**Title**: Real-Time Sensory-Motor integration of Hippocampal Place Cell Replay and Prefrontal Sequence Learning in Simulated and Physical Rat Robots for Novel Path Optimization

**Authors**: Nicolas Cazin[1,2], Pablo Scleidorovich[3], Alfredo Weitzenfeld[3], Peter Ford Dominey[1,2*]

1. INSERM, U1093, Cognition Action Plasticité Sensorimotrice, Université de Bourgogne, Dijon, France
2. Robot Cognition Laboratory, Institut Marey, INSERM U1093 CAPS, UBFC, Dijon, France
3. College of Engineering, University of South Florida, USA

*Corresponding author: peter.dominey@inserm.fr

**Abstract**: An open problem in the cognitive dimensions of navigation concerns how previous exploratory experience is re-organized in order to allow the creation of novel efficient navigation trajectories. This behavior is revealed in the "traveling sales-rat problem" (TSP) when rats discover the shortest path linking baited food wells after a few exploratory traversals. We recently published a model of navigation sequence learning, where sharp wave ripple (SWR) replay of hippocampal place cells transmit "snippets" of the recent trajectories that the animal has explored to the prefrontal cortex (PFC) (Cazin et al. 2019). PFC is modeled as a recurrent reservoir network that is able to assemble these snippets into the efficient sequence (trajectory of spatial locations coded by place cell activivation). The model of hippocampal replay generates a distribution of snippets as a function of their proximity to a reward, thus implementing a form of spatial credit assignment that solves the TSP task. The integrative PFC reservoir reconstructs the efficient TSP sequence based on exposure to this distribution of snippets that favors paths that are most proximal to rewards. While this demonstrates the theoretical feasibility of the PFC-HIPP interaction, the integration of such a dynamic system into a real-time sensory-motor system remains a challenge. In the current research we test the hypothesis that the PFC reservoir model can operate in a real-time sensory-motor loop. Thus, the main goal of the paper is to validate the model in simulated and real robot scenarios. Place cell activation encoding the current position of the simulated and physical rat robot feeds the PFC reservoir which generates the successor place cell activation that represents the next step in the reproduced sequence in the readout. This is input to the robot, which advances to the coded location and then generates de-novo the current place cell activation. This allows demonstration of the crucial role of embodiment. If the spatial code readout from PFC is played back directly into PFC, error can accumulate, and the system can diverge from desired trajectories. This required a spatial filter to decode the PFC code to a location and then recode a new place cell code for that location. In the robot, the place cell vector output of PFC is used to physically displace the robot and then generate a new place cell coded input to the PFC, replacing part of the software recoding procedure that was required otherwise. We demonstrate how this integrated sensory-motor system can learn simple navigation sequences, and then, importantly, how it can synthesize novel efficient sequences based on prior experience, as previously demonstrated (Cazin et al. 2019). This contributes to the understanding of hippocampal replay in novel navigation sequence formation, and the important role of embodiment.

## 1. Introduction:

As rats learn to search for multiple sources of food or water in a complex environment, processes of spatial sequence learning and recall in the hippocampus (HC) and prefrontal cortex (PFC) are taking place. An open problem in the cognitive dimensions of navigation concerns how previous exploratory experience is re-organized in order to allow the creation of novel efficient navigation trajectories. This behavior is revealed in the "traveling sales-rat problem" when rats discover the shortest path linking baited food wells after a few exploratory traversals. Figure 1 illustrates a schematic view of the TSP problem in rat navigation.

Recent studies show that spatial navigation in the rat hippocampus involves the replay of place cell firing during awake and sleep states generating small sequences of spatially related place cell activity we call "snippets". We introduce this term as shorthand for "place cell activation during sharp wave ripples", and also to recall the notion that they have been "snipped out" of the larger navigation sequence. These "snippets" occur primarily during sharp-wave-ripple (SWR) events. Much attention has been paid to replay during sleep in the context of long term memory consolidation. Here we focus on the role of replay during the awake state, as the animal is learning across multiple trials. We hypothesize that these "snippets" can be used by the PFC to achieve multi-goal spatial sequence learning. We recently published an integrated model of hippocampus and PFC that is able to form spatial navigation sequences based on snippet replay (Cazin et al 2019), and now for the first time test this model in a real-time sensory-motor integration, both in a simulated robot, and a physical robot in order to reproduce rat TSP behavior. This extends our existing spatial cognition robotic model for simpler goal-oriented tasks (Barrera et al 2011, Barrera & Weitzenfeld 2008) with a new replay-driven model for memory formation in the hippocampus and spatial sequence learning and recall in PFC. The goal of the current research is to validate this model in simulated and real robot scenarios. This will allow us to learn how physical embodiment at a real location in space after a displacement, vs. a simulated estimation of the location, can impact the system.

