

Path Learning and Path Decoding with Grid Cells and Place Cells

Lijuan Su, Jean-Marc Fellous and Onur Ozan Koyluoglu

April 15, 2016

Abstract

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 from dorsal to ventral levels 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, $M \times M$ 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;
- Does the weight matrix encode the sequences or the frequencies of traversed locations along the path?

2 Models and Methods

2.1 Place Cell Models

In our model, we consider N place cells with a continuously increasing field size across the 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

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

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	Linear (50-625)
N	100

2.2 Grid Cell Models

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 two model configurations: one is with a continuously increasing field size across the ventral axis; the other is with several discrete field sizes across the 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 (Lytle et al., 2013), we have derived the grid cell model as follows:

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

$$\lambda = p1 * sp * sp + p2 * sp + p3 \quad (4)$$

where $\mathbf{s} = (x, y)$ is the location vector (1×2 vector in 2D space), k is the inter-vertex spacing between grid points (in cm), $\mathbf{c} = (x_0, y_0)$ 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 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 a continuously increasing field size across the ventral axis; the other is with several discrete field sizes across the ventral axis. Grid Cell Model configurations are shown in the Table 2.

Table 2: Grid Cell Model Configuration

	Linear	Modular
Phase (s)	Random	Random
Spacing (sp)	Linear (20cm-120cm)	Discrete (38cm,48cm,65cm,98cm)
Oritation (θ)	Random between (0-10)	Discrete (0,3,6,9)
M	100	100

2.3 Overlap of the firing fields between different cells

The firing fields of different cells maybe overlapped, which is related to the weight connection beween these cells. Here we define the overlap of cell i and cell j on the whole maze. To compare the overlaps for different cell pairs, given the firing fields of v^i and v^j , we use three methods to compute one value which represents the overlap of two cells over the whole maze.

Method 1: Compute the Pearson product-moment correlation coefficients of the firing fields of v^i and v^j .

Method 2: Sum of the errors of the firing rates of cell i and cell j on the whole maze, i.e the sum of the elements of o_2^{ij} .

$$o_2^{ij} = \begin{bmatrix} v_{11}^i - v_{11}^j & v_{12}^i - v_{12}^j & \dots & v_{1m}^i - v_{1m}^j \\ v_{21}^i - v_{21}^j & v_{22}^i - v_{22}^j & \dots & v_{2m}^i - v_{2m}^j \\ \vdots & \vdots & \vdots & \ddots \\ v_{m1}^i - v_{m1}^j & v_{m2}^i - v_{m2}^j & \dots & v_{mm}^i - v_{mm}^j \end{bmatrix}$$

Method 3: Sum of the products of the firing rates of cell i and cell j on the whole maze, i.e the sum of the elements of o_3^{ij} .

$$o_3^{ij} = \begin{bmatrix} v_{11}^i * v_{11}^j & v_{12}^i * v_{12}^j & \dots & v_{1m}^i * v_{1m}^j \\ v_{21}^i * v_{21}^j & v_{22}^i * v_{22}^j & \dots & v_{2m}^i * v_{2m}^j \\ \vdots & \vdots & \vdots & \ddots \\ v_{m1}^i * v_{m1}^j & v_{m2}^i * v_{m2}^j & \dots & v_{mm}^i * v_{mm}^j \end{bmatrix}$$

where m is the size of the maze, and v_{xy}^i means the firing rate of cell i at position (x, y) ; and v_{xy}^j means the firing rate of cell j at position (x, y) .

2.4 Random and Reward Path from Experiment and Simulation

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, 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. Path algorithm explaining the movement of a foraging rat is derived from (Hasselmo et al., 2007) and can be summarized by the following equations:

$$\begin{aligned} \Delta x(t) &= S(1 - m)p_x + m\Delta x(t - 1) \\ \Delta y(t) &= S(1 - m)p_y + m\Delta y(t - 1) \end{aligned} \tag{5}$$

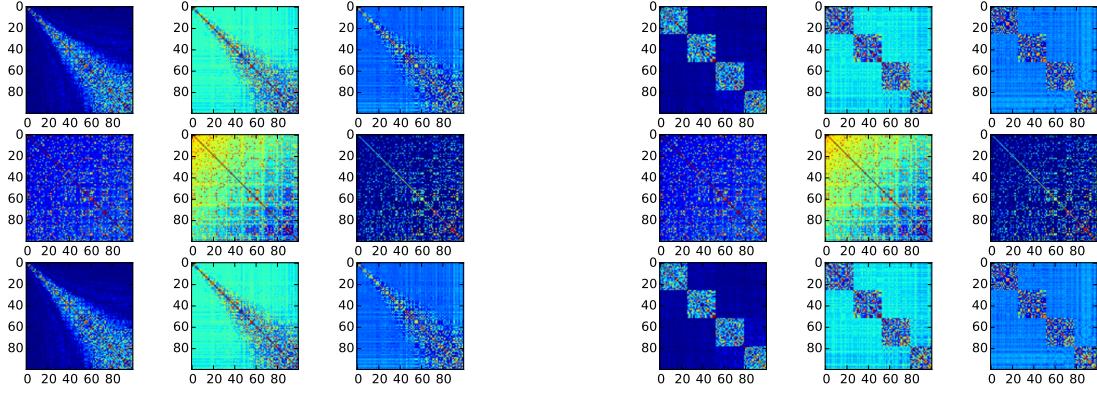


Figure 1: Overlap between Cells: Row1 is GG, Row2 is PP, Row3 is GP; Column1 is Method1, Column2 is Method2, Column3 is Method3.

where $S = 5$, $m = 0.8$, and $p_x, p_y \sim \mathcal{N}(0, 1)$. 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

$$\begin{aligned}\Delta x_r(t) &= -\Delta x(t) \\ \Delta y_r(t) &= -\Delta y(t)\end{aligned}\tag{6}$$

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

The path configurations 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 included one random path and one 3-rewards path and one 4-rewards path.

