (19)$$

Where W is the weight matrix between all the cells, w_{ij} is the weight between cell i and cell j ; n is the number of cells; v_i is the firing field of cell i on the whole maze; v_j is the firing field of cell j on the whole maze; o_{ij} is the overlap of cell i and cell j on the whole maze; $(m * m)$ is the maze size. O (weight-maze) is the sum of weighted overlaps of all cells;

4.2 Path Decoding Evaluation on Different Models

Given one path, using different cell models and different neural connectivity learning rules, animals can learn different neural connections between grid cells and grid cells (GG), between grid cells and place cells (GP), between place cells and place cells (PP). Here in our paper, our models on path learning and path decoding were tested with four paths, which included the experimental random path as shown in Figure 1(a) and Figure 7, the experimental reward path as shown in Figure 1(b) and Figure 8, the simulation random path as shown in Figure 1(c) and Figure 9, the simulation reward path as shown in Figure 1(d) and Figure 10.

In our experiment, we tested two grid cell models, two hebbian learning rules: Unsupervised hebbian learning rule (U-HL) defined in Equation 15; Decay hebbian learning rule (D-HL) defined in Equation 16. And we also test two grid cells models: one is with linear continuously increasing field sizes across the dorsal-ventral axis; the other one is with several modular discrete field sizes across the dorsal-ventral axis. Grid Cell Model configurations are shown in the Table 2.

Neural connectivities between grid cells and place cells with these two different grid cell models are learned through two different hebbian learning. One is the simple Unsupervised Hebbian Learning (U-HL) computed by Equation 15 and the other one is the Decayed Hebbian Learning (D-HL) computed by Equation 15. For grid cells with modular discrete field sizes across the dorsal-ventral axis, we also test the path learning and path decoding between uniformal and modular neural connectivity learning shown in Table 4. In this table, the Module1 represents uniformal neural connectivity learning, which means that the neural connectivity learning are same between grid cells within the same module and grid cells from different modules. And the Module2 represents modular neural connectivity learning, which means that the neural connectivity learning are different between grid cells within the same module and grid cells from different modules. Here we want to test which neural connectivity learning rule is more effective for path decoding and path learning.

Grid and place cells can form a topographical map for navigational tasks, which is characterized by three weight matrices, where the entries correspond to synaptic connections between grid and place cells. Animals can learn and decode the path with grid cells to grid cells network (GG), grid cells to place cells network (GP) or place cells to place cells network (PP). In our model, we assume animals learnt three weight matrices from the navigational path, which are GG, GP and PP. For each weight matrix, we can decode the path from the learnt weight matrix (shown as Equation 17) using Equation 18 and Equation 19.

As shown in figure 7, we evaluated the performance of our models on the experimental random path learning and decoding. Figure 7(a) showed the path learning and decoding results

from Linear grid cell model and Unsupervised hebbian learning rule; Figure 7(d) is evaluated on Linear grid cell model with Decay hebbian learning rule; Figure 7(b) is tested on Module1 grid cell model with Unsupervised hebbian learning rule; Figure 7(e) is tested on Module1 grid cell model with Decay hebbian learning rule; Figure 7(c) is from Module2 grid cell model with Unsupervised hebbian learning rule; Figure 7(f) is from Module2 grid cell model with Decay hebbian learning rule. In each subfigure: figures in the first row are the learnt weight matrices of GG, GP and PP, figures in the second row are the decoded path from the corresponding learnt weight matrix.

To compare the decoded path and the actual path, we compute the similarity between them using three methods, which are defined in PCC method 12, MSE method 14 and CCR method 14. Given the sequences representation of one path, we transform the sequences representation of the path to the frequency representation of this path. And then compare the similarity between the decoded path and the frequency representation of the actual path.

As shown in Figure 7, Figure 7(g) is the path decoding evaluation with PCC method defined in 12, Figure 7(h) is the path decoding evaluation with MSE method defined in 13, and Figure 7(i) is the path decoding evaluation with CCR method defined in 14. For each subfigure: group 1 is tested on Lineared grid cell model with U-HL learning rule, which corresponded to Figure 7(a); group 2 is tested on Lineared Grid Cell model with D-HL learning rule, which corresponded to Figure 7(d); group 3 is tested on Module1 Grid Cell model with U-HL learning rule, which corresponded to Figure 7(b); group 4 is tested on Module2 Grid Cell model with D-HL learning rule, which corresponded to Figure 7(e); group 5 is tested on Module2 Grid Cell model with U-HL learning rule, which corresponded to Figure 7(c); group 6 is the Module2 Grid Cells with D-HL learning rule, which corresponded to Figure 7(f). For each group, the path decoding are evaluated with GG, GP and PP network.