*** Insert Figure 1: TSP problem. ***

In contrast to existing work on sequence learning that relies heavily on sophisticated learning algorithms and synaptic modification rules, we proposed to use an alternative computational framework known as 'reservoir computing' (Dominey 1995) in which large pools of prewired neural elements process information dynamically through reverberations. Such a reservoir computational model consolidates snippets into larger spatial sequences that may be later recalled by subsets of the original sequences (Cazin et al 2019). The resulting integrated system has allowed us to confirm in an embodied robot context that snippets derived from multiple sequences could be consolidated into a single novel sequence that corresponds to the shortest path linking a set of rewarded targets. Perhaps more importantly, we observed an interesting advantage for the model when operating in the embodied-cognitive real-time framework of a robot, illustrated in Figure 2, that will be explained in more detail below.

*** Insert Figure 2. Rodent, simulation and robot experiments ***

Figure 2 illustrates the relation between the rat model, simulation, and robotics. In all these cases, the behaving agent is in an arena with rewarded locations, and the target behavior is the discovery of the efficient path that links these rewarded locations. In the pursuit of a robotic demonstration of this capability using reservoir computing and reward-modulated replay, the following sections overview the key background areas for the proposed research: (1) sleep and awake replay in rats, (2) place cell models of spatial cognition, and (3) reservoir computing and sequence learning.

## 1.1 Snippet Replay in Rats

The hippocampus stores information during the acquisition of new memories and these memories are replayed as snippets – sequences of place cell activations - during sleep as part of a memory consolidation process (Buzsáki 1989, Nadel & Moscovitch 1997, Wilson & McNaughton 1994). Consolidation is believed to involve synaptic changes reflecting the integration and refinement of memory representations (McClelland et al 1995), likely facilitated by replay. This replay involves neural populations that were active during a task immediately preceding the sleep period. Reactivations of specific neural activity patterns

during sleep have been observed in several brain areas including the hippocampus, amygdala, neocortex and striatum e.g. (Bendor & Wilson 2012, Carr et al 2011, Euston et al 2007, Foster & Wilson 2006, Hoffman & McNaughton 2002, Ji & Wilson 2007, Peyrache et al 2009). Other evidence suggests that replay may also occur during the awake state indicating online memory processes or the planning of behaviors yet to be performed (Carr et al 2011, Davidson et al 2009, Diba & Buzsaki 2007, Gupta et al 2010). In the hippocampus, it has been shown that reactivation occurs primarily in a compressed manner, during the occurrence of fast (150-200 Hz) and short (60-120ms) oscillations called sharp waves/ripples complexes (SWR). Different subsets of cells reactivate in different SWRs, each cell emitting only a few spikes. The interspike interval between reactivating cells is within the range of that required to induce spike-timing dependent synaptic plasticity (STDP). One hypothesis therefore is that the sequence of reactivation episodes allows for online (awake replay, focus of this research) and offline (sleep replay) synaptic modifications that will eventually lead to the consolidation and integration of specific memory items.

Interestingly, the presence of rewards increases replay in hippocampus and ventral striatum (Lansink et al 2008, Singer & Frank 2009), suggesting an interaction between reinforcement learning and replay (De Lavilléon et al 2015, Johnson & Redish 2005). The computational mechanisms underlying these reactivations and their possible consequences on learning have been investigated in hippocampus (Hasselmo 2008, Johnson & Redish 2005). It was suggested that replay may improve reinforcement learning algorithms, in simple tasks in which rats had to memorize a sequence of turns in a corridor-type maze with a single goal location (Johnson & Redish 2005). Cazé et al (2018) provide an extended review of different forms of replay and their potential contributions and manifestations in different forms of model free and model based reinforcement learning. This is of interest in the context of our claim that replay can implement a simple form of reward propagation for reinforcement learning.