Table 3: Path Configurations

	Experiment Path	Simulation Path
Random	ExpRan	SimRan
3 Rewards	ExpRew3	SimRew3
4 Rewards		SimRew4

2.5 Sequency and Frequency Representation of Path

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 ??.

2.6 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

$$w_{ij}(t) = w_{ij}(t-1) + \alpha x_i(s(t))x_j(s(t))\tag{7}$$

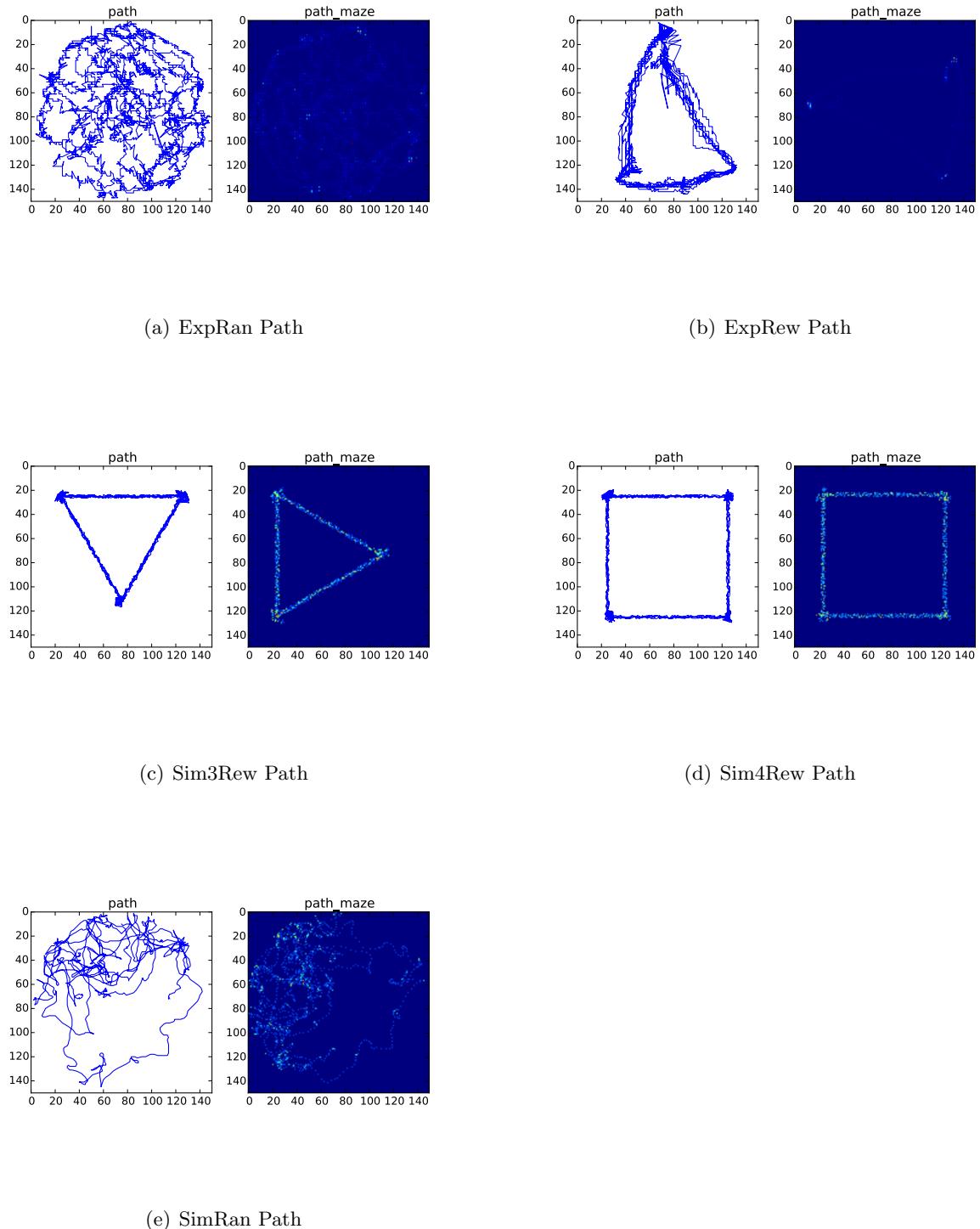


Figure 2: Sequency and Frequency Representation of Path

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 7, 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

$$w_{ij}(t) = \begin{cases} w_{ij}(t-1) + \alpha_{ij}x_i(s(t))x_j(s(t)) & \text{if } x_i(s(t)) \cdot x_j(s(t)) > 0 \\ w_{ij}(t-1) - \gamma_{ij}x_i(s(t)) & \text{if } x_j(s(t)) = 0, x_i(s(t)) \geq 0 \end{cases} \quad (8)$$

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 7 and Equation 8. 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	$\alpha = 0.3, \gamma = 0.6$	$\alpha = 0.3, \gamma = 0.6$	$\alpha_1 = 0.3, \gamma_1 = 0.6; \alpha_2 = 0.03, \gamma_2 = 0.06$

3 Path Learning and Path Decoding Over Navigational Path

3.1 Definition of Weight Matrix on Maze

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

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

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

$$O = \sum w_{ij} * o^{ij}$$

- 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;

3.2 Path Learning and Path Decoding Analysis

Given one path, using different hebbian learning rules, the animal updates the neural connections between grid cells and grid cells (GG), between grid cells and place cells (GP), between place cells to place cells (PP). Here in our paper, we tested two hebbian learning rules: unsupervised hebbian learning rule (U-HL) defined in Equation 7; decay hebbian learning rule (D-HL) defined in Equation 8.