For experimental random path shown in figure 7, with MSE evaluation method shown in Figure 7(h), the results shown: 1)Unsupervised hebbian learning rule is better than the Decay hebbian learning rule for values of group1 (or group3, group5) is less than group2 (or group4, group6); 2)Module1 grid cell model is the best, linear grid model is the second, and module2 grid cell model is the third; 3)With unsupervised hebbian learning: GP network is the best, PP network is the second, and GG network is the third; but with decay hebbian learning: PP network is the best, GP is the second, and GG is the third. However with ccr evaluation method shown in figure 7(i), the results are on the contrary for greater value in MSE means worst but better in CCP.

For experimental reward path shown in figure 8, with MSE evaluation method shown in Figure 8(h), the results shown: 1)Unsupervised hebbian learning rule is better than the Decay hebbian learning rule; 2)Module1 grid cell model is the best, linear grid model is the second, and module2 grid cell model is the third; 3)With unsupervised hebbian learning: GP network is the best, PP network is the second, and GG network is the third; but with decay hebbian learning: PP network is the best, GP is the second, and GG is the third. And with CCR evaluation method shown in figure 8(i), the results showed that: 1)Unsupervised hebbian learning rule is better than the Decay hebbian learning rule; 2)Module2 grid cell model is the best, linear grid model is the second, and module1 grid cell model is the third; 3)With unsupervised hebbian learning: PP network is the best, GG network is the second, and GP network is the third; but with decay hebbian learning: GG network is the best, PP is the second, and GP is the third.

For simulation random path shown in figure 9, with MSE evaluation method shown in Figure 7(h), the results shown: 1)Unsupervised hebbian learning rule is better than the Decay hebbian learning rule; 2)Linear grid cell model is the best, Module1 grid model is the second, and module2 grid cell model is the third; 3)With unsupervised or decay hebbian learning: GP network is the best, PP network is the second, and GG network is the third. However with CCR evaluation method shown in figure 9(i), the results showed that: 1)Unsupervised hebbian learning rule is better than the decay hebbian learning rule; 2)Linear grid cell model is the best, Module1 grid model is the second, and module1 grid cell model is the third; 3)With unsupervised hebbian learning: GP network is the best, PP network is the second, and GG network is the third; but with decay hebbian learning: GG network is the best, GP is the second, and PP is

the third.

For simulation reward path shown in figure 10, with MSE evaluation method shown in Figure 8(h), the results showed: 1)PP network is the best, GP network is the second, and GG network is the third; 2)Unsupervised hebbian learning rule is worse than the decay hebbian learning rule for GP and PP network, but uncertain for GG network; 3)For unsupervised hebbian learning rule: Linear grid cell model is the best, Module1 grid model is the second, and Module2 grid cell model is the third; For decay hebbian learning rule: Module1 is the best, Linear grid cell is the second, Module2 is the third. And with CCR evaluation method shown in figure 10(i), the results showed that: 1)Unsupervised hebbian learning rule is better than the decay hebbian learning rule; 2)For different grid cell models, module2 grid cell model is the best, linear grid model is the second, and module1 grid cell model is the third; 3)With unsupervised hebbian learning: GG network is the best, PP network is the second, and GP network is the third; but with decay hebbian learning: GG network is the best, GP is the second, and PP is the third.

