



A computational model for spatial cognition combining dorsal and ventral hippocampal place field maps: multiscale navigation

Pablo Scleidorovich¹ · Martin Llofriu^{1,2} · Jean-Marc Fellous³ · Alfredo Weitzenfeld¹ 

Received: 29 May 2019 / Accepted: 22 November 2019 / Published online: 9 January 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

Classic studies have shown that place cells are organized along the dorsoventral axis of the hippocampus according to their field size, with dorsal hippocampal place cells having smaller field sizes than ventral place cells. Studies have also suggested that dorsal place cells are primarily involved in spatial navigation, while ventral place cells are primarily involved in context and emotional encoding. Additionally, recent work has shown that the entire longitudinal axis of the hippocampus may be involved in navigation. Based on the latter, in this paper we present a spatial cognition reinforcement learning model inspired by the multiscale organization of the dorsal–ventral axis of the hippocampus. The model analyzes possible benefits of a multiscale architecture in terms of the learning speed, the path optimality, and the number of cells in the context of spatial navigation. The model is evaluated in a goal-oriented task where simulated rats need to learn a path to the goal from multiple starting locations in various open-field maze configurations. The results show that smaller scales of representation are useful for improving path optimality, whereas larger scales are useful for reducing learning time and the number of cells required. The results also show that combining scales can enhance the performance of the multiscale model, with a trade-off between path optimality and learning time.

Keywords Hippocampus · Reinforcement learning · Spatial cognition

Communicated by Jean-Marc Fellous.

This article is part of the Special Issue entitled ‘Complex Spatial Navigation in Animals, Computational Models and Neuro-inspired Robots’.

✉ Pablo Scleidorovich
pablos@mail.usf.edu

Martin Llofriu
mllofriualon@mail.usf.edu

Jean-Marc Fellous
fellous@email.arizona.edu

Alfredo Weitzenfeld
aweitzenfeld@usf.edu

¹ Department of Computer Science and Engineering,
University of South Florida, Tampa, FL, USA

² Department of Computer Science and Engineering,
Universidad de la Republica, Montevideo, Uruguay

³ Department of Psychology, University of Arizona, Tucson,
AZ, USA

1 Introduction

Over the last few decades, much attention has been paid to understanding the roles and computational mechanisms of the hippocampus and medial entorhinal cortex in the context of spatial cognition. O’Keefe and Dostrovsky (1971) reported the existence of place cells in the hippocampus. Place cells are hippocampal neurons whose activation pattern is highly correlated with the position of the animal in an allocentric frame of reference (O’Keefe and Nadel 1978; McNaughton et al. 1996). Additional cells have been reported to be involved in spatial cognition. Head direction cells fire when rats are oriented in a particular allocentric direction (Taube et al. 1990; Chen et al. 1994). Grid cells activate regularly over 2D space forming an hexagonal lattice that is believed to support path-integration-like processes (Fyhn et al. 2004; Hafting et al. 2005). Border cells have been found to activate near the borders of environments independently of possible rescalings (Savelli et al. 2008; Solstad et al. 2008). Object-vector cells fire when obstacles are found at a given distance and direction from the rat, thus encoding “vectors” to nearby obstacles (Hoydal et al. 2018). Although a lot of

progress has been made since the initial findings, much is yet to be understood on how all these cell types work and how they come together to support complex and ethologically realistic spatial navigation tasks.

In this work, we focus our attention on findings that show that the brain uses multiple scales of representations to encode the environment. More specifically, it has been observed that the spatial specificity of place cells decreases along the dorsoventral axis of the hippocampus, with dorsal place cells having smaller more stable fields than ventral cells (Jung et al. 1994; Kjelstrup et al. 2008; Maurer et al. 2005). Broadly accepted theories suggest that the dorsal hippocampus is mainly involved in spatial learning, while the ventral hippocampus participates in motivation-related processes (Fanselow and Dong 2010; Harland et al. 2017; Poppenk et al. 2013; Strange et al. 2014). Recent work has, however, extended this view, suggesting that in certain conditions, ventral hippocampal neurons may also be involved in spatial navigation (Contreras et al. 2018; Harland et al. 2017; de Hoz et al. 2003).

Based on the observed scale differences between the dorsal and ventral hippocampus, we present a computational model for multiscale spatial cognition, where we hypothesize the benefits of exploiting multiple place cell activation field sizes to reduce both learning time and the total number of cells. The model uses place cells to encode the state in a reinforcement learning algorithm which learns how to navigate to a goal from a set of different starting locations. Computationally, the model uses hierarchical radial basis functions (HRBF) with reinforcement learning methods to approximate the state and action value functions. The term “hierarchical” refers to the way in which the radial basis functions are organized into uniform layers according to their size. To our knowledge, HRBFs have been used mainly in the context of surface reconstruction problems such as in Bellocchio et al. (2007) and Pouderoux et al. (2004). In the context of reinforcement learning algorithms, HRBFs have also been used in the previous version of our model described in Llofriu et al. (2015).

There exist other models of multiscale levels of place cell representation in the hippocampus. In general, these models concentrate on understanding the multiscale nature of place cell representation, but are more limited in their investigation of how multiple layer sizes may impact goal-oriented spatial navigation. In the rest of this section, we summarize such models and discuss the main differences with our work.

Lyttle et al. (2013) extended a model of the medial entorhinal cortex (MEC) and the hippocampus from de Almeida et al. (2009). The model was used to provide a possible explanation for the difference in properties observed between dorsal and ventral place cells; however, the model was not used for spatial cognition and no attempt was made to explain how the different field sizes are used in this context.

Erdem and Hasselmo (2012, 2014) describe a navigation model based on the entorhinal cortex and the hippocampus of rats. The original model used forward look-ahead probes during periods of inactivity reminiscent of sharp-wave ripples (Jadhav et al. 2012; Leonard et al. 2015). Much like sharp-wave ripples are believed to guide navigation (Johnson and Redish 2007; Pfeiffer and Foster 2013), the probes assessed potential paths that the rat could take and used them to choose an action. In their latter model, Erdem and Hasselmo added multiscale representation levels to extend the range of the look-ahead probes improving the ability to reach the goal. While their work shows how multiscale can benefit a model using forward look-ahead probes, our work assesses the benefits within a reinforcement learning architecture during goal-oriented tasks.

Llofriu et al. (2015) developed a multiscale model for spatial cognition based on the differences between place field sizes and the dorsoventral axis of the hippocampus. The model was assessed in an open-field maze in simulated and real environments. Their research compared the performance of the multiscale model with the performance of using a single scale in terms of the number of steps and learning time. In contrast to their work, our work focuses in understanding the contributions that each scale provides to the multiscale model. Furthermore, we perform a more extensive parameter analysis and assess results as a function of maze complexity.

Llofriu et al. (2019) extended their previous model to reproduce the results obtained from rat experiments described by Contreras et al. (2018). In the experiments, it was observed that inactivating either ventral or dorsal hippocampus significantly reduced the performance of rats in an open-field maze only when obstacles were introduced.

Finally, although not necessarily a model of the hippocampus, it is worth noting that Fan et al. (2017) developed an adaptive multiscale visual place recognition system inspired by grid cells found in the entorhinal cortex. Their system is capable of adaptively selecting scale sizes based on an optimized distance metric rather than using prefixed values. In the tested datasets, the adaptive system resulted in better recall performance rates when compared to previous fixed multiscale approaches and outperforms state of the art robotic navigation algorithms.

Our work is based on the unknown nature of the readout of the multiscale representation found in the longitudinal axis of the hippocampus. How is the information combined and used? What functional consequences might this combination have in the context of complex spatial navigation? The model we developed is used to assess the possible contributions that each scale might provide to the multiscale model. We assess contributions in terms of learning speed and solution optimality (i.e., the number of actions required to solve a task). Also, we compare the results in multiple mazes of different complexities. Much like small scales are used to capture small

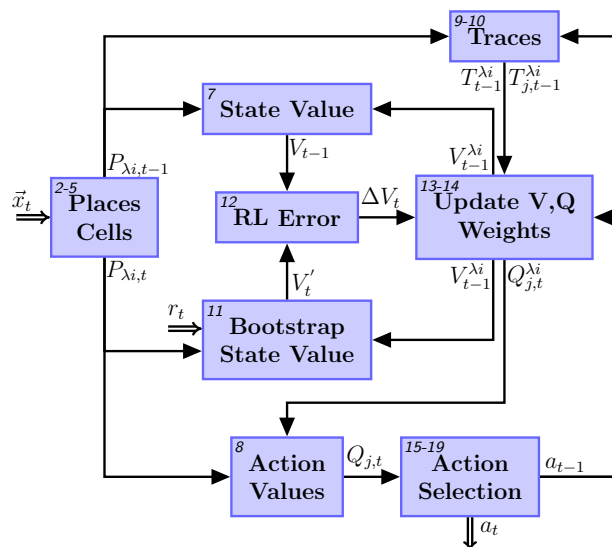


Fig. 1 Algorithm summary. Summary of the computations performed during each cycle. The diagram on the left shows the flow of information (dependencies) between components. The pseudocode on the right shows the execution order. In the diagram, the numbers on the upper

```

t ← t + 1;
Get position and reward ( $\vec{x}_t, r_t$ );
Compute place cells ( $P_{\lambda i, t}$ );
if t > 0 then
    Compute bootstrap ( $V'_t$ );
    Compute RL Error ( $\Delta V_t$ );
    Update V,Q weights ( $V_t^{\lambda i}, Q_{j, t}^{\lambda i}$ );
end
Compute the state value ( $V_t$ );
Compute the action values ( $Q_{j, t}$ );
Perform action selection ( $a_t$ );
Update traces ( $T_t^{\lambda i}, T_{j, t}^{\lambda i}$ );
Perform action ( $a_t$ );

```

left corner of each box indicate the associated equations. Also, double arrows show the inputs and outputs of the model, and subindexes t and $t - 1$ indicate whether the dependency was computed in the current or previous cycle, respectively

details in surface reconstruction and large scales provide the general shape (Ferrari et al. 2005), we hypothesize that in our spatial cognition model, smaller scales of representation will be most useful when precision is required, while larger scales will be most useful to accelerate learning and reduce the number of neurons required to solve a task.

In the remainder of the paper, Sect. 2 presents the computational model, Sect. 3 the evaluation method, Sect. 4 the experiments and results, and Sect. 5 the conclusions and discussion.

2 Model

The model is implemented as an actor critic algorithm (Sutton et al. 2000) with discrete action space and continuous state space encoded by the activity of hippocampal place cells of different field sizes.

The following subsections describe the algorithm in detail, with each subsection corresponding to the elements from Fig. 1 which shows a summary of the algorithm.

2.1 Action space (a_t)

In each cycle, the model performs one of eight possible actions denoted by the angle of travel direction θ_j . Each θ_j represents moving a single step (8 cm) in a specific allocentric direction given by Eq. (1) and shown in Fig. 2. The action taken at time t is denoted by a_t , where $a_t = j$ corresponds to the rat having moved one step in direction θ_j at time t .

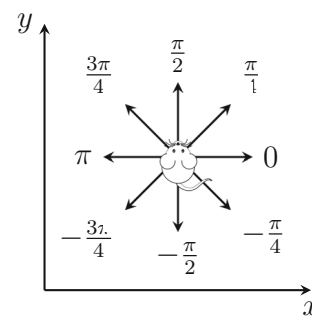


Fig. 2 Action space. The figure shows the eight possible allocentric directions in which the rat can move

$$\theta_j = \frac{\pi}{4} j \quad \forall j : 0 \leq j < 8 \quad (1)$$

2.2 Place cells ($P_{\lambda i, t}$)

In our model, place cells are organized into layers according to their field size so that all cells within a given layer have equal field size. The number of layers, along with their respective field size, and the number of cells vary according to the experiment being performed. Figure 3 illustrates the organization of two sample layers used in the experiments.