The animal learnt three weight matrices from the navigational path, which are GG, GP and PP. For each weight matrix, using the definition of O we can compute the weight-maze which represents the decoded path from the learnt weight matrix.

As shown in Figure 3: Fig *a* is the path learning and path decoding from linear grid cell model with unsupervised hebbian learning rule; Fig *b* is the path learning and path decoding from linear grid cell model with decay hebbian learning rule; Fig *c* is the path learning and path decoding from module1 cell model with unsupervised hebbian learning rule; Fig *d* is the path learning and path decoding from module1 grid cell model with decay hebbian learning rule; Fig *e* is the path learning and path decoding from module2 grid cell model with unsupervised hebbian learning rule; Fig *f* is the path learning and path decoding from module2 grid cell model with decay hebbian learning rule. In each subfigure: Row 1 is the learnt weight matrix, Row 2 is the decoded path (weighted-maze) from the corresponding learnt weight matrix; Column 1 means path learning and path decoding with GG, Column 2 means path learning and path decoding with GP, Column 3 means path learning and path decoding with PP.

To compare the decoded path from the weight matrix and the actual path, we compute the similarity between them using three methods, same with the computation of the overlap between the firing fields of two cells. Given the sequences representation of one path, we transform the sequences representations to the frequencies representation of the path. And then compare the similarity between the decoded path (weighted-maze) and the freq representation of the actual path.

As shown in Figure 4, Fig a is the comparison with Method1, Fig b the comparison with Method2, and Fig c is the comparison with Method3. For each subfigure: group 1 is the Lineared Grid Cells with U-HL; group 2 is the Lineared Grid Cells with D-HL; group 3 is the Module1 Grid Cells with U-HL; group 4 is the Module2 Grid Cells with D-HL; group 5 is the Module2 Grid Cells with U-HL; group 6 is the Module2 Grid Cells with D-HL. For each group, the decoding results (similarity score) are analysed with GG, GP and PP.

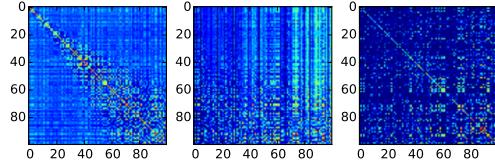
4 Conclusion and Discussion

In this paper we try to decode the learned path from the learned weight matrices, and compare the decoding results with different grid cell models and different learning rules. 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:

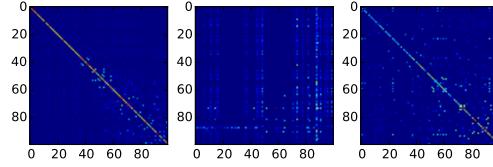
- Which cells are more effective, only grid cells, only place cells, or grid-place cells together?
- Which model is better, linear or modular grid cell model?
- Which learning rule is better, unsupervised or decay Hebbian learning rules?
- Which update strategy is better, the uniformal or different learning rules for cells from one module and cells from two different modules?
- Does the animal remember the sequences or the frequencies of traversed locations along the path?

5 References

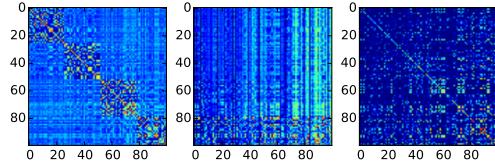
- Brun VH, Solstad T, Kjelstrup KB, Fyhn M, Witter MP, Moser EI, Moser MB (2008) Progressive increase in grid scale from dorsal to ventral medial entorhinal cortex. *Hippocampus* 18:1200–1212.
- Burak Y, Fiete IR (2009) Accurate path integration in continuous attractor network models of grid cells. *PLoS 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.



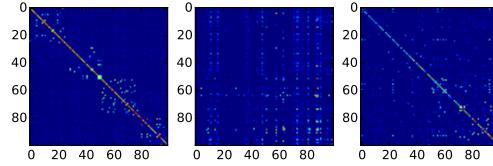
(a) Linear Grid Cells with U-HL



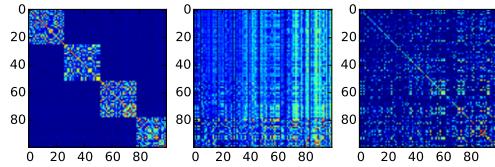
(b) Linear Grid Cells with D-HL



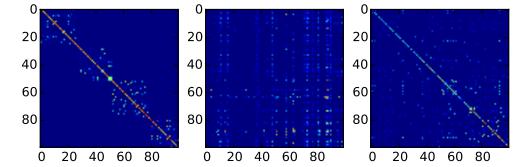
(c) Module1 Grid Cells with U-HL



(d) Module1 Grid Cells with D-HL



(e) Module2 Grid Cells with U-HL



(f) Module2 Grid Cells with D-HL

Figure 3: Path Learning and Path decoding on ExpRan Path.