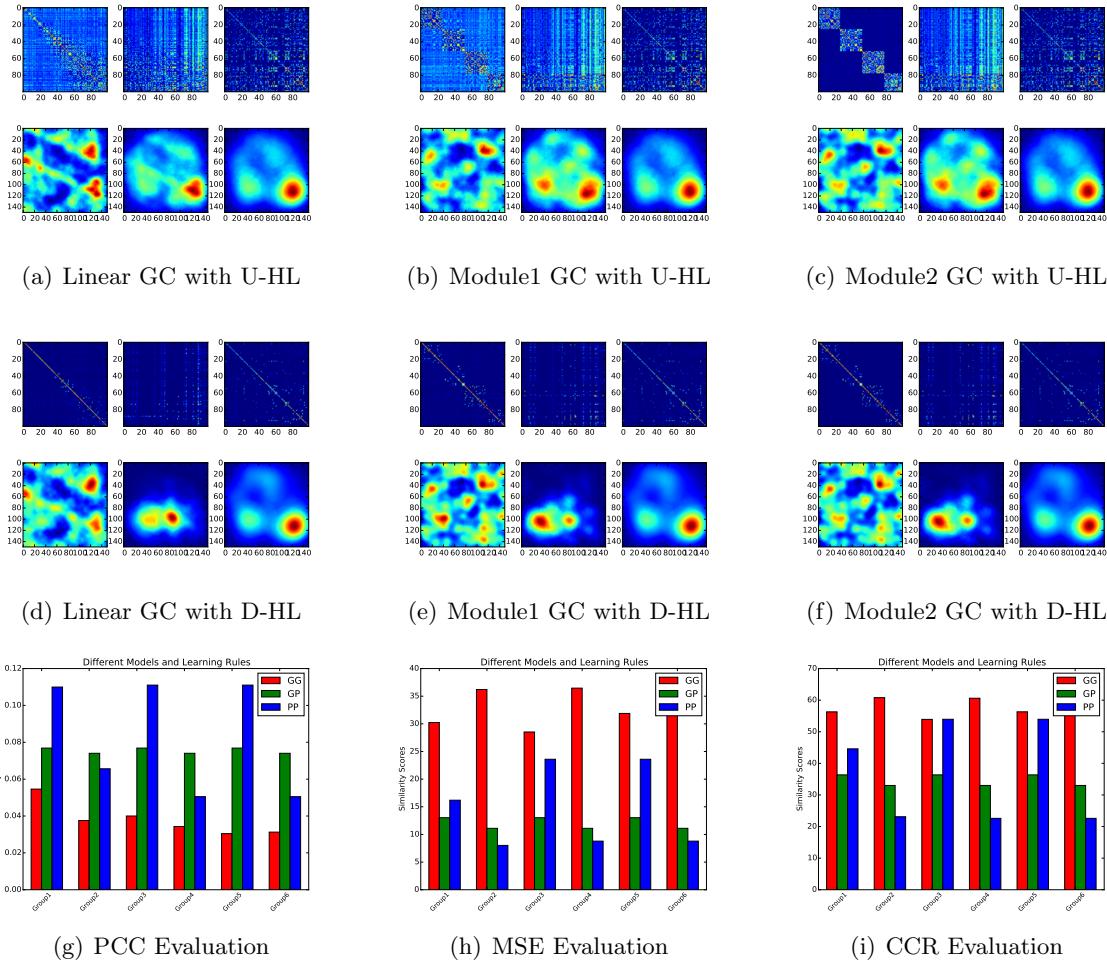


Figure 7: Path Decoding and Evaluation on ExpRan Path

5 Conclusion and Discussion

In this paper we try to explore how animals learn and decode path with neural connectivities, and evaluated the decoding results with different grid cell models and different learning rules.