Each layer's place cells are distributed uniformly over space according to a rectangular grid. To do so, we define the distance d_λ between consecutive place cells according to

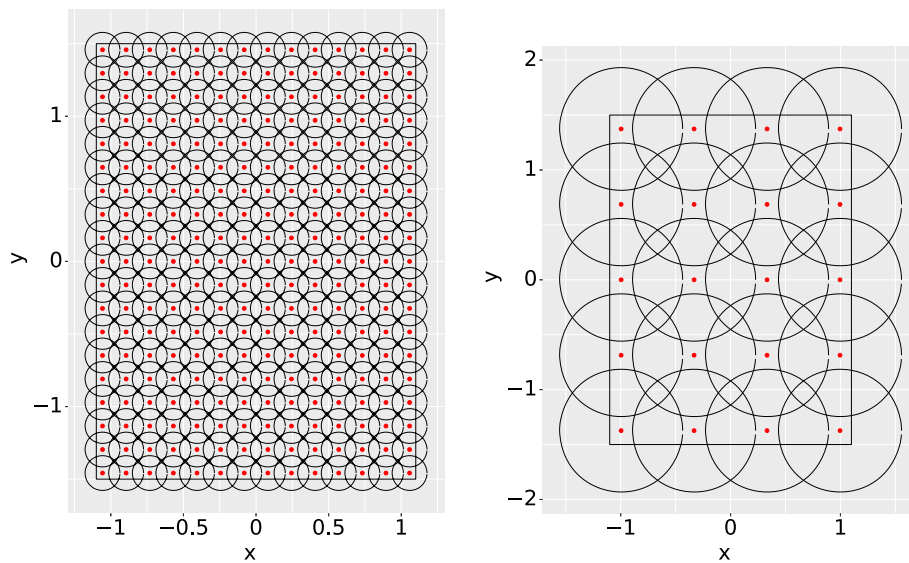


Fig. 3 Sample PC layers. The image shows two sample PC layers used in the experiments. Circles indicate the boundary of each place field, while red dots indicate their centers. In both cases, the maze's width and height measure 2.2 m and 3 m, respectively. The layer in the left figure has parameters $r_\lambda = 0.12$ m (place cell radius) and $n_{\lambda x} = 14$

(the number of columns). The layer in the right figure has parameters $r_\lambda = 0.56$ m and $n_{\lambda x} = 4$. In both cases, the number of rows is proportional to the number of columns multiplied by the aspect ratio $\frac{h}{w}$ (color figure online)

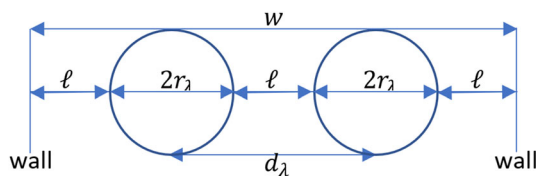


Fig. 4 Derivation of d_λ . The figure shows how to derive d_λ and the positions of the cells. When cells cannot cover the full maze (i.e., when $2r_\lambda n_{\lambda x} < w$), they are distributed so that all gaps are of equal length ℓ , which leads to $\ell = (w - 2r_\lambda n_{\lambda x}) / (n_{\lambda x} + 1)$ and $d_\lambda = \ell + 2r_\lambda$. The position of the first cell is given by $\ell + r_\lambda$. On the other hand, when cells can cover the full width (i.e., when $2r_\lambda n_{\lambda x} > w$), then ℓ becomes negative, in which case instead of interpreting ℓ as the size of the gaps, it can be interpreted as the size of the overlap between consecutive cells and as the size of the overlaps between the first and final cells with the outside of the maze

Eq. (2). In Eq. (3), we use the resulting distance along with row and column numbers to define the center of each place cell. Figure 4 illustrates how these equations are derived.

$$d_\lambda = \frac{w + 2r_\lambda}{n_{\lambda x} + 1} \quad (2)$$

$$\vec{x}_{\lambda i} = d_\lambda \left[\frac{i \% n_{\lambda x} + 1}{i / n_{\lambda x} + 1} \right] - \left[\frac{r_\lambda + \frac{w}{2}}{r_\lambda + \frac{w}{2}} \right] \quad \forall_i : 0 \leq i < n_\lambda \quad (3)$$

where

- d_λ is the distance between consecutive place cells in layer λ ,
- r_λ is the radius of place cells in layer λ ,

- $n_{\lambda x}$ is a parameter that indicates the number of columns of the grid for layer λ ,
- w and h are the width and length of the maze (2.2 m and 3 m for all experiments),
- $\vec{x}_{\lambda i}$ is the center of place cell i in layer λ ,
- $n_\lambda = n_{\lambda x} \lfloor n_{\lambda x} \frac{w}{h} \rfloor$ indicates the total number of cells in layer λ , and
- % and / indicate the modulo and integer division operations.

The activation of each place cell is modeled using Gaussian radial basis functions whose values are set to 0 for all points outside of a given radius from their centers (O'Keefe and Burgess 1996). To calculate the activation, we calculate the distance from the rat's position to the place cell's center according to Eq. (4). If the distance is larger than the cell's radius, then the activation is set to 0, otherwise it is given by the Gaussian kernel as shown in Eq. (5), as illustrated in Fig. 5.

$$d_{\lambda i, t} = \|\vec{x}_t - \vec{x}_{\lambda i}\| \quad (4)$$

$$P_{\lambda i, t} = \begin{cases} 0 & d_{\lambda i, t} > r_\lambda \\ e^{-\left(\frac{d_{\lambda i, t}}{r_\lambda}\right)^2 \ln(\alpha)} & \text{otherwise} \end{cases} \quad (5)$$

where

- \vec{x}_t is the rat's position at time t ,
- $d_{\lambda i, t}$ is the distance from $\vec{x}_{\lambda i}$ to \vec{x}_t ,

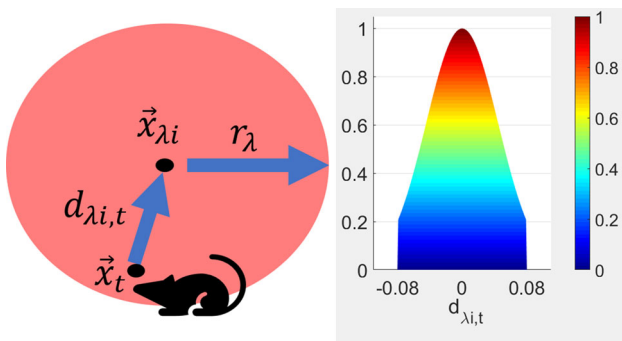


Fig. 5 Place cells. Left: variables involved in computing the activation of a place cell (Eq. 4–5). Right: activation of a place cell as a function of $d_{\lambda i, t}$

- $P_{\lambda i, t}$ is the activation of place cell i from layer λ at time t ,
- r_λ is the radius of place cells in layer λ , and
- α is a constant (0.2) that represents the activation value of a place field at its border.

2.3 State and action values (V_t , $Q_{j, t}$)

The state and action value functions are computed using linear function approximations (Sutton and Barto 2018), using normalized Gaussian hierarchical radial basis functions according to Eq. (6). Note that if the radial basis functions were not normalized, then each basis function will generate a local gradient pointing toward the center of the function, and the value function would be prone to have local optima, as discussed by Kretchmar and Anderson (1997).

$$P'_{\lambda i, t} = \frac{P_{\lambda i, t}}{\sum_{\alpha k} P_{\alpha k, t}} \quad (6)$$

where

- $P'_{\lambda i, t}$ is the normalized activity of place cell i from layer λ at time t .

To compute the state and action value functions at time t , each place cell is associated a value $V_t^{\lambda i}$, and each (place cell, action) pair is associated a value $Q_{j, t}^{\lambda i}$. The state value and action value functions are given by Eqs. (7)–(8). The corresponding associations are illustrated in Fig. 6.

$$V_t = \sum_{\lambda i} V_t^{\lambda i} P'_{\lambda i, t} \quad (7)$$

$$Q_{j, t} = \sum_{\lambda i} Q_{j, t}^{\lambda i} P'_{\lambda i, t} \quad (8)$$

where

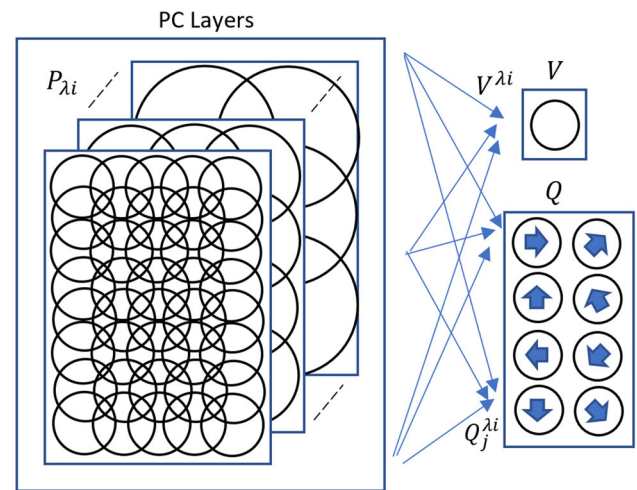


Fig. 6 Neural network architecture. All place cell layers are fully connected to V and Q by the weights $V^{\lambda i}$ and $Q^{\lambda j}$

- V_t is the state value at time t ,
- $Q_{j, t}$ is the action value for action j at time t ,
- $V_t^{\lambda i}$ is the value associated with place cell i from layer λ at time t , and
- $Q_{j, t}^{\lambda i}$ is the value associated with place cell i , action j from layer λ at time t .

2.4 Eligibility traces ($T_t^{\lambda i}$, $T_{j, t}^{\lambda i}$)

To accelerate learning, we use exponentially decreasing eligibility traces given by Eqs. (9)–(10), which are an adaptation from Llofriu et al. (2019) to normalized radial basis functions. Traces are initialized to 0 when $t = 0$.

$$T_t^{\lambda i} = \max \left\{ \alpha_\lambda T_{t-1}^{\lambda i}, P'_{\lambda i, t} \right\} \quad (9)$$

$$T_{j, t}^{\lambda i} = \max \left\{ \alpha_\lambda T_{j, t-1}^{\lambda i}, \delta_j^{a_t} P'_{\lambda i, t} \right\} \quad (10)$$

where

- $T_t^{\lambda i}$ is the trace at time t for cell i from layer λ ,
- $T_{j, t}^{\lambda i}$ is the trace at time t for cell i from layer λ for action j ,
- α_λ is a decay parameter for layer λ whose value depends on the experiment being performed, and
- $\delta_j^{a_t}$ is the Kronecker delta function, whose value is equal to 1 when $a_t = j$ (i.e., if the action taken at time t was action j) and 0 otherwise.

2.5 RL error and update (ΔV_t)

After performing action a_{t-1} , receiving reward r_t , and reaching \vec{x}_t , our model calculates the reinforcement learning error

by first performing a one-step bootstrap of the value function according to Eq. (11) and then calculating the time difference error according to Eq. (12) (Sutton and Barto 2018).

$$V'_t = r_t + \gamma \sum_{\lambda i} V_{t-1}^{\lambda i} P'_{\lambda i, t} \quad (11)$$

$$\Delta V_t = V'_t - V_{t-1} \quad (12)$$

where

- γ is RL's discount factor (constant set to 0.999),
- r_t is the reward received after performing action a_{t-1} . Note that $r_t = 1$ only when the rat reaches the food, i.e., when it reaches a distance of 0.1 m or less from the feeder. For all other times $r_t = 0$.
- ΔV_t is the error of the estimation of the value function in the previous cycle computed at time t .

The state and action values associated with each cell are updated according to Eqs. (13)–(14), which are an adaptation of the update for linear methods from Sutton and Barto (2018), using radial basis functions, bootstrapping, and eligibility traces.

$$V_t^{\lambda i} = V_{t-1}^{\lambda i} + \alpha T_{t-1}^{\lambda i} \Delta V_t \quad (13)$$

$$Q_{j,t}^{\lambda i} = Q_{j,t-1}^{\lambda i} + \alpha T_{j,t-1}^{\lambda i} \Delta V_t \quad (14)$$

where

- α is a constant learning rate (set to 0.6).

Note that the update shown in Eq. (13) is only computed if the action performed a_{t-1} was optimal (in other words, if it is the on-policy action), or if the error is positive, i.e., the action result was better than the currently expected value. Here, we define an action to be optimal if it has the highest action value $Q_{j,t}^{\lambda i}$ among the actions that can be performed, i.e., actions that are not blocked by an obstacle.