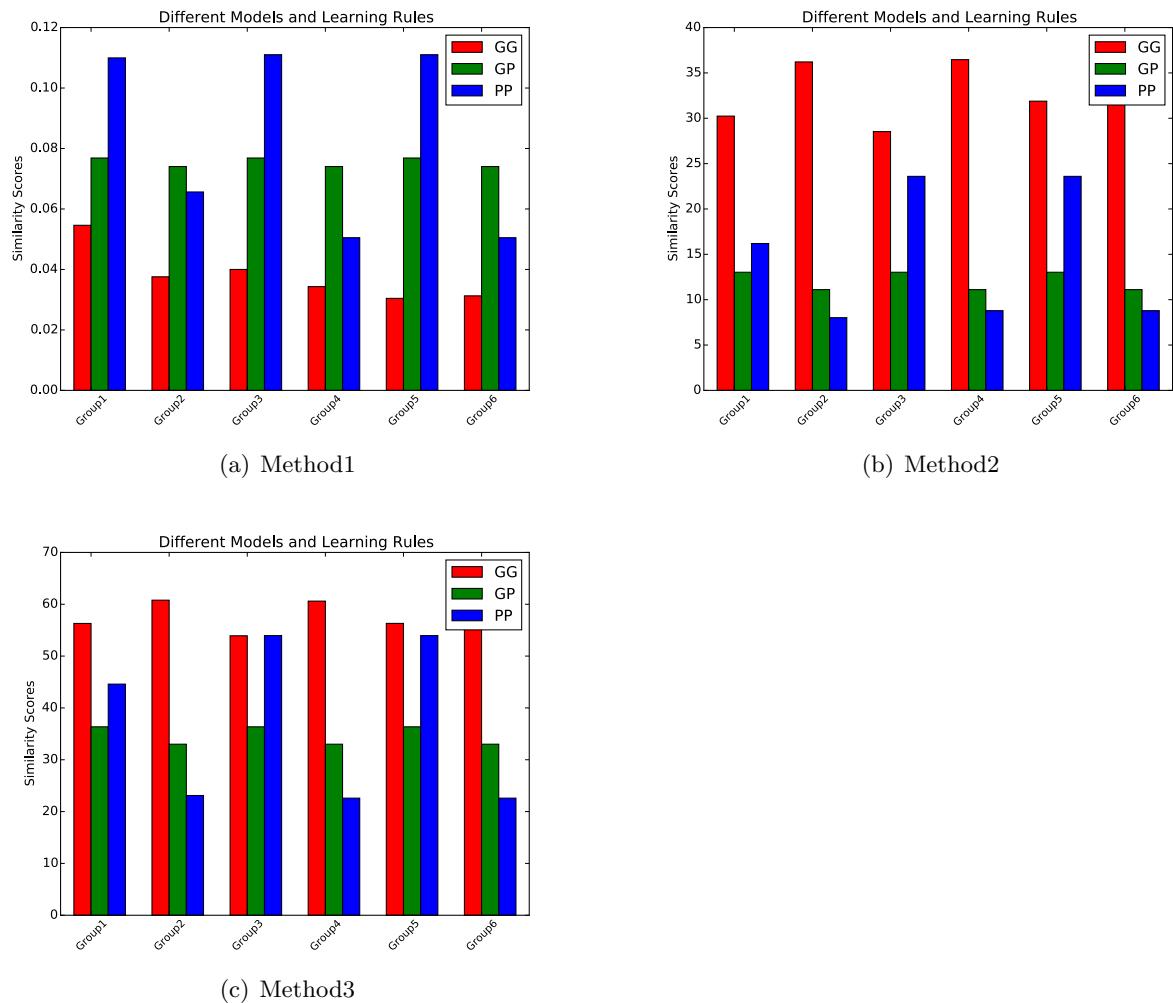
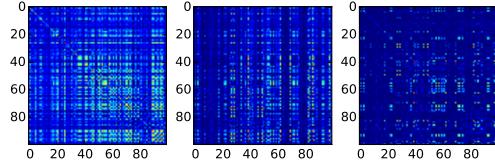
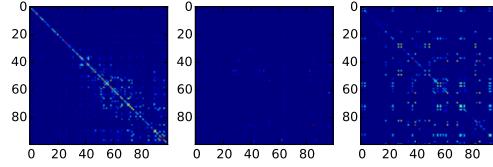


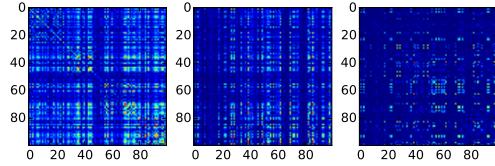
Figure 4: Comparison between Decoded Path and Actual Path on ExpRan Path.



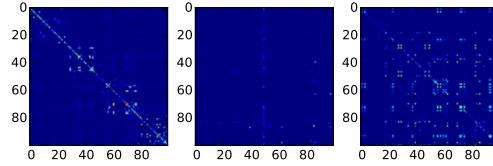
(a) Linear Grid Cells with U-HL



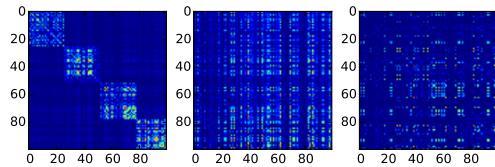
(b) Linear Grid Cells with D-HL



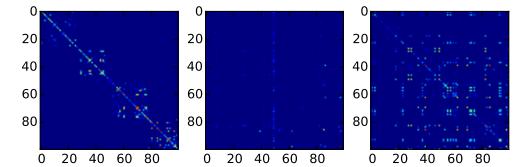
(c) Module1 Grid Cells with U-HL



(d) Module1 Grid Cells with D-HL



(e) Module2 Grid Cells with U-HL



(f) Module2 Grid Cells with D-HL

Figure 5: Path Learning and Path decoding on ExpRew Path.