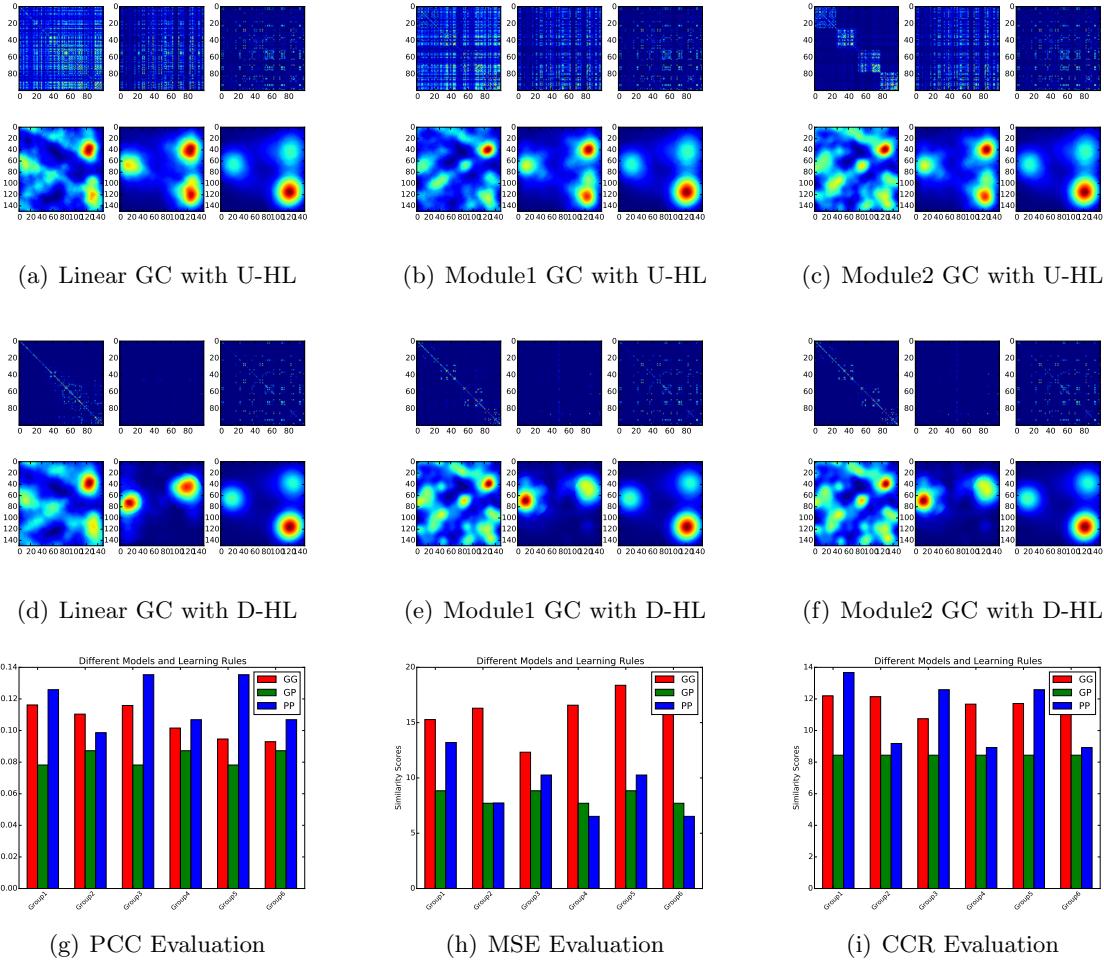


Figure 8: Path Decoding and Evaluation on ExpRew Path

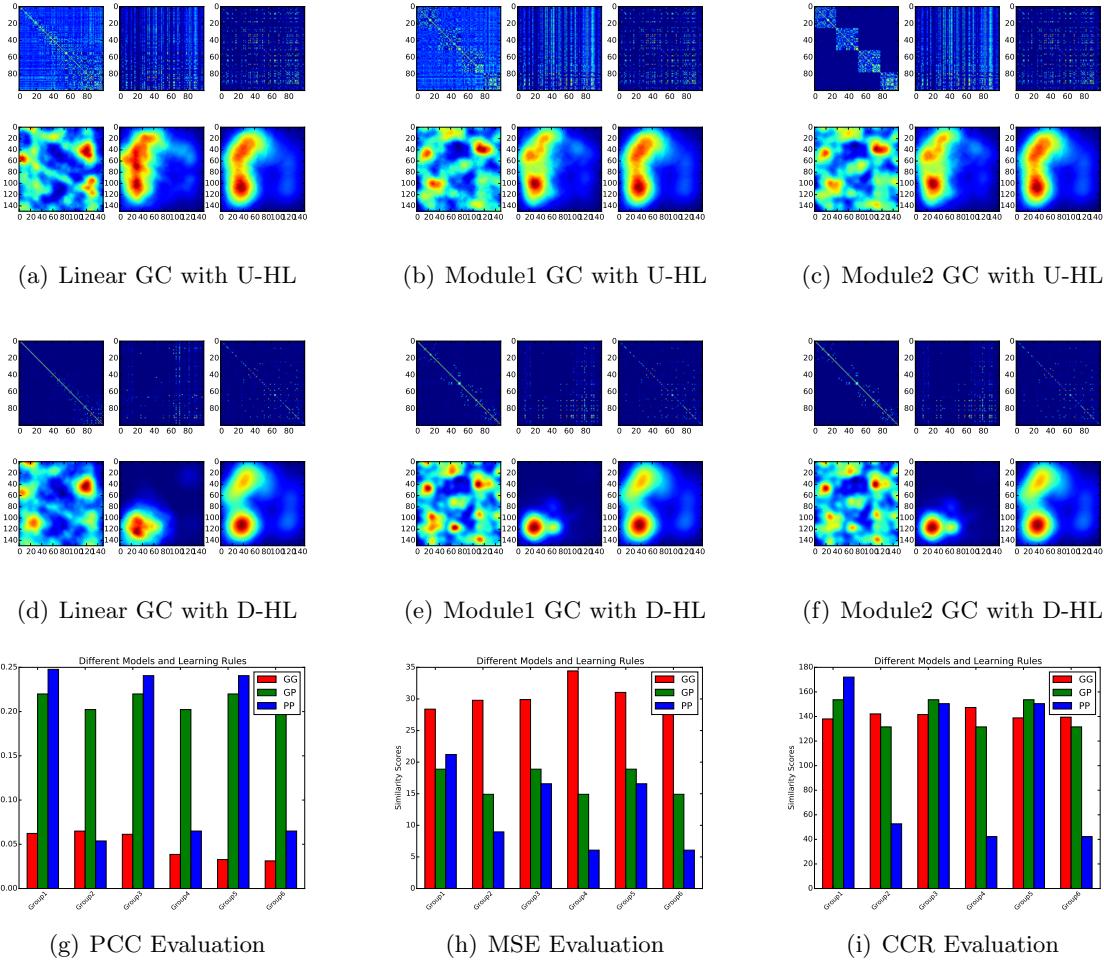


Figure 9: Path Decoding and Evaluation on SimRan Path

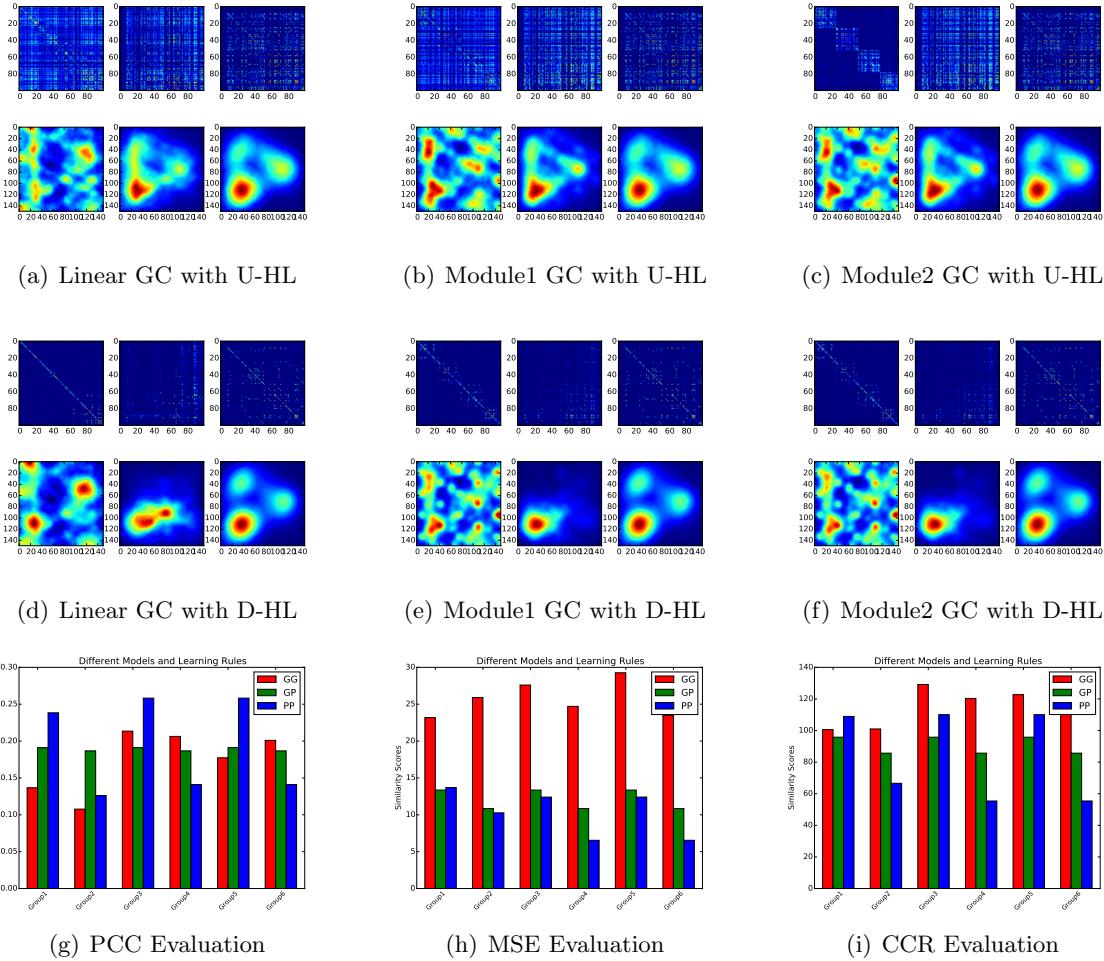


Figure 10: Path Decoding and Evaluation on Sim3Rew Path

From the comparison between the decoded path and the actual path, we try to answer the following questions related to effective path learning and path decoding: Which network are more effective for path learning and path decoding, grid cells network, place cells network, or grid-place cells network together; Which grid cell model is better, linear or modular grid cell model; Which learning rule is better, unsupervised or decay Hebbian learning rule; Which update strategy is better, the uniformal or modular learning rules for cells within the same module and cells from two different modules.

With all the evaluation on many different paths, we can safely draw a conclusion: 1)Unsupervised hebbian learning rule is better than the Decay hebbian learning rule; 2)Module grid cell model is