2.6 Action selection

In each execution cycle, the model chooses an action randomly from a distribution derived from the action values. This four-step process is summarized in Fig. 7. To derive the distribution, the model computes an initial set of probabilities by applying the softmax() function to the action values according to Eq. (15).

$$p_t^0(j) = \frac{e^{Q_{j,t}}}{\sum_k e^{Q_{k,t}}} \quad (15)$$

where

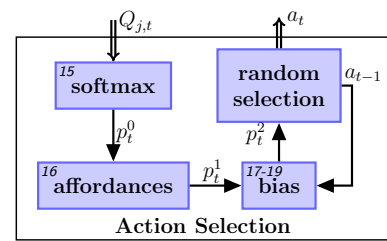


Fig. 7 Action selection process. The image shows a summary of the action selection process. The numbers on the upper left corner of each box indicate the associated equations. As in Fig. 1, arrows show the flow of information between components

- $p_t^0(j)$ is the probability of performing action j after applying softmax().

The model uses the concept of affordances (Guazzelli et al. 1998) to prevent the rat from choosing actions which are not possible to perform such as going through a wall. To do so, we define an action to be impossible if the distance to the closest object in the respective direction is smaller than a given constant (which depends on robot size and step size). The concept is illustrated in Fig. 8. Based on this definition, the probability of impossible actions is set to 0 and the result renormalized. If, in the process, all probabilities become 0, then the result is replaced with a uniform distribution among possible actions. In either case, the result is given by Eq. (16). This also allows for smooth, non-chaotic behavior that privileges continuity of movement over abrupt changes in direction.

$$p_t^1(j) = \begin{cases} \frac{\omega_j^1}{\sum_k \omega_k^1} & \sum_k \omega_k^1 p_t^0(k) = 0 \\ \frac{\omega_j^1 p_t^0(j)}{\sum_k \omega_k^1 p_t^0(k)} & \text{otherwise} \end{cases} \quad (16)$$

where

- $p_t^1(j)$ is the probability after removing impossible actions and renormalizing and
- ω_j^1 is a Boolean value which is 0 if action θ_j cannot be performed and 1 otherwise.

Finally, to prevent the rat from performing undesired trajectories, e.g., continuously moving one step forward and one step backward, the probabilities are transformed before randomly choosing the action. To do so, each action is assigned a biasing weight based on its similarity to the previous action. Similar actions are assigned large weights, while different actions are assigned small weights as illustrated in Fig. 9. This assignment is carried out by rotating a fixed array which represents a density distribution with a single peak at index 0 according to Eq. (17).

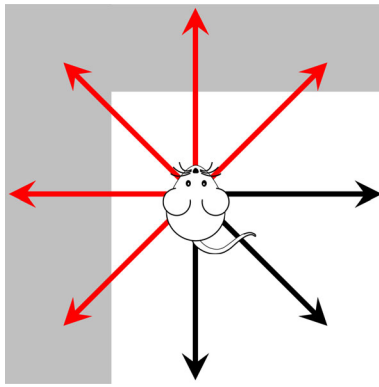


Fig. 8 Affordances. Red arrows indicate impossible actions (e.g., obstructed), while black arrows indicate possible ones (color figure online)

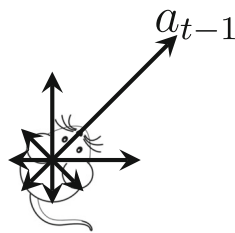


Fig. 9 Bias assignment. Actions similar to a_{t-1} are assigned large weights, while opposing actions are assigned smaller weights

$$b(j) = B((j - a_{t-1}) \% 8) \quad (17)$$

where

- $b(j)$ is bias assigned to action θ_j and
- B is the fixed array for assigning weights with values $[0.83, 0.06, 0.01, 0.01, 0.01, 0.01, 0.01, 0.06]$.

Then, to allow the rat to exploit the policy learned by the algorithm, as the number of trials increase, the bias is decreased by interpolating it with a uniform distribution according to Eq. (18). Finally, to apply the bias, the probability of each action is multiplied by its respective weight and the result renormalized according to Eq. (19).

$$\omega_j^2 = \delta^n b(j) + (1 - \delta^n) \frac{1}{8} \quad (18)$$

$$p_i^2(j) = \frac{\omega_j^2 p_i^1(j)}{\sum_k \omega_k^2 p_i^1(k)} \quad (19)$$

where

- ω_j^2 are the weights associated with each action.
- δ is constant which determines how fast the weights exponentially decay to a uniform distribution.
- n is the trial number.

- $p_i^2(j)$ is the probability after applying the bias.

3 Evaluation methods

3.1 Evaluation task

The evaluation task consists in having simulated rats learn how to find food in an open-field maze from multiple starting positions. Rats are unable to see the feeder, and food is provided once the rat is within 10 cm of the feeder. Each rat must execute a total of $N = m N_m$ trials where m is the number of starting positions and N_m is a parameter that defines the number of trials per starting position. Note that the maze, starting positions, and the number of trials per starting position vary according to the experiment.

The reason we use multiple starting positions is threefold. First, since our place cells do not take into account the presence of walls, it allows us to assess whether the model can learn the correct policy on both sides of the walls. Second, it also helps to reduce the learning time of locations that are far away from the feeder since rats starting closer to the goal are more likely to find the food, which in turn helps to propagate the value function sooner to other locations. Finally, experiments in rats (e.g., water maze) have shown that tasks that involve multiple starting positions usually require hippocampal information processing (Garthe and Kempermann 2003; Vorhees and Williams 2006).

During the task, a trial starts when the rat is placed in the maze and it ends either when the rat reaches the food or when it has performed 3000 steps, in which case the rat is removed without getting any reward. Note that for the mazes and starting positions used, 3000 steps allow rats to move at least 70 times the distance from the farthest starting location to the goal. Figure 10 illustrates three out of the eight mazes used in the experiments.

To have the rat learn the path from each starting position, the starting position is changed on each trial according to a random set of permutations $\{\sigma_i : 0 \leq i < N_m\}$ where each σ_i is chosen from a uniform distribution between all permutations of m elements. Given the set, the starting position of trial n is given by $\sigma_{n/m}(n \% m)$ where $/$ and $\%$ denote integer division and modulo operations. The starting positions are chosen in this way to reduce transfer effects of using a fixed order. Table 1 illustrates an example with $m = 3$ and $N_m = 2$.

3.2 Assessment

To assess the performance of each rat, for each trial we measure the optimality ratio of the path performed which we denote by $o(n)$. The optimality ratio is defined as the distance traveled by the rat divided by the distance of the shortest path

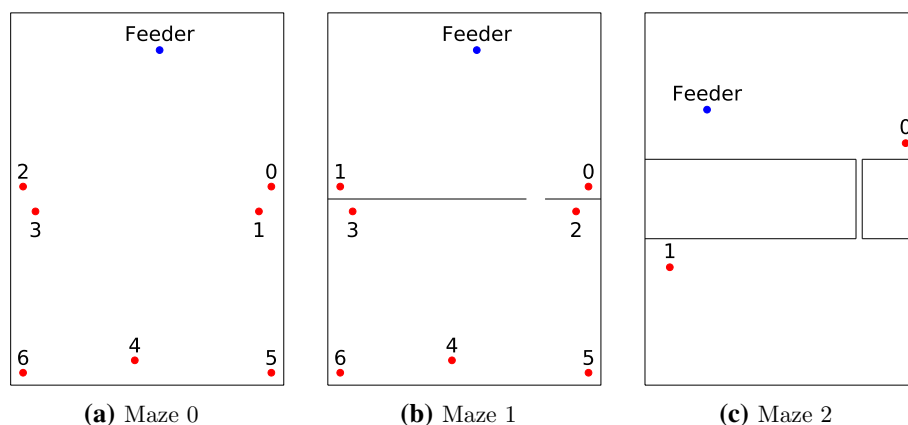


Fig. 10 Sample mazes. The figure shows three out of the eight mazes used in the experiments. All mazes have the same width and height (2.2 m and 3 m, respectively). Red dots indicate starting locations, while blue dots indicate feeder locations. Solid lines indicate walls that cannot be traversed. Note that starting locations are numbered according to

their distance to the feeder from closest to farthest. Mazes 3 through 7 are not shown since they only differ with maze 2 in the width of the gap, which is incremented in steps of 5 cm from 5 cm in maze 2 to 30 cm in maze 7 (color figure online)

Table 1 Starting positions example for $m = 3$ (the number of starting locations) and $N_m = 2$ (the number of episodes per starting locations)

Trial	0	1	2	3	4	5
Starting at	$\sigma_0(0)$	$\sigma_0(1)$	$\sigma_0(2)$	$\sigma_1(0)$	$\sigma_1(1)$	$\sigma_1(2)$

Note that first, we use a random permutation σ_0 to choose the starting locations. After traversing all starting locations, then a second random permutation σ_1 is used

$$o(l, n) = o\left(n + \sigma_n^{-1}(l)\right) \quad (20)$$

where

- $o(l, n)$ is the optimality ratio of the n th trial starting at position l .
- σ_n^{-1} is the inverse of σ_n .

from the respective starting location to the goal. The distance of the shortest path is calculated by first constructing a visibility graph (Lozano-Pérez and Wesley 1979) and then applying Dijkstra's algorithm (Dijkstra 1959). Note that $o(n) \geq 1$ for all n , and $o(n) = 1$ only when the path performed is optimal.

Since the performance of the rat in each episode depends on the starting position, the series $o(n)$ of size N is split into m subsequences of size N_m (one per starting position) according to Eq. (20).

Finally, instead of evaluating each location separately, we use a global metric to assess how the optimality ratio decreases over time in the maze overall. Thus, we compute the geometric mean of the subsequences according to Eq. (21).

$$\bar{o}(n) = \sqrt[m]{\prod_l o(l, n)} \quad \forall n : 0 \leq n < N_m \quad (21)$$

Table 2 Experiment configurations. The table shows the parameter ranges for each experiment. The last two rows are filled only for the experiments using two layers of place cells. Cells marked with “min”

Experiment	1	2	3	4	5
Experiment Name	Traces	Single min	Single same	Two scales	Gap
Traces	0–0.9	0.7	0.7	0	0.7
Mazes	1	0–1	0–1	0–1	2–7
Scale 1 r_λ (cm)	4–56	4–56	4–56	4–56	4–32
Scale 1 n_{λ_x}	Min	Min	40	Min	Min
Scale 2 r_λ (cm)				4–56	
Scale 2 n_{λ_x}				Min	

indicate that the number of cells used for a given layer was the minimum necessary to cover the entire maze with the respective scale. Bold values indicate the main parameter being varied in the experiment

3.3 Mazes

Through all experiments, a total of eight rectangular mazes were used (mazes 0 through 7), all sharing the same dimensions ($2.2\text{ m} \times 3\text{ m}$ which corresponds with the available space for future robot experimentation). Figure 10 shows the first three mazes.

Maze 0 and maze 1 are identical, except that an extra wall with a small gap was added in the middle of the maze separating starting positions 0 and 2 from 1 and 3. Since starting positions are sorted according to their distance to the goal, the names for locations 1 and 2 are swapped in mazes 0 and 1. Maze 0 was chosen since it is the simplest possible maze in the sense that there are no obstacles. Maze 1 was chosen to assess our hypothesis that small field size will be more helpful when precision is needed.

Mazes 2 through 7 are also almost identical. The only difference between them is the size of the gap which increases from 5 cm in maze 2 to 30 cm in maze 7 in steps of 5 cm. The main difference between these mazes and maze 1 is that, in maze 1, place cells close to the obstacle wall can activate on both sides of the wall. On the other hand, in mazes 2 through 7, this does not happen since the walls got thicker and the field sizes used in the respective experiments were chosen to prevent it from happening. Mazes 2 through 7 were chosen in this manner to assess how the size of the gap (precision needed) would affect the performance of each scale.

4 Experiments and results

The following subsections describe each of the experiments performed using the model along with the results obtained. Each experiment compares the performance of the model under different parameter configurations or maze setups. For each setup, 100 simulated rats were tested using different random seeds. Table 2 summarizes all experiment configurations.