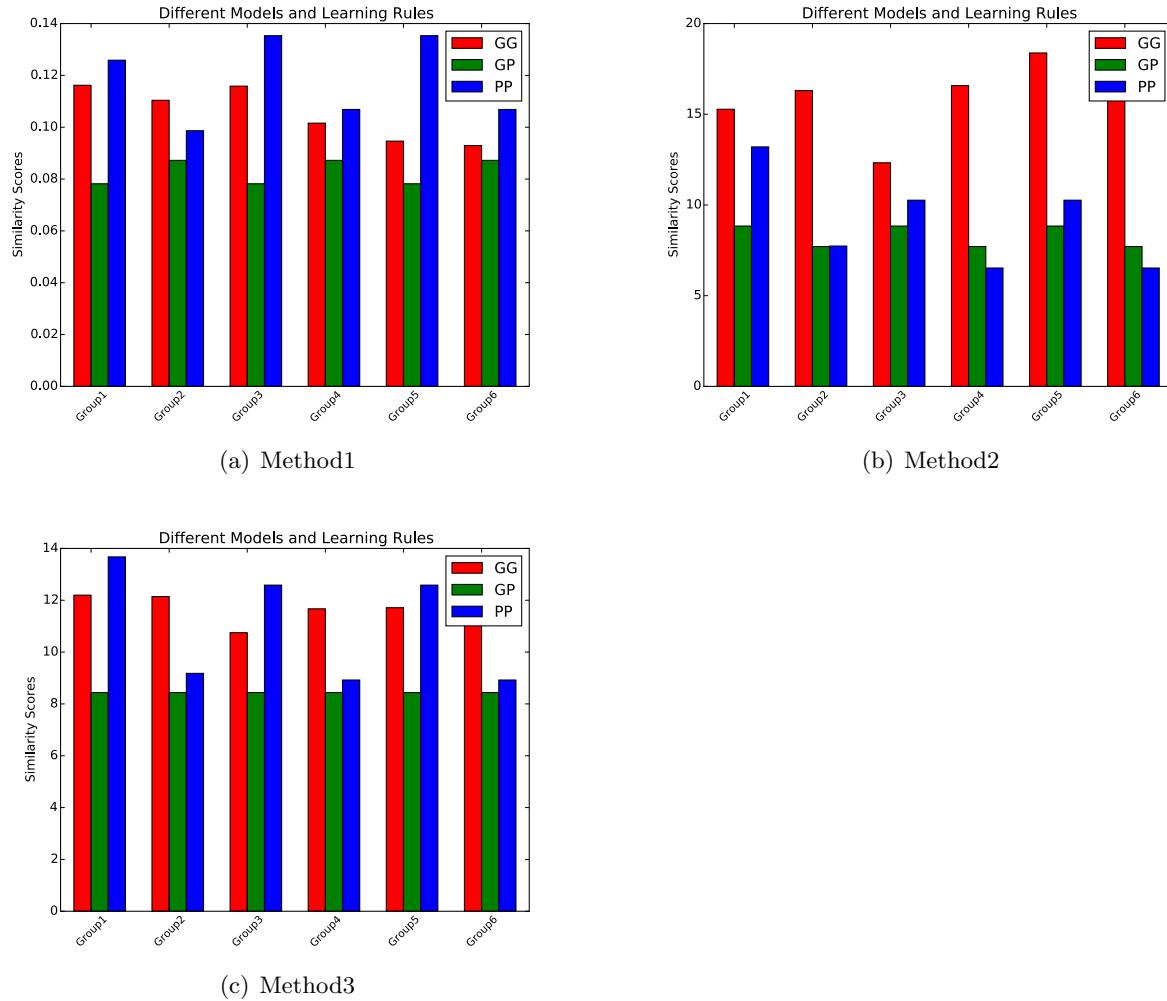
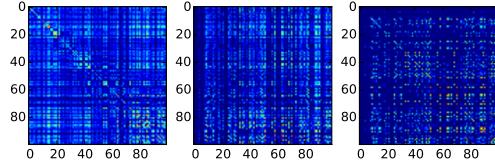
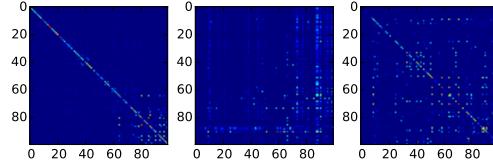


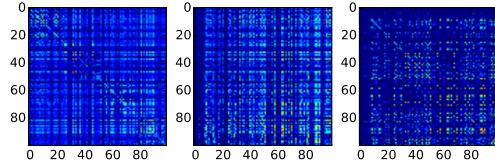
Figure 6: Comparison between Decoded Path and Actual Path on ExpRew Path.