the better than linear grid model on reward path learning and decoding; 3)With CCR Evaluation PP network is the best, however with CCR evaluation GG network is the best. This may mean that GG network is better to reinforce the traversed locations but may fail to avoid the untraversed locations error, and PP network has a tradeoff of the learning between the traversed locations and untraversed locations. And they together form a topographical map for path learning and path decoding in navigational tasks.

6 References

- Brun VH, Solstad T, Kjelstrup KB, Fyhn M, Witter MP, Moser EI, Moser MB (2008) Progressive increase in grid scale from dorsal to ventral medial entorhinal cortex. *Hippocampus* 18:1200–1212.
- Burak Y, Fiete IR (2009) Accurate path integration in continuous attractor network models of grid cells. *PLoS computational biology* 5:e1000291.
- de Almeida L, Idiart M, Lisman JE (2009) The input–output transformation of the hippocampal granule cells: from grid cells to place fields. *The Journal of Neuroscience* 29:7504–7512.
- Hafting T, Fyhn M, Molden S, Moser MB, Moser EI (2005) Microstructure of a spatial map in the entorhinal cortex. *Nature* 436:801–806.
- Hasselmo ME, Giocomo LM, Zilli EA (2007) Grid cell firing may arise from interference of theta frequency membrane potential oscillations in single neurons. *Hippocampus* 17:1252–1271.
- Lyttle D, Gereke B, Lin KK, Fellous JM (2013) Spatial scale and place field stability in a grid-to-place cell model of the dorsoventral axis of the hippocampus. *Hippocampus* 23:729–744.
- Moser EI, Kropff E, Moser MB (2008) Place cells, grid cells, and the brain’s spatial representation system. *Annu. Rev. Neurosci.* 31:69–89.
- Moser EI, Roudi Y, Witter MP, Kentros C, Bonhoeffer T, Moser MB (2014) Grid cells and cortical representation. *Nature Reviews Neuroscience*.
- O’Keefe J, Burgess N (1996) Geometric determinants of the place fields of hippocampal neurons. *Nature* 381:425–428.
- O’Keefe J, Nadel L (1978) *The hippocampus as a cognitive map* Clarendon Press Oxford.
- Solstad T, Moser EI, Einevoll GT (2006) From grid cells to place cells: a mathematical model. *Hippocampus* 16:1026–1031.