The results of each experiment compare the median optimality ratio of different groups of rats as a function of the trial [as defined by Eq. (21)]. To do so, three main types of plots are used: time series showing the evolution of the median optimality ratio vs the trial for the different groups of rats; boxplots (Krzywinski and Altman 2014) showing the distribution of the optimality ratio in the last trial for each group, and heatmaps summarizing the results of Dunn tests (Dinno 2017) that compare the distributions from the boxplots. Dunn tests were computed using Python's library "scikit-posthocs," using Bonferroni adjustments of p values (Abdi 2007).

4.1 Experiment 1

The first experiment assesses how to choose the eligibility trace decay parameter for different field sizes. For this reason, in this experiment we use a single layer of place cells λ_0 , varying the field size r_{λ_0} from 0.04 m (half a rat step) to 0.56 m (approximately one-fourth of the maze's width) in steps of 0.04 m, and varying the trace decay parameter α_{λ_0} from 0 to 0.9 in steps of 0.1. For each field size, n_{λ_x} is chosen so that the maze is covered using the minimum number of cells, which is obtained by imposing the condition $d_{\lambda}\sqrt{2} < 2r_{\lambda}$ and then using Eq. (2) to solve for n_{λ_x} . Due to the high computational costs of the experiment, the results were computed only on maze 1.

4.1.1 Results

The results can be divided into two categories: small scales (between 0.04 and 0.32 m) and large scales (0.36–0.56 m).

When assessing convergence time, the median optimality ratio [as defined in Eq. (21)] converged faster as the trace decay rate was increased (which is expected since that is the purpose of eligibility traces). The effect was most noticeable for the smallest scale, and it faded out as the scale got larger. After reaching scale 0.36, no further speedup was observed. Instead, after reaching the minimum around trial 10, the median optimality ratio started diverging faster as the scale got larger. The results are shown in the left column of plots of Fig. 11. The plots show the median optimality ratio as a function of the trial for each trace value. The first two rows correspond to small scales 0.04 and 0.32, while the following two correspond to large scales 0.52 and 0.56.

To assess whether different traces affected the optimality ratio to which the model converged to, for each scale, we compared the distributions of the results obtained in the last trial. These distributions along with their respective boxplots are shown in the second column of Fig. 11. To assess whether the differences were statistically significant, a Dunn test was performed comparing the distributions of each pair of traces. The results are summarized in the heat maps shown in the third column of Fig. 11. Green cells indicate that the distributions being compared are statistically different with p values smaller than 0.03 for dark green cells, 0.05 for mid-green cells, and 0.07 for light green cells. Red cells indicate no statistical difference was observed (i.e., p values greater or equal to 0.07).

For small scales, the distribution of traces between 0 and 0.8 was not statistically different for all scales except 0.08 and 0.24. For scales 0.24 and 0.08, exceptions were observed. For scale 0.24, trace 0.4 was statistically better than traces 0, 0.1,

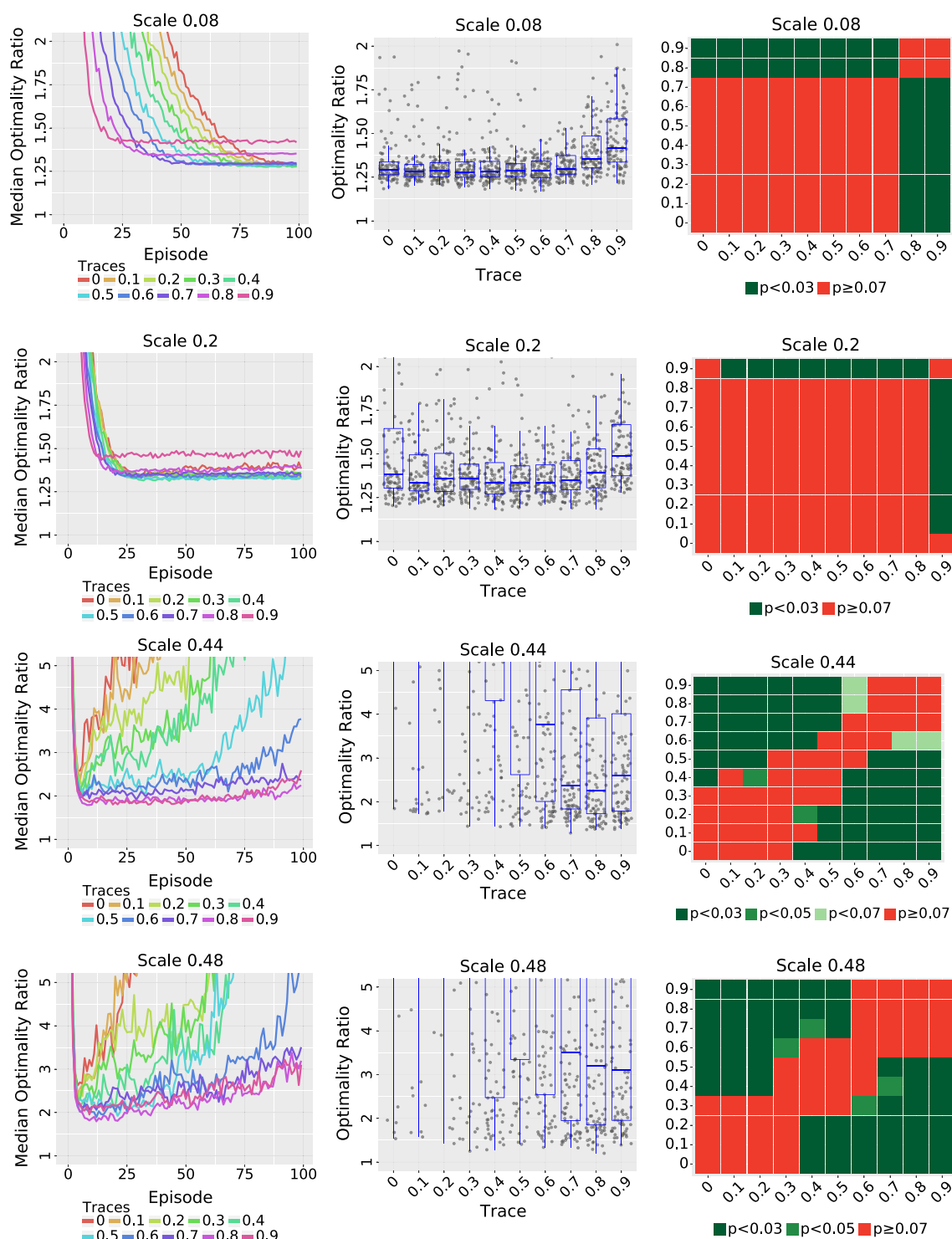


Fig. 11 Experiment 1 results. The left column shows the median optimality ratio versus the trial for each trace value. Note that high optimality ratios are omitted from the plot for clarity purposes. The middle column shows the distribution of the results during the last trial. The right column shows the results of a Dunn test comparing the distribution of

each pair of traces. Dark green cells indicate p values lower than 0.03, mid-green cells indicate p values lower than 0.05, light green cells indicate p values lower than 0.07, and red cells indicate p values of 0.07 or higher. From top to bottom, each row corresponds to scales 0.08, 0.20, 0.52, and 0.56, respectively (color figure online)

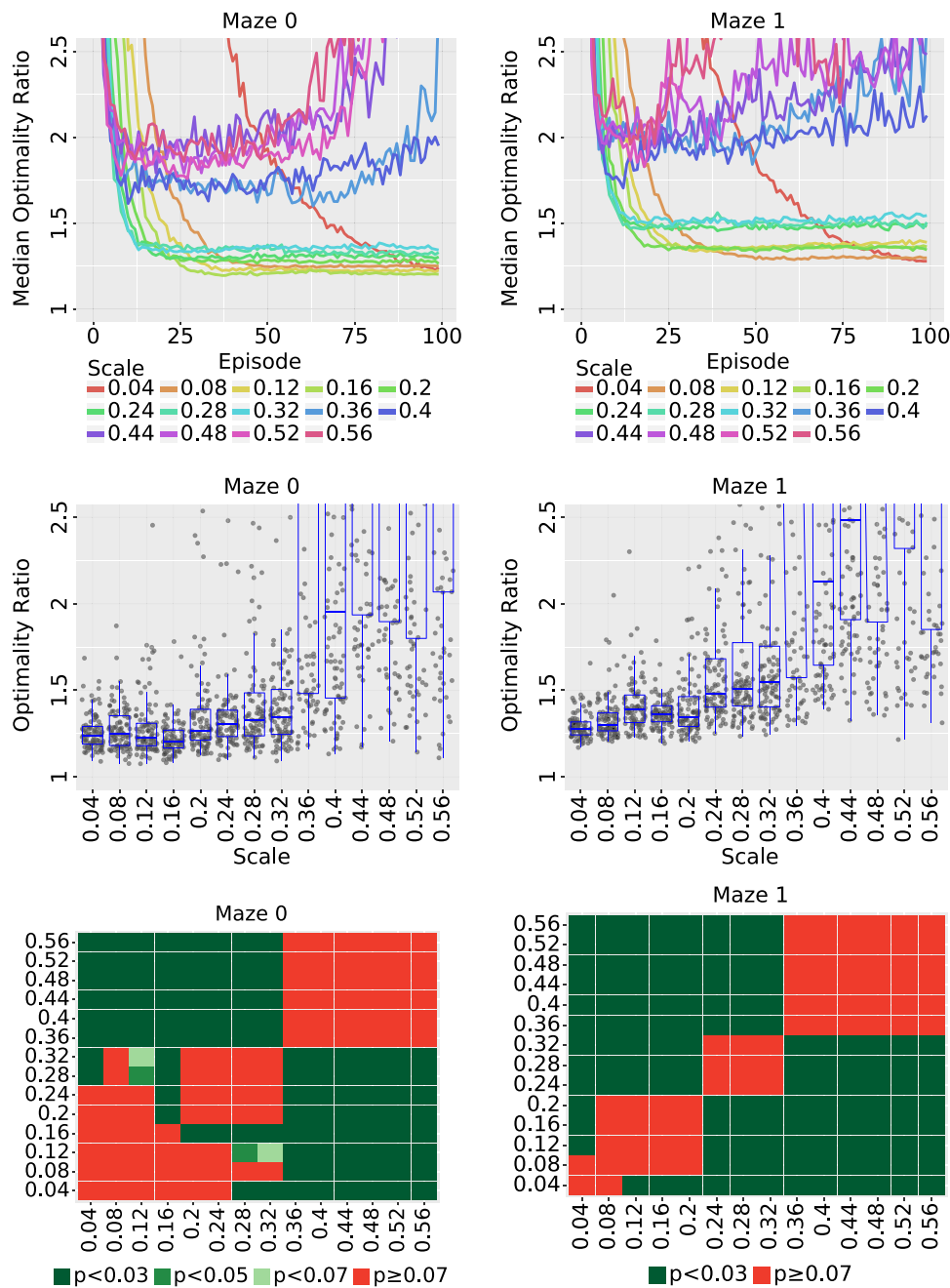


Fig. 12 Experiment 2 results: The results of using a single layer of place cells with the minimum number of cells required to cover the maze (thus larger scales have less cells). The left column corresponds to the results on maze 0, while the right column corresponds to the results on maze 1. The upper row shows the median optimality ratio as a function of the trial for each scale. Note that high optimality ratios are omitted from the plot for clarity purposes. The middle row shows the distribution of

the results during the final trial. The bottom row shows the results of the Dunn tests comparing the distributions shown in middle row. Dark green cells indicate p values lower than 0.03, mid-green cells indicate p values lower than 0.05, light green cells indicate p values lower than 0.07, and red cells indicate p values of 0.07 or higher (color figure online)

and 0.2 with p values of 0.03, 0.05, and 0.03, respectively, and trace 0.8 was statistically worse than trace 0.2 with p value of 0.07. For scale 0.08, trace 0.8 was statistically worse than traces between 0 and 0.7. For trace 0.9, the median was worse

than all other traces for all scales, but statistical significance varied on a case-by-case scenario. In summary, the choice of the trace value did not significantly affect the optimality of the solution except for traces 0.8 and 0.9 which yielded

the degraded results. Note that this analysis is only valid for scales larger than 0.04, since scale 0.04 had not yet reached convergence by the last trial for most trace values.