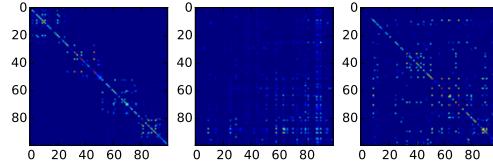
(a) Linear Grid Cells with U-HL



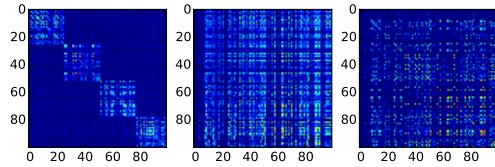
(b) Linear Grid Cells with D-HL



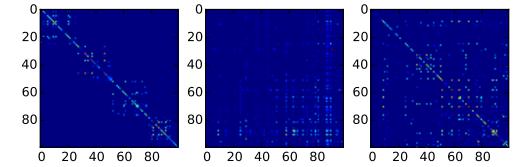
(c) Module1 Grid Cells with U-HL



(d) Module1 Grid Cells with D-HL



(e) Module2 Grid Cells with U-HL



(f) Module2 Grid Cells with D-HL

Figure 7: Path Learning and Path decoding on SimRew3 Path.

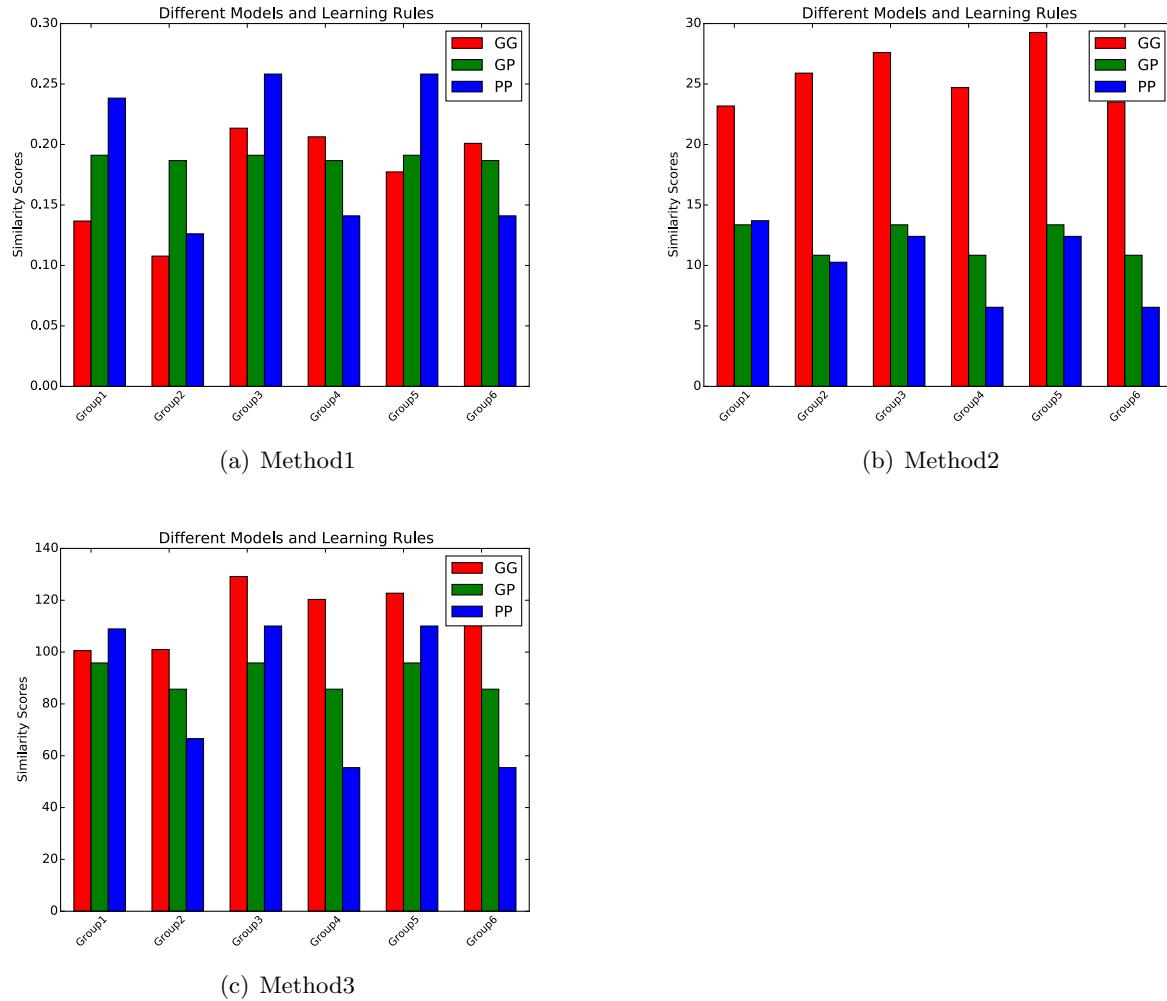
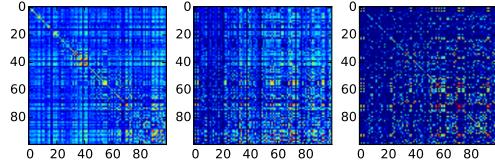
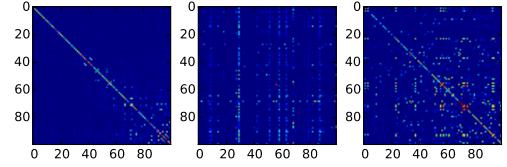


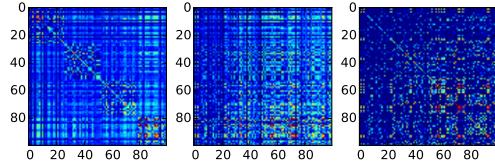
Figure 8: Comparison between Decoded Path and Actual Path on SimRew3 Path.