Most of the replay events occur in the forward direction (place cells activate in the same order as they would activate if the rat was navigating through them), before a movement is initiated, while a smaller fraction occur in the backward direction at or near reward sites. Interestingly, forward replay was found to be more directly correlated with the actual path of the animal than backward replay (Diba & Buzsaki 2007).

Our working hypothesis in Cazin et al. (2019) is that forward replay during active learning triggers the formation of 'snippets', short sequences of place cell firings ordered in the order they were traversed. We also hypothesize that backward replay allows for 'credit assignment' of the snippets when a reward is obtained, hence reinforcing those paths that yielded a reward (Foster & Wilson, 2006), presented in more detail below.

## 1.2 Computational Models of Spatial Cognition in Hippocampus and Robotics Experimentation

The study of behavioral and neurophysiological mechanisms in rats responsible for spatial cognition has inspired the development of many computational models of hippocampus place cells in the context of goal-oriented learning tasks in robotic systems. A number of models have been developed including those of (Arleo & Gerstner 2000, Barrera et al 2011, Barrera et al 2015, Barrera & Weitzenfeld 2008, Brown & Sharp 1995, Burgess et al 1994, Caluwaerts et al 2012, Dollé et al 2010, Gaussier et al 2007, Gaussier et al 2002, Guazzelli et al 1998, Redish & Touretzky 1997). An extensive review of models of spatial cognition in the hippocampus can be found in Moser et al (2017). Here we focus on the process of path optimization based on replay that is modulated by a novel form of reward propagation.

In this context, we (Barrera et al 2015) described a model for spatial cognition in open arena environments, i.e. having no corridors, as in the Morris water maze. Recently, the computational model was extended to incorporate grid and head direction cells, i.e. "biological odometry" as described in Tejera et al (2018). This system provides the robotic simulation framework for our integrated experiments linking a sensory-motor system to the PFC reservoir model for sequence generation. Additionally, in (Llofriu et al 2015) we describe a place cell model applied to a multiple goal task where a subset of feeders need to be visited in random order. This model does not include any replay of snippets, and does not involve learning a particular sequence of feeder visits. In the current research we use a simple model of place cell coding based on a uniform tessellation of the arena, described below.

## 1.3 Reservoir Computing and Sequence Learning

Reservoir computing refers to a class of neural network models in computational neuroscience and machine learning (Lukosevicius & Jaeger 2009). These systems are characterized by a sparsely connected recurrent network of neurons (spiking or analog), with

fixed connection weights (excitatory and inhibitory). Because of the recurrent connections, this "reservoir" is a dynamical system that has inherent sensitivity to the serial and temporal structure of input sequences. Reservoir neurons are connected to readout neurons by modifiable connections, and these can be trained in different tasks (e.g. sequence recognition, prediction, classification). The first instantiation of such models was by Dominey (Dominey 1995), with the reservoir corresponding to recurrent prefrontal cortical networks, and the modifiable readout connections corresponding to the corticostriatal projections, with dopamine-modified synapses (Dominey 1995, Dominey et al 1995). These models addressed sensorimotor sequence learning, and demonstrated the inherent sensitivity of these recurrent systems to serial and temporal structure in motor behavior and in language (Dominey 1998a, Dominey 1998b, Dominey et al 2009, Dominey & Ramus 2000, Hinaut & Dominey 2013) Maass developed a related approach with spiking neurons and demonstrated the non-linear computational capabilities of these systems (Maass et al 2002). In the machine learning context, Jaeger demonstrated how such systems have inherent signal processing capabilities (Jaeger & Haas 2004), including important robustness to noise such as interference across multiple consecutive symbols, nonlinear distortion, and additive white Gaussian noise. Interestingly, these reservoir properties appear to be found in cortex. For example, electrophysiological studies have revealed that cortical neurons in primary sensory areas (e.g. V1) have reservoir properties of fading memory (Nikolic et al 2009). That is, stimuli presented in the past tend to resonate in the recurrent network and influence the processing of subsequent stimuli. Equally interestingly, when these networks are exposed to inputs with multiple dimensions (e.g. target identification, serial order, match/non-match) neurons represent non-linear mixtures of these dimensions (Dominey et al 1995, Rigotti et al 2013). Such nonlinear mixed effects have recently been seen in primate frontal cortex (Rigotti et al 2013). This argues in favor of a reservoir-like function in recurrent networks of the cortex in general, and in prefrontal cortex specifically. We have demonstrated how such recurrent networks can learn about sequential and temporal structure (Dominey 1998b), including serial order regularities that are expressed in sequence segments (Dominey & Ramus 2000). In Cazin (Cazin et al 2019), reservoir computing has been exploited in terms of its inherent ability to allow the concatenation of multiple contiguous subsequences into a coherent sequence, thus addressing a major open question in navigation trajectory learning.