On the other hand, for large scales, when assessing the median optimality ratio of the final trial, large traces (values between 0.7 and 0.9) help to either prevent the median from diverging or to reduce the speed of divergence as observed in the last two rows of Fig. 11.

4.2 Experiment 2

The second experiment assesses how the field size affects the convergence time and optimality of the solution.

As in experiment 1, only one layer of cells was used with field sizes varying from 0.04 to 0.56 in steps of 0.04. Once again, the number of cells for each scale was chosen so that the maze was covered using the minimum number of cells. Based on the results of experiment 1, the trace decay parameter was set to 0.7 for all scales.

This experiment was performed both in mazes 0 and 1. Maze 0 assesses the model when no obstacles are present, providing a scenario with minimal complexity. On the other hand, maze 1 introduces a wall which divides the maze in two halves connecting them by a small gap which results in two sources of complexity for the model. First, since the gap is small, the maze requires precise movements for crossing from one side to the other. Thus, larger place fields may result in worse optimality ratios due to being too coarse. Second, since our model of place cells does not take walls into account, a single place cell can activate on both sides of the obstacle wall, thus incorrectly generalizing the value functions from one side of the wall to the other.

4.2.1 Results

Figure 12 shows the results of experiment 2 (which uses a single layer of places with just enough cells to cover the maze). As observed in the first row of the figure, the model converges faster for larger PC scales than smaller PC scales. Also, for scales between 0.04 and 0.32m the results seem to be stable once the minimum is reached, while for scales between 0.36 and 0.56m, the median diverges after reaching the minimum.

When comparing the medians of each PC scale in the final trial (second and third rows of Fig. 12), we found that small scales (0.04–0.32) were better than large scales (0.36–0.56) for both mazes with p values smaller than 0.03. When comparing the results between the different scales of the small group, the results differed for mazes 0 and 1. For maze 0, statistically significant differences were found when comparing scale 0.04 and 0.12 with scales 0.28 and 0.32 and when comparing 0.16 with scales 0.2 to 0.32, and in all cases

the smaller scale achieved better results. On the other hand, for maze 1, the differences between scales are more noticeable. Scale 0.04 was statistically better than scales ranging from 0.12 to 0.32, and scales 0.08 to 0.2 were statistically better than scales 0.24 to 0.32. On both mazes, all significant differences had p values smaller than 0.03, except on maze 0 when comparing scale 0.12 to scales 0.28 ($p < 0.5$) and 0.32 ($p < 0.7$).

Summarizing the results, in general, we observed that smaller scales converge to better (median) optimality ratios than larger scales. The difference between scales is more noticeable on maze 1 (the more complex maze) than maze 0. Cases where the median of smaller scales converged to worse results than larger scales were not statistically significant ($p > 0.07$).

4.3 Experiment 3

Experiment 3 repeats experiment 2 but changing the number of cells used for each scale. Whereas in experiment 2 the number of cells varied for different scales, in experiment 3 the value is constant with all scales having $n_{\lambda x} = 40$. Note that this value was the one used for scale 0.04 in experiment 2.

The objective of this experiment is to assess how the results differ from experiment 2 when comparing the different scales using the same number of cells. For example, if a large scale was outperformed by a smaller scale in experiment 2, would the result change if both scales had the same number of cells?

4.3.1 Results

Figure 13 shows the results of experiment 3. In contrast to the results from experiment 2, when using the same number of cells for all scales, the larger scales take longer to converge. (Note that the number of trials in this experiment is 5 times the number of the previous experiment.)

When comparing the distributions of the results during the final trial, the observations that smaller scales converge to better results than larger scales still apply, except that now only scales between 0.04 and 0.24 converge. Also, note that although in both mazes scale 0 is worse than scales 0.04 and 0.08, the result of the Dunn test indicate that the difference is not significant ($p > 0.07$).

4.4 Experiment 4

Experiment 4 assesses whether eligibility traces could be replaced by multiscale representations which could be a biologically plausible mechanism for speeding up learning. Experiment 4 repeats experiment 2 but setting traces to 0, and using two layers of place cells instead of one, assessing all possible combinations of the original set of layers. Since originally there were 14 scales, in this experiment we

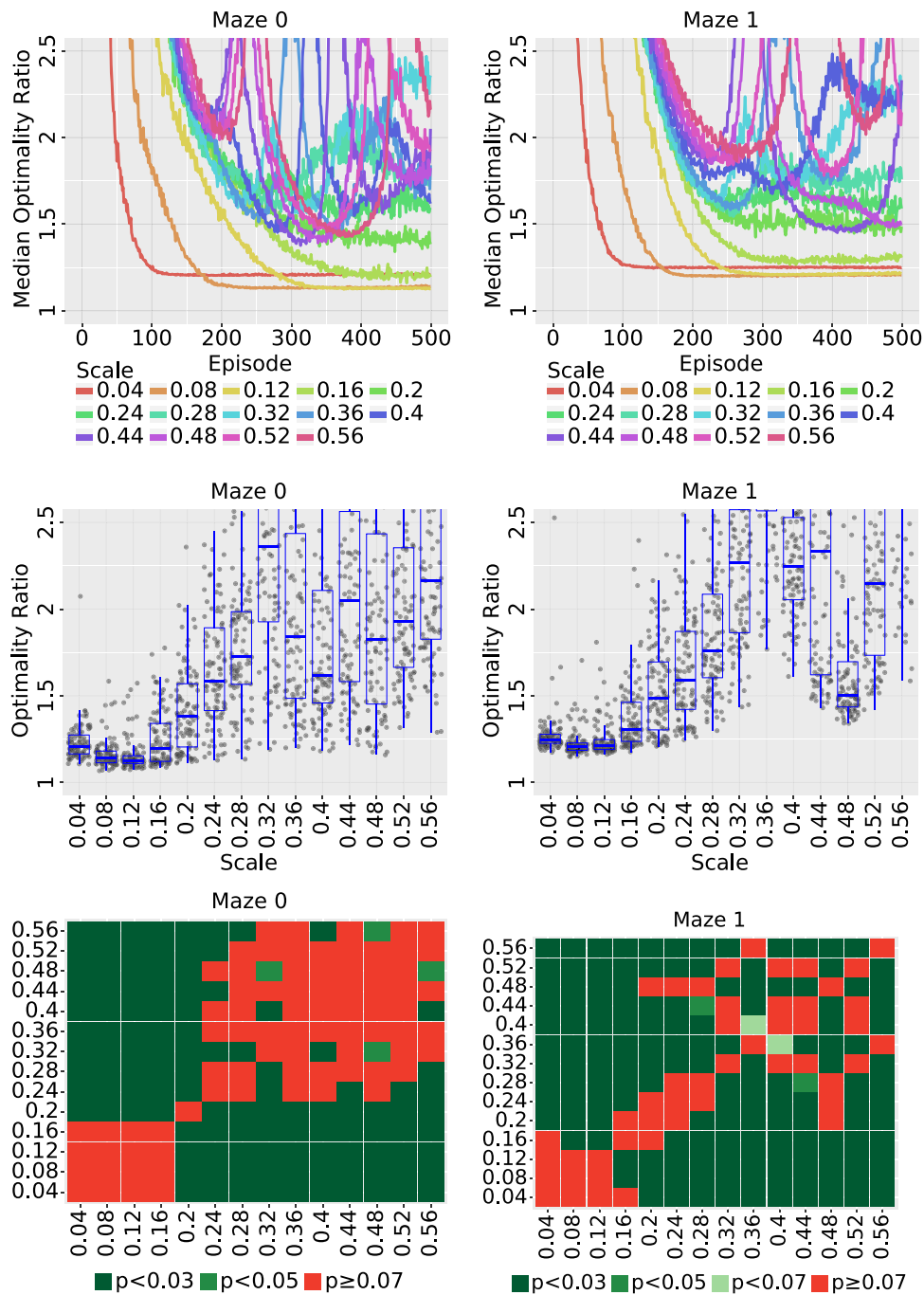


Fig. 13 Experiment 3 results: The results of using a single layer of place cells with a fixed number of cells for all scales. The left column corresponds to the results on maze 0, while the right column corresponds to the results on maze 1. The upper row shows the median optimality ratio as a function of the trial for each scale. Note that high optimality ratios are omitted from the plot for clarity purposes. The middle row shows the

distribution of the results during the final trial. The bottom row shows the results of the Dunn test comparing the distributions shown in the middle row. Dark green cells indicate p values lower than 0.03, mid-green cells indicate p values lower than 0.05, light green cells indicate p values lower than 0.07, and red cells indicate p values of 0.07 or higher (color figure online)

assess $\frac{14 \cdot 13}{2} = 91$ possible combinations. Also, note that to allow for fair comparisons before and after mixing the scales, the single-scale models from experiment 2 were also rerun without the use of traces.

4.4.1 Results

Figure 14 summarizes the results of experiment 4. The results analyze what happens to the speed and value of convergence

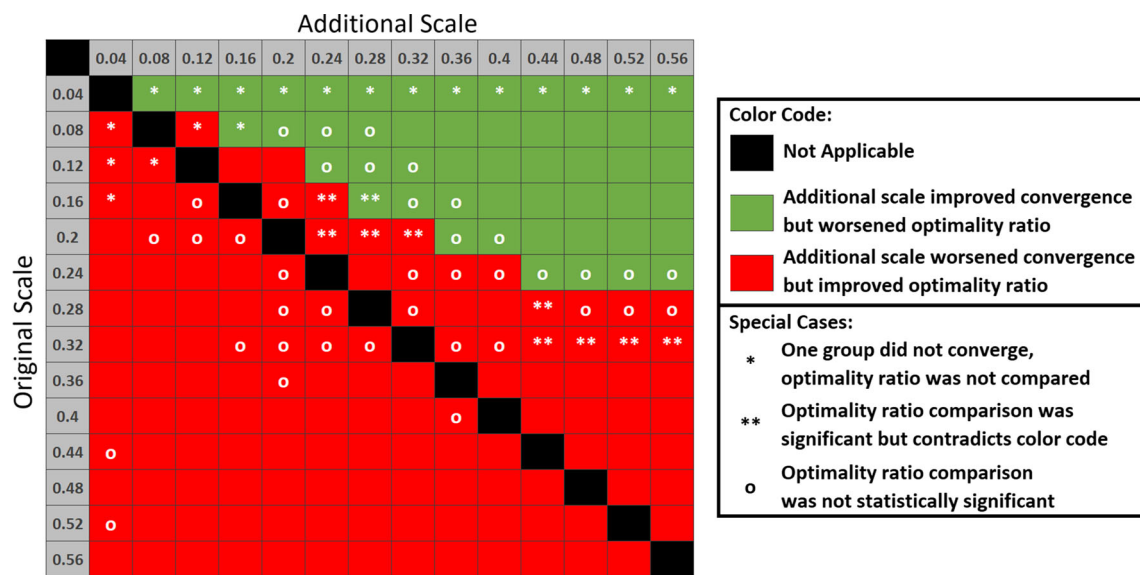


Fig. 14 Experiment 4 results: The results of experiment 4 for mazes 0 and 1. Each cell in the map compares the results of the model using the single layer of place cells indicated by the row, with the results after adding the second layer indicated by the column. The results are indicated by the colors of the cells. Black cells indicate “not applicable”, that is, comparisons that were skipped since using two equal layers is equivalent to using a single layer. Red cells indicate that adding the second scale slowed down the convergence speed, but significantly improved the median optimality ratio of the final trial ($p < 0.05$) both in mazes 0 and 1. Similarly, green cells indicate that adding the second layer increased the convergence speed but significantly worsened the optimality ratio both in mazes 0 and 1. Single asterisks “*” indicate that the

median optimality ratios were not compared due to at least one group not having converged by the final trial for at least one of the mazes. Double asterisks “**” indicate cells for which the median optimality ratio comparison resulted in the significant results contradicting the color of the cell for at least one maze. “o”s indicate cells for which the comparison of the optimality ratio was not significant for at least one maze. In general, adding a second layer of smaller place cells (cells below the black line) resulted in slower convergence to better optimality ratios. When adding a second layer of larger place cells (cells above the black line), convergence speed increased to worse optimality ratios only if the second scale was large enough (color figure online)