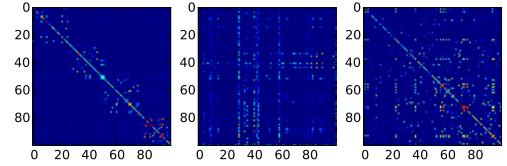
(a) Linear Grid Cells with U-HL



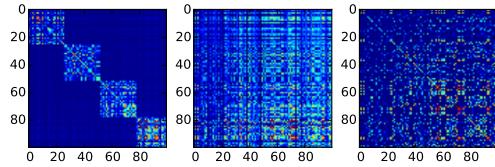
(b) Linear Grid Cells with D-HL



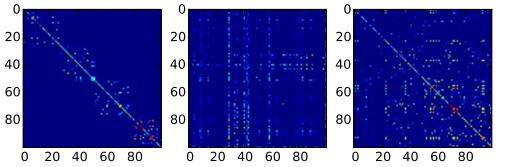
(c) Module1 Grid Cells with U-HL



(d) Module1 Grid Cells with D-HL



(e) Module2 Grid Cells with U-HL



(f) Module2 Grid Cells with D-HL

Figure 9: Path Learning and Path decoding on SimRew4 Path.

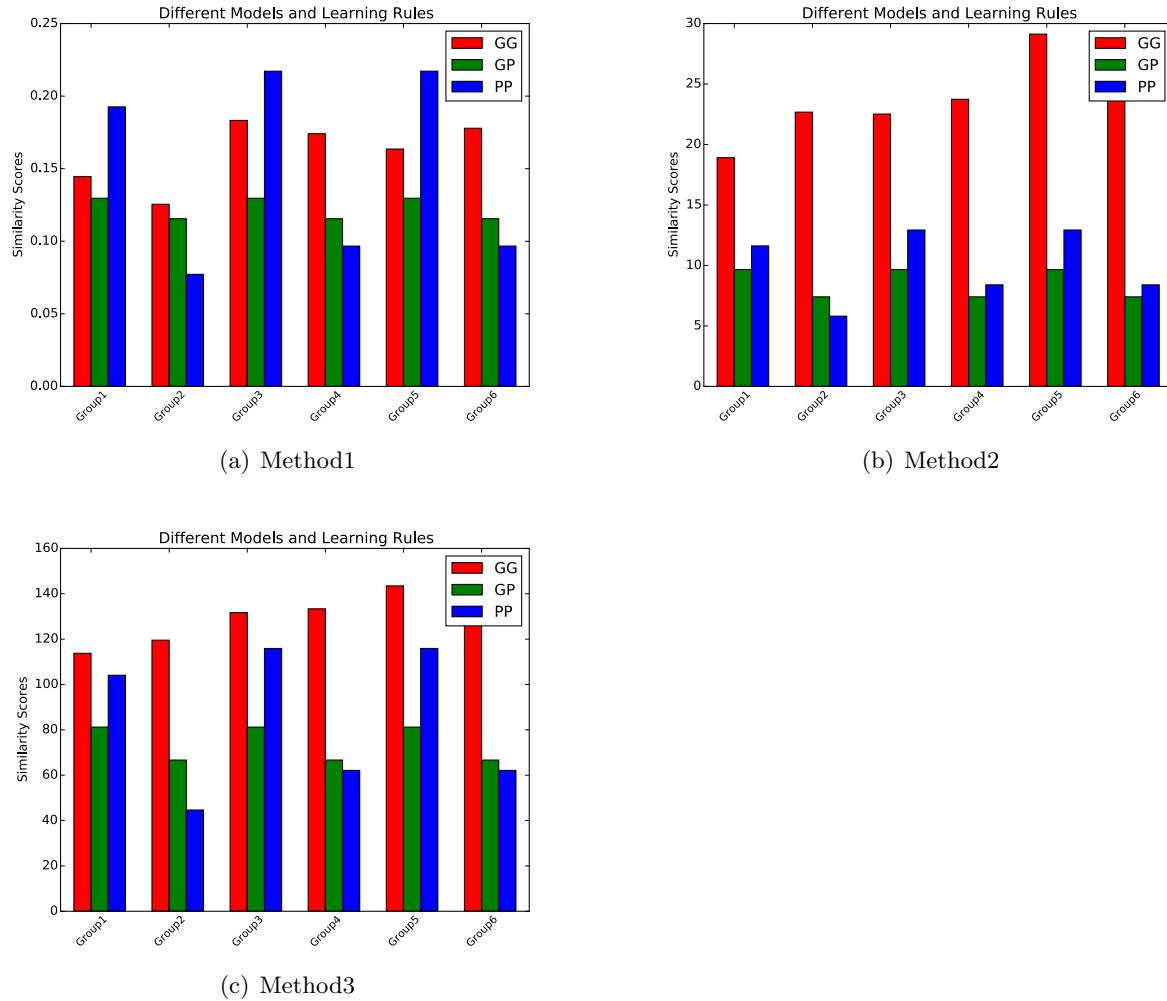
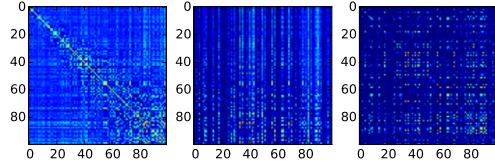
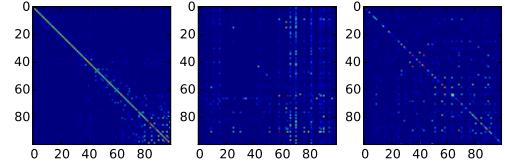


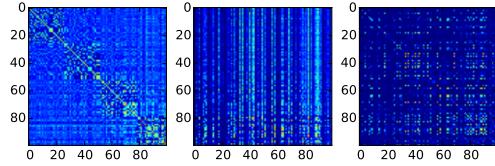
Figure 10: Comparison between Decoded Path and Actual Path on SimRew4 Path.