In the current research, the PFC model is a recurrent reservoir network, that has the useful property of maintaining a fading history of past internal states due to the recurrent

connections (Cazin et al 2019). It is trained to take a place cell code of the current location as input, and to generate the next position in the learned trajectory, coded in a place cell code in the output neurons. The model can learn an integrated version of a single sequence, even when trained on randomly presented snippets or subsequences of the global sequence. In the TSP problem, this integration capability can allow the system to generate the efficient sequence based on training from snippets issued from sequences that contain parts of the efficient sequence.

### 1.4 A Potential Role of Embodiment

Robotic validation can allow us to determine how such learning may be facilitated by embodiment – the interplay of information and physical processes – when models are allowed to operate in the physical world (Pfeifer et al 2007). In this context, recurrent networks are typically used to generate time series, and can also typically suffer from instability as small errors in the output are re-injected into the input and eventually accumulate and produce instability (Jaeger et al 2007). We encountered this problem in (Cazin et al 2019) and implemented a denoising process called the spatial filter, which translated the output coded place-cell vector into its corresponding Cartesian coordinates, and then recoded this as a place cell vector that was reinjected as input to the PFC reservoir. In the current research we can predict that using a form of physical embodiment in the robot may contribute to denoising by physical computation. That is, the robot will physically transform the place cell code of the next trajectory position into a movement to that location. Then the place cell code corresponding to that location is generated and used as input in the next time cycle of the reservoir. Thus, part of the objective of this research is to determine these potentially beneficial effects of physical embodiment on the functioning of the PFC-HIPP model.

We first test the model with a simulation of the robot, and then perform 5 experiments with the physical robot. Two experiments examine simple sequence learning and reproduction. Three others address the optimization problem, based on the TSP – traveling salesrat problem. Here the system is run through three sequences that each contain part of the optimal sequence. Based on the reward propagation learning, we test whether the PFC-HIPP model can integrate these subsequences to generate the novel efficient sequence.

## 2. Methods -

Our approach to answering these questions about the impact of robotic embodiment on our sequence learning model of reward-modulated replay in hippocampus and reservoir computing in PFC (see Figure 3) involves testing the model in a real-time sensorimotor loop. The tests involve navigation between multiple goals (baited cups in rat experiments and rewarded floor landmarks in a robot arena – see Figures 1 and 2).

*** Insert Figure 3. Hippocampus-PFC model architecture. ***

### 2.1 Replay-Driven Model of Spatial Sequence Learning in the Hippocampus-PFC network using Reservoir Computing

The computational model consists of 3 main components: Hippocampus Replay module, PFC Reservoir and readout for Sequence Learning, and the spatial filter. The goal of the model is to implement online replay-based memory acquisition in the Hippocampus and reservoir-computing based sequence learning in PFC. The Hippocampus generates snippets of place fields which are used in the PFC reservoir to reconstruct complete spatial sequence trajectories. After successful learning, when primed with a subsequence, the PFC completes that initial trajectory as the system becomes entrained by the input sub-sequence. The main processing steps of the Hippocampus-PFC network model are described below, following Cazin et al. (2019).

*Hippocampal replay based on proximity to rewards*

The function of the snippet replay model is to favor snippets that are on efficient paths linking rewarded sites (e.g. paths linking feeders A, B and C in panel B, Figure 1), and not those that are on inefficient paths (as in paths linking C, D and E in the same panel). This is achieved in two phases, first by propagating errors backwards from rewarded locations by reverse replay, and then second, by generating snippets to train the PFC by calculating the probability of replay as a function of proximity to these propagated rewards. This backward propagation of distant rewards along spatial trajectories towards the start yields a computationally simple form of reinforcement learning.