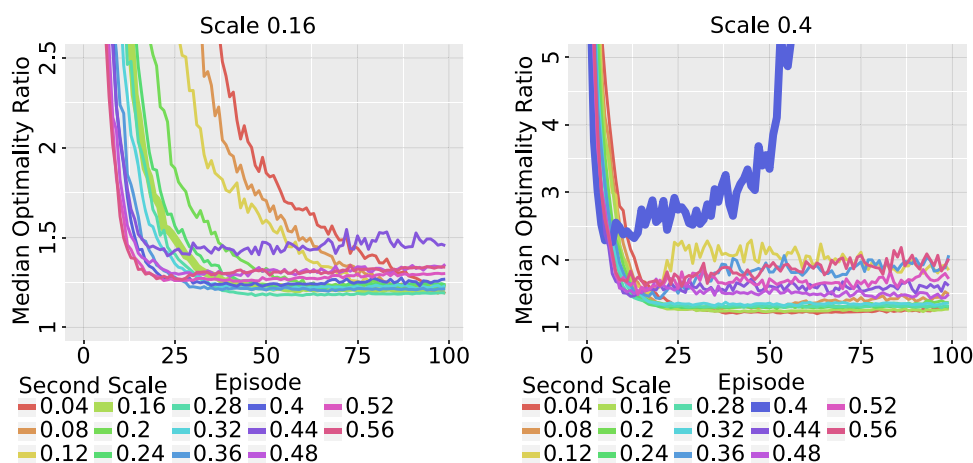


Fig. 15 Sample combinations. The median optimality ratio is shown as a function of the trial for different layer combinations using the minimum number of cells to cover the maze for each layer. Note that high optimality ratios are omitted from the plot for clarity purposes. The

graph on the left shows all combinations using scale 0.16, while the graph on the right shows all combinations using scale 0.4. The thicker lines on each plot show the results of using a single layer of place cells (layer 0.16 for the left plot and 0.4 for the right plot)

when adding a second layer of cells. The triangle of cells below the black line correspond to adding a layer of smaller cells, while the cells above correspond to adding a layer of larger cells.

As observed in the image, adding a layer of cells smaller than the original scale always resulted in slowing down convergence but also in improving the median optimality ratio of the final trial when a statistically significant difference was

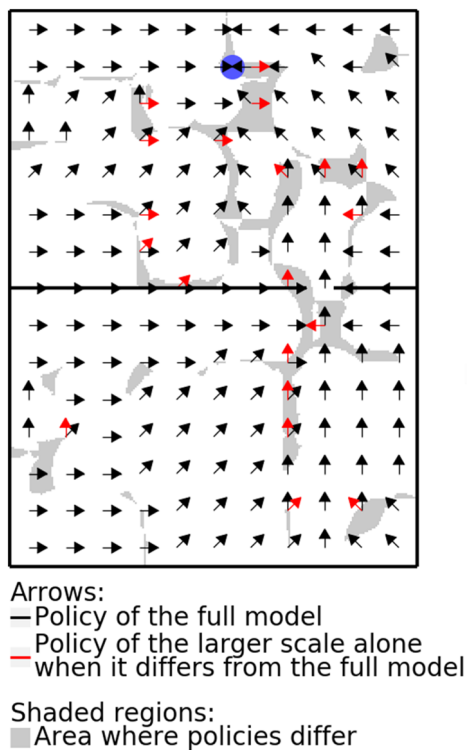


Fig. 16 Policy comparison. The figure illustrates a comparison between two policies after the final trial using the two-layer model with scales 0.16 and 0.28. The blue dot at the top of the image indicates the position of the feeder. Solid lines indicate the walls of the maze. Black arrows represent the actions corresponding to the highest action values at the sampled locations using the full model. Shaded regions indicate areas where the policy generated by the full model differs from the policy generated by its larger scale alone. Red arrows indicate the actions chosen by the latter policy. For the most part, both policies coincide. The areas where the policies differ (i.e., areas where scale 0.16 made a difference) seem to correspond with the areas where the policy is changing (color figure online)

observed ($p < 0.05$). Note that in some cases (marked by single asterisks in Fig. 14), the comparison between optimality ratios was not performed due to at least one still converging by the final trial thus preventing a fair comparison.

On the other hand, when adding a layer of cells larger than the original scale, convergence speed was not always increased. As observed in Fig. 14, for scales 0.24 and smaller, the gain in speed was only observed if the new cells were sufficiently large. In all other cases, convergence was slowed down. Based on experiments 2 and 3, we attribute this observation to the conflicting properties between the scale size and the number of cells. As observed in experiment 2, when using the minimum number of cells, large scales allow for faster convergence than small scales. On the other hand, when increasing the number of cells as in experiment 3, convergence is slowed down. Thus, using larger scales allows for

increased convergence speeds, but the incorporation of additional cells reduces it.

In the cases where the convergence speed was increased and median optimality ratio comparisons were made, only the combination (0.16, 0.28) was statistically significantly improved. In all other cases, the difference was either not significant or the ratio was worse than the original value.

As an example, in Fig. 15 we show the results for maze 0 of combining layers 0.16 and 0.40 with all other layers. Notably, the larger the second scale used, the faster the convergence, although when combining scale 0.16 with 0.20 (which is larger), the result converges slower than scale 0.16 alone. Finally, note that originally scale 0.40 did not converge, but by combining it with another scale it either converges or diverges more slowly.

To further assess how the multiscale model combines the information from multiple layers, we proceeded to plot the policy generated by a sample configuration of the two-layer model and we compared it to the policy generated by the layer of larger place cells alone. Figure 16 illustrates the comparison using layers 0.16 and 0.28. As observed in the image, both policies coincide almost everywhere differing only in the shaded regions, which seem to coincide with the areas where the policy is changing. Thus, it seems that the contribution made by the smaller scale is only used in areas where the policy generated by the larger scale is changing. Although in our experiment, those areas represent just a small portion of the maze, that is mostly due to using only eight actions and few obstacles, which results in policies that are constant through big portions of space. Out of the regions where the smaller scale made a difference, we distinguish two noteworthy areas. One is the area around the gap in the maze, where the smaller scale helped the policy point in the direction of the gap. The other is the area around the feeder, where the smaller scale contributed in making the policy point toward the feeder rather than away. Both cases represent areas where detail is required (where the policy changes rapidly) and the smaller scale was useful in capturing those details.

4.5 Experiment 5

In experiment 5, we assess how the required precision of movement in a maze affects the optimality of the solution for different scales. This experiment repeats experiment 2 but using the scales between 0.04 and 0.32 m only and performing the task on mazes 2 through 7 instead of 0 and 1.

Mazes 2–7 are similar to maze 1, but the obstacle wall is made thicker to prevent place cells from activating on both sides of the wall. Also, the size of the gap increases from maze 2 to maze 7, and thus, the precision required to move from one half of the maze to the other decreases between mazes.

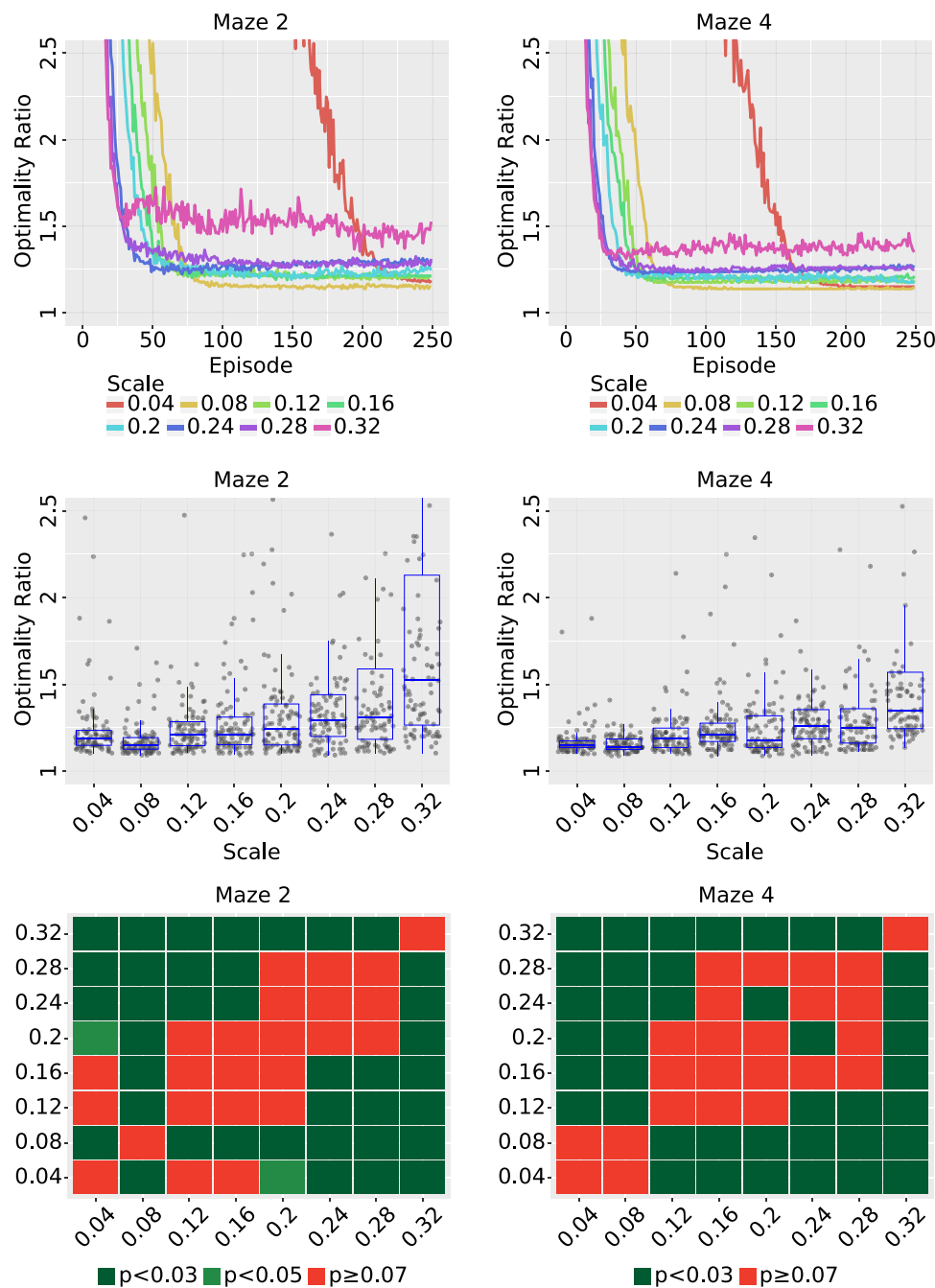


Fig. 17 Experiment 5 results (fixed maze): The results of using a single layer of place cells with the minimum number of cells required to cover the maze for scales 0.04–0.32. The left column corresponds to the results on maze 2, while the right column corresponds to the results on maze 4. The upper row shows the median optimality ratio as a function of the trial for each scale. Note that high optimality ratios are omitted from the plot for clarity purposes. The middle row shows the distribution of the results during the final trial. The bottom row shows the results of the Dunn test comparing the distributions shown in the middle row. Dark green cells indicate p values lower than 0.03, mid-green cells indicate p values lower than 0.05, light green cells indicate p values lower than 0.07, and red cells indicate p values of 0.07 or higher (color figure online)

4.5.1 Results

In experiment 5, we analyze two types of results. First, when fixing the maze and changing the scales (Fig. 17), the results were the same as in experiment 2 for all mazes (2 through 7).

That is, smaller scales reach better optimality ratios but take longer to converge. The only exception to the rule was scale 0.04, but this result is ignored since the median optimality ratio was still decreasing by the last trial, meaning it still had not reached convergence.