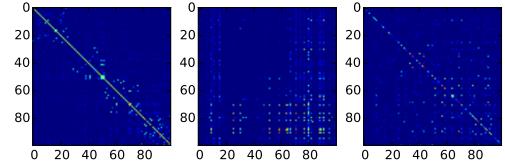
(a) Linear Grid Cells with U-HL



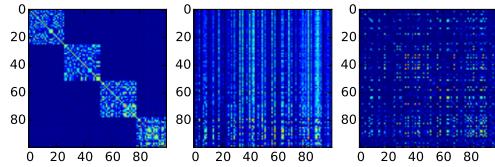
(b) Linear Grid Cells with D-HL



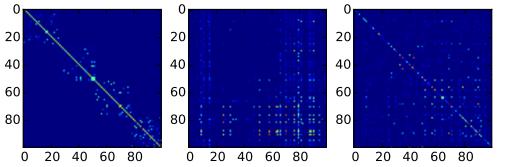
(c) Module1 Grid Cells with U-HL



(d) Module1 Grid Cells with D-HL



(e) Module2 Grid Cells with U-HL



(f) Module2 Grid Cells with D-HL

Figure 11: Path Learning and Path decoding on SimRan Path.

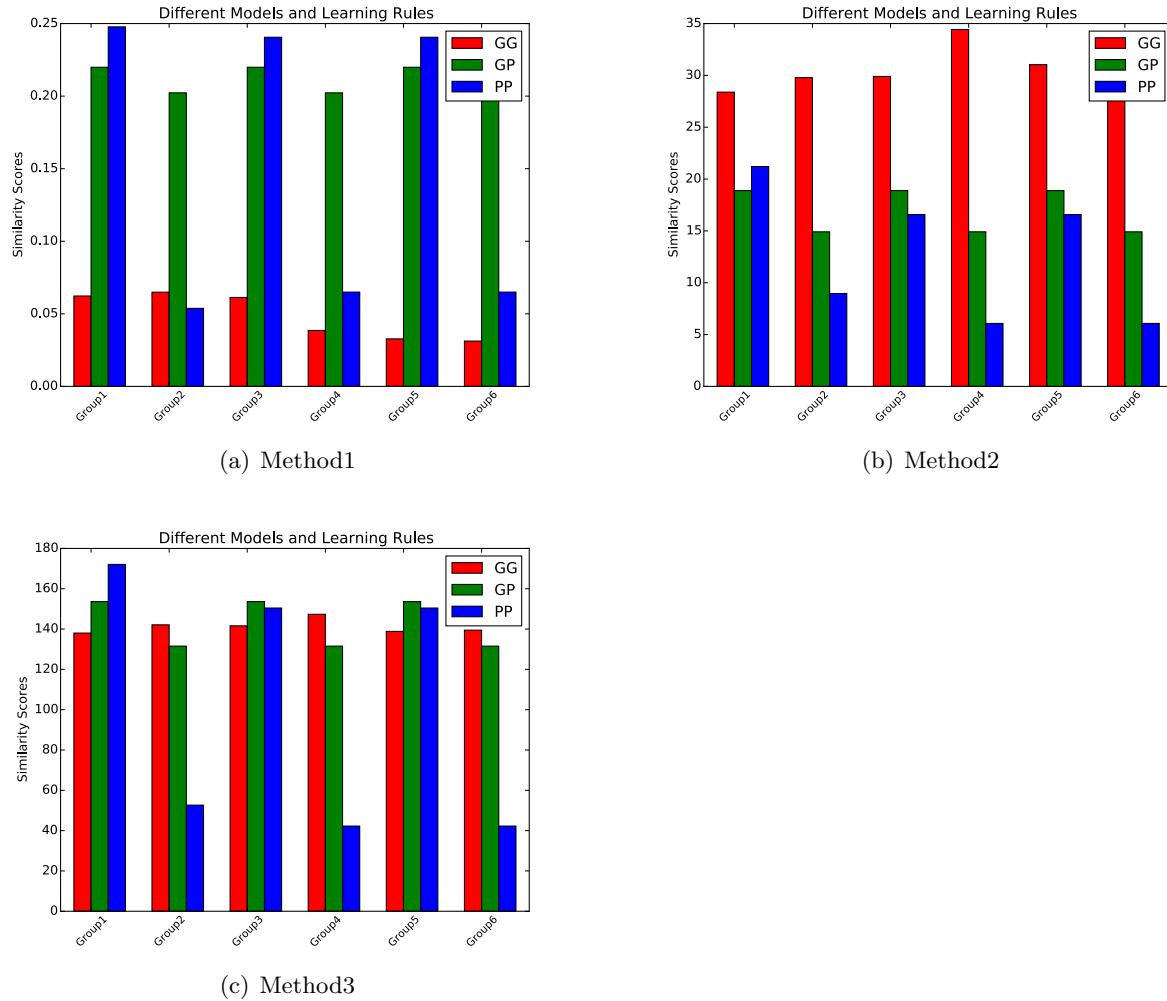


Figure 12: Comparison between Decoded Path and Actual Path on SimRan Path.