Hippocampus replay phenomenon observed during SWR complexes in active rest phase is modeled by a training set made of condensed subsequences of place-cell activation patterns replayed at random. Sequences are represented with a resolution of 20 samples/meter. Each update cycle of the reservoir is considered to correspond to 5 ms of real time. At this rate, replay occurs at a rate corresponding to 10 meters/second, which is in the range of that observed in behaving animals (Davidson et al 2009).

The distribution that is sampled for drawing a random place-cell activation pattern might be uniform or modulated by new or rewarding experience as described in (Carr et al 2011). In particular, we model a random replay based on reward. In (Ambrose et al 2016) the authors show that during SWR sequences place-cell activation occur in reverse order at the end of a run. We will focus on reward modulated and reverse replay.

A given location in the 2D space generates a place-cell activation pattern in a set of 2D (16x16) Gaussian place-fields illustrated in Figure 2B. We define a snippet as the concatenation of successive place-cell activation patterns:

$$S(n;s) = X_{in}(t_n \rightarrow t_{n+s}) \quad (1)$$

Where $X_{in}$ is the place cell activation vector that will be input to the reservoir, $n$ is the offset into the trajectory, and $s$ is the number of concatenated place-cell activation patterns.

In our model of hippocampus replay, we define a time budget noted $T$ that corresponds to the duration of a replay episode. A replay episode $E$ is a set of snippets of length $s$:

$$E(s) = \{S(n;s)\} \quad (2)$$

The sum of the durations of snippets replayed in $E$equals at least to $T$. If the time budget is exceeded; one snippet is truncated in order to fit the time budget.

Hippocampus place-cell replay can occur in forward or backward direction as suggested in (Foster & Wilson 2006). We use reverse replay to propagate reward backwards along the experience trajectory, and model the reverse replay as follows: For a given trajectory k of $N_k$ samples, there are $N_k - s$ possible snippets that could potentially be generated from that trajectory, but only a limited number of snippets will actually be selected to fit the time budget $T$. A snippet $S(n)$ has a likelihood of being replayed as a function of its proximity to a reward. This is implemented by a generative model of snippet replay likelihood that is first learnt by propagating time delayed reward information according to the replay direction and the snippet duration. Using a reverse replay rate ($\beta_{learn}$) of 1 implements uniquely reverse

replay. Reward is propagated along the snippet in the reverse direction during this model building phase for the snippet replay likelihood.

The reward prediction vector $V(t_1 \rightarrow t_{N_k})$ is learnt by initializing it to small positive random values and then iteratively refined by applying the replay procedure below K times:

1. Draw a random contiguous time index subset $\tau \equiv T(n, s, r; \beta_{learn})$ according to the reverse rate $\beta_{learn}$:
   a. Select a time-step $t_n$such that $n \in \{1 \dots N_k\}$Erreur ! Signet non défini. according to the replay likelihood defined by:

$$P(t_1 \rightarrow t_{N_k}) = \frac{V(t_1 \rightarrow t_{N_k})}{\sum_{i=1}^{N_k} V(t_i)} \quad (3)$$

   b. Select a random number $r \in [0, 1]$ and a contiguous and monotonous time index sequence $S$ such that:

$$S = \begin{cases} t_{\max(1,n-s+1)} \rightarrow t_n, & r < \beta_{learn} \\ t_n \rightarrow t_{\min(N_k,n+s)}, & r \geq \beta_{learn} \end{cases} \quad (4)$$

2. Update the reward estimate V over increasing indices of $\tau$ by computing the update equation:

$$V(S_k) = \alpha\big(R(S_{k-1}) + \gamma V(S_{k-1})\big) + (1 - \alpha)V(S_k) \quad (5)$$

Where:

- $\alpha \in [0, 1]$ is the learning rate constant
- $\gamma \in [0, 1]$ is the discount constant
- $R(t)$ is the observed instantaneous reward information

(6)

This is a convex combination of the current estimate of the reward information $V(S_k)$ at the next time step and the instantaneous reward information $R(S_{k-1}) + \gamma V(S_{k-1})$ based on the previously observed reward signal $R(\tau_{k-1})$ and delayed previous reward estimate $\gamma V(S_{k-1})$). This implements a form of temporal difference learning. It is sufficient to define a coarse reward signal as:

$$R(t) = \begin{cases} 1 \; if \; a \; baited \; feeder \; is \; encountered \; at \; time \; t \\ 0 \; otherwise \end{cases}$$

The snippet generation procedure is simply the repetition of the steps a and b of the replay procedure with $\beta_{generate}$ used instead of $\beta_{learn}$ until the sum of time subsequences durations overflows a fixed time budget duration.