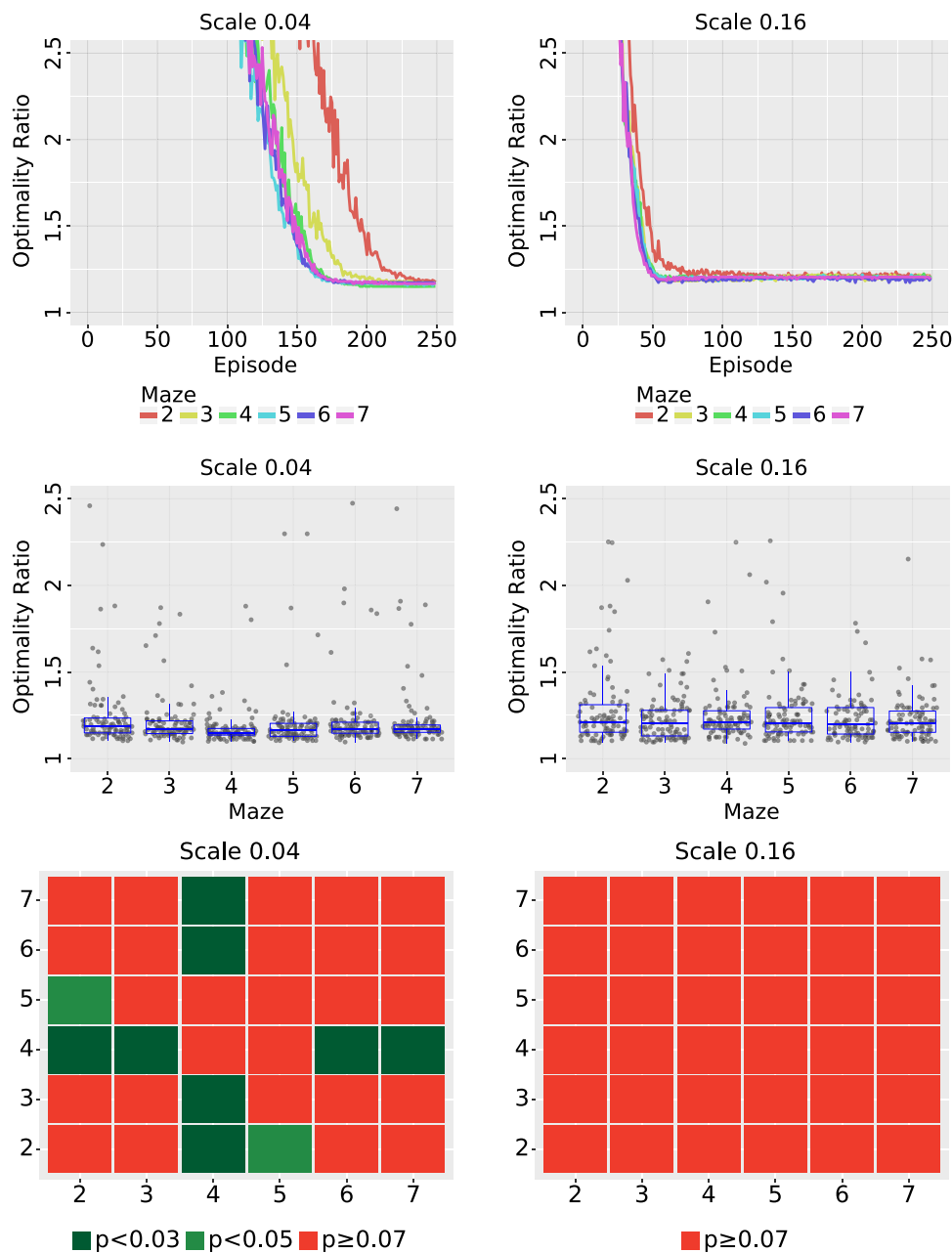


Fig. 18 Experiment 5 results (fixed scale): The results of using a single layer of place cells with the minimum number of cells required to cover the maze on mazes 2–7. The left column corresponds to the results of scale 0.04, while the right column corresponds to scale 0.16. The upper row shows the median optimality ratio as a function of the trial for each scale. Note that high optimality ratios are omitted from the plot for

clarity purposes. The middle row shows the distribution of the results during the final trial. The bottom row shows the results of the Dunn test comparing the distributions shown in the middle row. Dark green cells indicate p values lower than 0.03, mid-green cells indicate p values lower than 0.05, light green cells indicate p values lower than 0.07, and red cells indicate p values of 0.07 or higher (color figure online)

When fixing the scale and changing the maze, the only consistent result observed was that the two mazes with smallest gaps (mazes 2 and 3) took longer to converge than the other mazes. The results are shown in Fig. 18. Although we were expecting to find that mazes with smaller gaps would elicit convergence to worse ratios, this was not observed in

the experiment. Instead, in most scenarios, there were no statistical differences between the results of the last trial for different mazes, as can be seen in the last row, right column of Fig. 18. In the cases where we did observe statistical differences, the results were not consistent, sometimes getting better results for the mazes with narrower gaps (as when

comparing mazes 4 and 6 for scale 0.04), and sometimes for mazes with wider gaps (as when comparing mazes 2 and 4 for scale 0.04).

5 Conclusions and discussions

In this paper, we introduced a computational model for spatial cognition analyzing the effects of combining place fields of multiple scales during a goal-oriented learning task using reinforcement learning. The model uses HRBFs to represent place cells of multiple field sizes and to approximate the state and action value functions. The model was based on the field size gradient observed in place cells along the dorsoventral axis of the hippocampus. The model was used to assess possible benefits of using a multiscale architecture in a reinforcement learning algorithm.

The main results of this research are that multiscale representations can be used to decrease learning times and improve the optimality of the solutions based on the choice of the activation field sizes. Furthermore, a trade-off was generally observed between the learning time and the optimality of the solutions.

Additionally, simulations were performed to assess how the complexity of a maze can affect place field sizes. The results showed that differences in performance (median optimality ratio during the last episode) are more significant under the presence of obstacles, with smaller place fields generally outperforming larger place fields. This suggests that in order to understand the differences between scales, experiments should be performed in environments more complex than obstacle-free open mazes. This result goes in hand with the results from Contreras et al. (2018) where bilateral inactivation of either dorsal or ventral hippocampus was reported to impair spatial navigation only under the presence of obstacles. Based on the results from experiment 2, this could be due to any scale being sufficient to solve an obstacle-free open maze.

When analyzing the results of the experiments performed, the following specific observations were made:

1. Smaller scales consistently converged to better solutions than larger scales in all single-scale experiments. We believe this is due to larger scales generating coarser divisions of space in which the policy remains constant. For large enough scales (0.36 and higher), the median optimality ratio was observed to diverge due to more and more rats getting stuck in cycles, presumably due to the coarseness of the scale.
2. Assessing convergence times for each scale separately (experiments 2 and 3), we found the two contrasting results. When each scale had the minimum number of cells required to cover the maze, the median optimality

ratio decreased faster for larger scales than smaller ones. On the other hand, the result was reversed when all scales had the same number of cells, with the optimality ratio decreasing faster for smaller scales than larger ones. The latter result coincides with the results from Kretchmar and Anderson (1997) that performed the same experiment on a different problem from a pure reinforcement learning context. This phenomenon could be explained by the following. When approximating a function (in this case, the reinforcement learning value function), if the basis functions (place cells) do not overlap, a change in the value of a single basis function does not affect the approximation error of other basis functions. Thus, the update is constrained to that single basis. On the other hand, if the basis functions overlap with each other, an update to one basis affects the error obtained using the surrounding basis functions, generating waves of updates in order to learn the target function. This not only can account for slower convergence speeds, but could also explain the instabilities observed in some ratios of experiment 3. Although the results from experiments 2 and 3 would seem to suggest that using the fewest number of cells to cover the maze produces the best results for many scales, it must be noted that no noise was introduced in the simulations. Since noise is the main reason for adding redundancy (overlap between place cells), this result is most likely to differ if noise was introduced.

3. In experiment 1, the best results were reached using traces around 0.7 for all scales. For small scales (using the minimum number of cells), the use of traces showed significant improvement in convergence speed while converging to the same optimality ratio (for traces smaller than 0.7–0.8). On the other hand, traces did not improve convergence speed for the larger scales, nonetheless the median optimality ratio of the final trial was improved.
4. Multiscale computation can be used to accelerate learning rates or to improve the quality of the solutions. When mixing any two layers (using minimum number of cells on each layer), adding a smaller scale always resulted in slower convergence rates. On the other hand, adding a larger scale increased convergence rates only if the scale was large enough. In most cases, a tendency was observed where increasing convergence speed decreased the quality of the solutions and vice versa.
5. Experiment 4 suggested that when using two layers, the smaller layer is mainly useful in areas where the policy from the larger layer is changing. In particular, this includes areas where the optimal policy changes fast over space such as in areas close to the goal or around obstacles.
6. When assessing how each scale performs under different maze complexities, we did not find any conclusive results. In general, it was observed that smaller scales

converged to better performances than larger scales with differences becoming significant under the presence of obstacles. Nevertheless, when gradually decreasing complexity by changing the size of the gap between mazes 2–7, the only difference observed was a decrease in the time required to solve the task, but no significant changes in performance were observed.

While constructing the model based on recent evidence (Contreras et al. 2018; Harland et al. 2017; de Hoz et al. 2003), our main hypothesis was that the entire longitudinal axis of the hippocampus is involved in spatial navigation. More specifically, we hypothesize that dorsal place cells play a major role when details are required, while ventral place cells allow to reduce both learning time and the number of neurons required in larger, more complex environments. Although the experiments with the model would seem to support our hypothesis, the results are limited requiring further assessment, and we do not yet have rat experiments supporting the model's findings. Nevertheless, based on our experience with the model, we believe that the continuum of place field sizes observed in the hippocampus allows rats to learn policies with varying levels of detail that can adapt to many maze sizes and complexities. In particular, we believe that larger scales allow to quickly learn rough (coarse/gist) policies that can be improved upon by smaller scales when the environment requires more precision. This is consistent with the results observed in Fig. 16 where the contributions of the smaller scale were most noticeable around the gap in the maze and in the areas surrounding the feeder, considering that larger scales converged faster, but smaller scales converged to better (here: shorter) solutions. Also, we believe that by using place fields of appropriate sizes, the number of neurons involved in a task can be minimized, possibly increasing the amount of memory and therefore the ability to solve more complex environments. Given these observations, we predict the following: (1) deactivating large scales in large environments will increase learning times (i.e., the number of episodes to solve a task); (2) deactivating small scales, especially in complex environments, will reduce performance (increasing the number of steps required to solve a task) but not learning times; and (3) we predict that smaller scales will be more active at decision points and smaller environments, while larger scales will be more active in open areas and, in general, in larger environments.

Based on the results of the experiments, to better assess the differences between scales and the benefits of a multi-scale architecture, experiments should be performed in more complex environments. Thus, in the future, we plan to further analyze the model in environments with varying dimensions and more obstacles. Furthermore, we plan to assess combinations of more scales, possibly with a continuum spectrum of

sizes and compare our results to that of future rat experiments using similar mazes to the ones we proposed.

Although in this initial study, the model was useful in demonstrating the possible benefits of multiscale and model dynamics, the model should be extended to address aspects such as mapping place cells to the environment and increased environment complexity. Even though the model was able to solve the tasks presented, before including more scales or experimenting in more complex environments, the model would benefit by improving some of the following limitations:

1. First and foremost, our current model of place cell does not take walls into account. In particular, this means that the state and action value functions can be incorrectly generalized from one side of the wall to the other. Unlike our model, real place cells interact with walls in several ways such as “deactivating” when a wall is introduced intersecting its place field (Muller and Kubie 1987), or “activating” only under the presence of a wall at a given location (Rivard et al. 2004; Lever et al. 2009). Modeling these interactions such as in Llofriu et al. (2019) prevents the policy from incorrectly generalizing between different sides of the wall and may allow learning policies that are conditional on the presence of walls. Introducing these types of cells would also require further assessment of how navigation would be affected by them.
2. The state and action value functions were observed to diverge due to the algorithm suffering from the deadly triad, as explained in Sutton and Barto (2018). In particular, this results in numerical instabilities in computations and the possibility of the rat getting stuck in non-optimal policies. Fixing this issue may allow to better exploit the bigger scales used in the experiments which were observed to diverge.
3. For each layer, the current model distributes place cells uniformly over space which prevents recruiting dense populations of place cells were required while using sparse populations elsewhere. We believe that allowing arbitrary distributions will help to reduce the number of cells required to solve a task while keeping high performances.
4. The equation used for the update of the action value function was defined in terms of the error of the state value function. If the state value function converged, then the error would become 0, and thus, the action value function would also converge irrespective of whether it reached the target value. Future work should improve the update equation to allow the policy to learn regardless of the state value function having already converged.
5. The current action selection process is rather basic and can result in paths which would never be performed by real rats, also the added biases can influence the pol-

icy learned by the reinforcement learning algorithm. The model would greatly benefit from including a working memory model in the action selection process, as well as a curiosity model.

6 Complex spatial navigation

The computation model described in this paper addresses the problem of spatial navigation in complex environments by analyzing the potential benefits of the dorsoventral multiscale organization of place cell fields in the hippocampus, where smaller scales of representation seem to be useful in improving path optimality, whereas larger scales seem to be useful in reducing learning time. The results suggest that combining scales can enhance the performance of the multiscale model, with a trade-off between path optimality and learning time. Furthermore, the involvement of larger scales may reduce the number of total cells required to represent a particular environment, where the total number and size of the cells would depend on the environment complexity. Additionally, the results show that differences in performance are more significant under the presence of obstacles, with smaller place fields generally outperforming larger place fields, i.e., dorsal place cells play a major role when details are required, while ventral place cells allow to reduce both learning time and the number of neurons required in larger and more complex environments.

Acknowledgements This work is being funded by NSF IIS Robust Intelligence Research Collaboration Grant #1703340 at the University of South Florida and the University of Arizona entitled “Experimental and Robotics Investigations of Multi-Scale Spatial Memory Consolidation in Complex Environments”.

References

- Abdi H (2007) Bonferroni and Šidák corrections for multiple comparisons. *Encycl Meas Stat* 3:103–107
- Bellocchio F, Ferrari S, Piuri V, Borghese NA (2007) Online training of Hierarchical RBF. In: IEEE International conference on neural networks—conference proceedings, pp 2159–2164. <https://doi.org/10.1109/IJCNN.2007.4371292>. ISSN 10987576
- Chen LL, Lin LH, Green EJ, Barnes CA, McNaughton BL (1994) Head-direction cells in the rat posterior cortex—I. Anatomical distribution and behavioral modulation. *Exp Brain Res*. <https://doi.org/10.1007/BF00243212>. ISSN 00144819
- Contreras M, Pelc T, Llofriu M, Weitzenfeld A, Fellous JM (2018) The ventral hippocampus is involved in multi-goal obstacle-rich spatial navigation. *Hippocampus*. <https://doi.org/10.1002/hipo.22993>. ISSN 10981063
- de Almeida L, Idiart M, Lisman JE (2009) The input–output transformation of the hippocampal granule cells: from grid cells to place fields. *J Neurosci*. <https://doi.org/10.1523/jneurosci.6048-08.2009>. ISSN 0270-6474
- de Hoz L, Knox J, Morris RG (2003) Longitudinal axis of the hippocampus: both septal and temporal poles of the hippocampus support water maze spatial learning depending on the training protocol. ISSN 10509631
- Dijkstra EW (1959) A note on two problems in connexion with graphs. *Numer Math* 1(1):269–271
- Dinno A (2017) Package ‘dunn.test’: Dunn’s test of multiple comparisons using rank sums. R package version 1(4)
- Erdem UM, Hasselmo M (2012) A goal-directed spatial navigation model using forward trajectory planning based on grid cells. *Eur J Neurosci*. <https://doi.org/10.1111/j.1460-9568.2012.08015.x>. ISSN 0953816X
- Erdem UM, Hasselmo ME (2014) A biologically inspired hierarchical goal directed navigation model. *J Physiol Paris* 108(1):28–37. <https://doi.org/10.1016/j.jphysparis.2013.07.002>. ISSN 09284257
- Fan C, Chen Z, Jacobson A, Hu X, Milford M (2017) Biologically-inspired visual place recognition with adaptive multiple scales. *Robot Auton Syst* 96:224–237. <https://doi.org/10.1016/j.robot.2017.07.015> ISSN 09218890
- Fanselow MS, Dong H-W (2010) Are the dorsal and ventral hippocampus functionally distinct structures? *Neuron*. <https://doi.org/10.1016/j.neuron.2009.11.031>. ISSN 1097-4199
- Ferrari S, Frosio I, Piuri V, Borghese NA (2005) Automatic multiscale meshing through HRBF networks. *IEEE Trans Instrum Meas*. <https://doi.org/10.1109/TIM.2005.851471>. ISSN 00189456
- Fyhn M, Molden S, Witter MP, Moser EI, Moser MB (2004) Spatial representation in the entorhinal cortex. *Science*. <https://doi.org/10.1126/science.1099901>. ISSN 00368075
- Garthe A, Kempermann G (2013) An old test for new neurons: refining the morris water maze to study the functional relevance of adult hippocampal neurogenesis. ISSN 16624548
- Guazzelli A, Bota M, Corbacho FJ, Arbib MA (1998) Affordances, motivations, and the world graph theory. *Adapt Behav*. <https://doi.org/10.1177/105971239800600305>. ISSN 10597123
- Hafting T, Fyhn M, Molden S, Moser MB, Moser EI (2005) Microstructure of a spatial map in the entorhinal cortex. *Nature*. <https://doi.org/10.1038/nature03721>. ISSN 00280836
- Harland B, Contreras M, Fellous JM (2017) A role for the longitudinal axis of the hippocampus in multiscale representations of large and complex spatial environments and mnemonic hierarchies. In: The hippocampus-plasticity and functions. IntechOpen
- Hoydal OA, Skytøen ER, Moser M-B, Moser EI (2018) Object-vector coding in the medial entorhinal cortex. *bioRxiv*, p 286286. <https://doi.org/10.1101/286286>. ISSN 1476-4687. <https://www.biorxiv.org/content/early/2018/03/22/286286>
- Jadhav SP, Kemere C, German PW, Frank LM (2012) Awake hippocampal sharp-wave ripples support spatial memory. *Science*. <https://doi.org/10.1126/science.1217230>. ISSN 10959203
- Johnson A, Redish AD (2007) Neural ensembles in CA3 transiently encode paths forward of the animal at a decision point. *J Neurosci Off J Soc Neurosci*. <https://doi.org/10.1523/JNEUROSCI.3761-07.2007>. ISSN 1529-2401
- Jung MW, Wiener SI, McNaughton BL (1994) Comparison of spatial firing characteristics of units in dorsal and ventral hippocampus of the rat. *J Neurosci* 14(12):7347–7356
- Kjelstrup KB, Solstad T, Brun VH, Hafting T, Leutgeb S, Witter MP, Moser EI, Moser MB (2008) Finite scale of spatial representation in the hippocampus. *Science*. <https://doi.org/10.1126/science.1157086>. ISSN 00368075
- Kretschmar RM, Anderson CW (1997) Comparison of CMACs and radial basis functions for local function approximators in reinforcement learning. In: IEEE International conference on neural networks—conference proceedings, vol 2, October 2014, pp 834–837. <https://doi.org/10.1109/ICNN.1997.616132>. ISSN 10987576
- Krzywinski M, Altman N (2014) Points of significance. Nature Publishing Group, Berlin, Visualizing samples with box plots

- Leonard TK, Mikkila JM, Eskandar EN, Gerrard JL, Kaping D, Patel SR, Womelsdorf T, Hoffman KL (2015) Sharp wave ripples during visual exploration in the primate hippocampus. *J Neurosci*. <https://doi.org/10.1523/jneurosci.0864-15.2015>. ISSN 0270-6474
- Lever C, Burton S, Jeewajee A, O'Keefe J, Burgess N (2009) Boundary vector cells in the subiculum of the hippocampal formation. *J Neurosci*. <https://doi.org/10.1523/jneurosci.1319-09.2009>. ISSN 0270-6474
- Llofriu M, Tejera G, Contreras M, Pelc T, Fellous JM, Weitzenfeld A (2015) Goal-oriented robot navigation learning using a multi-scale space representation. *Neural Netw* 72:62–74. <https://doi.org/10.1016/j.neunet.2015.09.006>. ISSN 18792782
- Llofriu M, Scleidorovich P, Tejera G, Contreras M, Pelc T, Fellous JM, Weitzenfeld A (2019) A computational model for a multi-goal spatial navigation task inspired in rodent studies. In: *IJCNN*
- Lozano-Pérez T, Wesley MA (1979) An algorithm for planning collision-free paths among polyhedral obstacles. *Commun ACM*. <https://doi.org/10.1145/359156.359164>. ISSN 00010782
- 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(8):729–744. <https://doi.org/10.1002/hipo.22132>. ISSN 10509631
- Maurer AP, VanRhoads SR, Sutherland GR, Lipa P, McNaughton BL (2005) Self-motion and the origin of differential spatial scaling along the septo-temporal axis of the hippocampus. *Hippocampus*. <https://doi.org/10.1002/hipo.20114>. ISSN 10509631
- McNaughton BL, Barnes CA, Gerrard JL, Gothard K, Jung MW, Knierim JJ, Kudrimoti H, Qin Y, Skaggs WE, Suster M, Weaver KL (1996) Deciphering the hippocampal polyglot: the hippocampus as a path integration system. *J Exp Biol* 199(1):173–185
- Muller R, Kubie J (1987) The effects of changes in the environment on the spatial firing of hippocampal complex-spike cells. *J Neurosci*. <https://doi.org/10.1523/jneurosci.07-07-01951.1987>. ISSN 0270-6474
- O'Keefe J, Burgess N (1996) Geometric determinants of the place fields of hippocampal neurons. *Nature*. <https://doi.org/10.1038/381425a0>. ISSN 00280836
- O'Keefe J, Dostrovsky J (1971) The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Res*. [https://doi.org/10.1016/0006-8993\(71\)90358-1](https://doi.org/10.1016/0006-8993(71)90358-1). ISSN 00068993
- O'Keefe J, Nadel L (1978) *The hippocampus as a cognitive map*. Oxford University Press, Oxford
- Pfeiffer BE, Foster DJ (2013) Hippocampal place-cell sequences depict future paths to remembered goals. *Nature*. <https://doi.org/10.1038/nature12112>. ISSN 1476-4687
- Poppenk J, Evensmoen HR, Moscovitch M, Nadel L (2013) Long-axis specialization of the human hippocampus. ISSN 13646613
- Pouderoux J, Gonzato J-C, Tobor I, Guitton P (2004) Adaptive hierarchical RBF interpolation for creating smooth digital elevation models. In: *Proceedings of the 12th annual ACM international workshop on Geographic information systems*, pp. 232–240. <https://doi.org/10.1145/1032222.1032256>
- Rivard B, Li Y, Lenck-Santini P-P, Poucet B, Muller RU (2004) Representation of objects in space by two classes of hippocampal pyramidal cells. *J Gen Physiol*. <https://doi.org/10.1085/jgp.200409015>. ISSN 0022-1295
- Savelli F, Yoganarasimha D, Knierim JJ (2008) Influence of boundary removal on the spatial representations of the medial entorhinal cortex. *Hippocampus*. <https://doi.org/10.1002/hipo.20511>. ISSN 10509631
- Solstad T, Boccara CN, Kropff E, Moser M-B, Moser EI (2008) Representation of geometric borders in the entorhinal cortex. *Science*. <https://doi.org/10.1126/science.1166466>. ISSN 0036-8075
- Strange BA, Witter MP, Lein ES, Moser EI (2014) Functional organization of the hippocampal longitudinal axis. ISSN 14710048
- Sutton RS, Barto AG (2018) *Reinforcement learning: an introduction*. MIT Press, Cambridge. ISBN 9780262039246
- Sutton R, Mcallester D, Singh S, Mansour Y (2000) Policy gradient methods for reinforcement learning with function approximation. *NIPS*. AT&T Labs - Research, Florham Park. <https://doi.org/10.1177/004728757601400384>. ISSN 0047-2875
- Taube JS, Muller RU, Ranck JB (1990) Head-direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis. *J Neurosci Off J Soc Neurosci*. <https://doi.org/10.1523/jneurosci.07-07-01951.1987>. ISSN 0270-6474
- Vorhees CV, Williams MT (2006) *Morris water maze: Procedures for assessing spatial and related forms of learning and memory*. Nat Protoc. <https://doi.org/10.1038/nprot.2006.116>. ISSN 17542189

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.