The net result of this replay model, is that a snippet's probability of replay into the sequence learning model increases with its proximity to a reward. This will favor efficient trajectories with short distances between rewards, and will disfavor long and inefficient paths between rewards.

*Reservoir model of PFC for snippet consolidation*

We model the prefrontal cortex as a recurrent reservoir network. The version that we use to model the frontal cortex employs leaky integrator neurons in the recurrent network. At each time-step the network is updated as illustrated in Figure 3, and described here.

The modelled hippocampus place-cells projects into the reservoir through feed-forward synaptic connections noted $W_{ffwd}$. The projection operation is a simple matrix-vector product. Hence, the input projection through feed-forward synaptic connections is defined by:

$$U_{ffwd}(t_n) = W_{ffwd} * X_{in}(t_n) \quad (7)$$

Where:

- $W_{ffwd}$ is a fixed connectivity matrix whose values do not depend on time.

Synaptic weights are randomly selected at the beginning of the simulation. Various probability density functions (PDF) could be sampled and one condition about on $W_{ffwd}$ is its bijectivity, i.e. every stimulus $X_{in}(t_n)$ must have a distinct image through $W_{ffwd}$and each $U_{ffwd}$must correspond to a unique $X_{in}(t_n)$. Practically speaking (Lukosevicius 2012), sampling $U[-1, 1]$ a uniform distribution is sufficient. A positive synaptic weight in a connectivity matrix models and excitatory connection and a negative weight models an inhibitory connection between two neurons. A synaptic weight equals to zero models no connection between two given neurons. A larger absolute value of the synaptic weights represents a reinforced correlation between firing patterns of those two neurons.

Let N be the number of neurons in the Reservoir. Reservoir neurons are driven by both sensorial input $X_{in}(t_n)$ and, importantly, by the recurrent connections that project an image of the previous reservoir state back into the reservoir. The recurrent projection is defined as:

$$U_{rec}(t_n) = W_{rec} * X_{res}(t_{n-1}) \quad (8)$$

Where:

- $W_{rec}$ is a N by N square connectivity matrix,
- $X_{res}$ is the reservoir activation

Synaptic weights are drawn from a $U[-1, 1]$ uniform distribution, scaled by a $S(N; K) = K\frac{1}{\sqrt{N}}$ factor, where K is the number of place cells in the output vector. The same sign convention as in equation (7) applies for the recurrent connectivity matrix.

Self-connections (i.e. $w_{rec}^{i,i}$ with $i \in 1 \ldots N$) are forced to zero. $W_{rec}$ is also fixed and its values do not depend on time. The contribution of afferent neurons to the reservoir neurons is summarized by

$$U_{res}(t_n) = U_{ffwd}(t_n) + U_{rec}(t_n) \quad (9)$$

Then, the membrane potential of reservoir neurons $P_{res}$is computed by solving the following ordinary differential equation (ODE):

$$\tau \frac{\partial P_{res}}{\partial t} = -P_{res}(t_{n-1}) + U_{res}(t_n) \quad (10)$$

Where:

- $\tau$ is the neuron's time constant. It models the resistive and capacitive properties of the neuron's membrane.

We will consider a contiguous assembly of neurons that share the same time constant. . By choosing Euler's forward method for solving equation (10), the membrane potential is computed recursively by the equation:

$$P_{res}(t_n) = h * U_{res}(t_n) + (1 - h) * P_{res}(t_{n-1}) \quad (11)$$

Where:

- $h = \frac{\partial t}{\tau}$ is called the leak rate

It is a convex combination between instantaneous contributions of afferents neurons $U_{res}(t_n)$ and the previous value $P_{res}(t_{n-1})$ of the membrane potential. The current membrane potential state carries information about the previous activation values of the reservoir, provided by the recurrent synaptic weights. The influence of the history is controlled by the leak rate. A high leak rate will result in a responsive reservoir with a very limited temporal line of sight. A low leak rate will result in a slowly varying network whose activation values depend more on the global temporal structure of the input sequence.

Finally, the mean firing rate of a reservoir's neuron is given by:

$$X_{res}(t_n) = \sigma_{res}(P_{res}(t_n); \Theta_{res}) \quad (12)$$

Where:

- $\sigma_{res}$ is the non-linear activation function of the reservoir neurons
- $\Theta_{res}$ is a bias that will act as a threshold for the neuron's activation function.

We choose a $\sigma_{res} \equiv tanh$ hyperbolic tangent activation function with a zero bias. This can generate negative outputs from these neurons. The recurrent weights are [-1,1] and so the resulting positive and negative inputs to neurons can be considered physiologically as direct excitatory inputs, or inputs from an inhibitory interneuron.

At this point, the reservoir's states carry information about its present and past activation values within a limited timeline of sight. One single reservoir neuron does not contain the whole information but its activation fluctuations carry partial information about the serial and temporal structure of the stimulus sequence. This property corresponds to mixed selectivity observed in cortical neurons in the primate, and is a fundamental signature of cortical processing (Rigotti et al 2013).

*Learning in Modifiable PFC Connections to Readout*

Based on the rich activity patterns in the reservoir, it is possible to decode the reservoir's state in a supervised manner in order to produce the desired output as a function of the input sequence. The expected output is only required to be an activation pattern that is temporally congruent to the input stimulus. This decoding is provided by the readout layer and the matrix of modifiable synaptic weights linking the reservoir to the readout layer, noted $W_{ro}$ and represented by dash lines in Figure 3.

The readout activation pattern $X_{ro}(t_n)$ is given by the equation:

$$X_{ro}(t_n) = \sigma_{ro}(W_{ro} * X_{res}(t_n); \Theta_{ro}) \quad (13)$$

Where:

- $\sigma_{ro}$ is the non-linear activation function of the readout neurons
- $\Theta_{ro}$ is a bias that will act as a threshold for the neuron's activation function

We choose a $\sigma_{ro} \equiv tanh$ hyperbolic tangent activation function with a zero bias.

Notice that the update schema describe above is a very particular schema inherited from feedforward neural networks. We chose to use it because it is computationally efficient and deterministic.

Once the neural network states are updated, the readout synaptic weights are updated by using a stochastic gradient descent algorithm. By deriving the Widrow-Hoff Delta rule (Widrow & Hoff 1960) for hyperbolic tangent readout neurons, we have the following update equation:

$$W_{ro}(t_n) = W_{ro}(t_{n-b+1}) + \alpha * X_{res}(t_{n-b+1} \to t_n) * \left(\mathrm{X_{ro}}(t_{n-b+1} \to t_n) - X_{des}(t_{n-b+1} \to t_n)\right) * (1 - \mathrm{X_{ro}}(t_{n-b+1} \to t_n)^2) \quad (14)$$

Where:

- $X_{des}$ is the desired read-out
- $\alpha$ is a small positive constant called the learning rate
- $t_{n-b+1} \to t_n$ is the concatenation of b time steps from $t_{n-b+1}$to $t_n$

When $b = 1$, equation (14) computes a stochastic gradient descent. The case when $b > 1$ is called a mini-batch gradient descent (Ruder 2016) and allows one to estimate the synaptic weight gradient base on a b successive observations of predicted and desired activation values. A mini batch gradient allows one to compute efficiently and robustly the synaptic weight gradient. Empirically, $b = 32$ gives satisfying results.

In this study, we will focus on the prediction of the next place-cell activation pattern:

$$X_{des}(t_n) = X_{in}(t_{n+1}) \quad (15)$$

The learned output thus codes the next spatial location in a place cell code.

## 2.2 System Architecture for Robot Sequence Learning

In order to test the model in a real-time sensorimotor loop, the model was integrated into a robotic system, first in simulation, then with a physical robot (described in the next section).

*** Insert Figure 4 ***

### 2.2.1 System integration

The integration of the reservoir model with the simulator (and robot) is illustrated in Figure 4. The upper portion of the figure represents the physical or simulated robots. They simulate the movement of a rat in a 2x2m square arena. The rat's internal representation of this space is realized by a regular 16x16 grid of place cells on this surface (see Figure 1). The Spatial Cognition System (SCS) module generates a sequence of place cell (PC) field activations that begins when the rat is placed in the maze and finishes when the rat has visited all feeders. PC activations are implemented as Gaussian fields, where the activation of a cell is strongest at the center and becomes weaker the further away. During each episode the SCS module: (a) calculates the activation of all place cells (which we call an activation pattern), (b) records the activation pattern along with the reward received, and (c) performs navigation actions. After the completion of each episode the navigation module sends an input sequence to the reservoir module consisting of all the PC activation patterns and rewards received.

### 2.2.2 Training

In order to test the reservoir model in the real-time loop with the simulator, the simulator was first directed to follow the three trajectories corresponding to B-D in Figure 1. This generated a set of stored trajectories (including the rewards) that were used for training the reward propagation replay model. Once the replay module was trained, it was then used to generate snippets, where the replay probability of a given snipped was based on its proximity to reward. The results of an experiment with 1000 robot simulations are illustrated in Figure 5.

*** Insert Figure 5 ***

### 2.2.3 Evaluation

In each of the 1000 tests, the hippocampal replay model was exposed to the three inefficient trajectories linking the rewarded locations. The hippocampal reward propagation algorithm was applied to these trajectories. The resulting distributions of replay probability are

illustrated in Figure 5A. The height of the peaks represent the replay likelihood, and time is represented on the horizontal axis, flowing from left to right. The three sequences are represented in a color code, with **ABC**ED in blue, E**BCD**A in pink and BA**CDE** in yellow. This illustrates how the propagation of reward in replay favors those parts of each trajectory that link rewarded locations by short subsequences (corresponding to the peaks and mountains in the replay count profiles (Figure 5A)). The reservoir model is trained on snippets generate according to this distribution.

Panel B illustrates the superposition of the 1000 executions of the integrated reservoir model and robot simulator based on this training. We observe that the model successfully extracts the efficient subsequences in order to generate a novel trajectory that links the 5 baited feeders in the efficient sequence, thus solving the TSP problem. This was confirmed by a statistical comparison of the Fréchet distance (Wylie 2013) between the generated sequence and each of the four illustrated in Figure 1 A-D. A Kruskal-Wallis comparison confirms that the trajectories generated autonomously are significantly more similar to the target sequence ABCDE than to the experienced non-efficient sequences ($p < 0.0001$).

This provides evidence that when place cell codes are used as the output from the reservoir to the simulated rat, and as input from the simulated rat to the reservoir, the system can properly operate in this sensory-motor loop.

### 2.3 Robot Experiments

Based on the successful demonstration that the reservoir model could interact with a simulated robot, we then proceeded to determine if the model would be robust to conditions of real sensors and actuators inherent to a physical robot.

#### 2.3.1 Integration in the robot platform environment

The interaction between the reservoir model and the physical robot requires the proper management of the place cell codes generated by the model, processed by the robot, and then output from the robot to the model in the sensorimotor loop. The corresponding infrastructure and algorithm for managing this interaction is executed in a distributed manner as shown in Figure 6. Three main processes are executed in parallel in different nodes of the network. The node SSL-Vision processes images from a camera to track the position of the robot. The Spatial Cognition System Node (SCS) is in charge of controlling the experiments, executing

509 the model and sending the commands to the robot. Finally, the robot node listens for
510 commands and executes them.

511

512 *** Insert Figure 6 ***

513

514 *Robot Platform*

515 The robot used in the experiments was assembled at USF from "off-the-shelf" components. It
516 consists of a differential wheeled robot running a ROS (Robotic Operating System) listener
517 node on a Raspberry Pi 3 board. A Raspbian Stretch image[1] with ROS already installed on it
518 was burnt onto the micro SD card of the board. On top of the robot, a color marker was added
519 to track the robot as illustrated in Figure 2B. The listener node running in the robot waits for
520 oncoming messages from SCS. The messages control the linear and angular speeds of the
521 robot.

522 *Sensory Feedback via Robocup Small Size League (SSL) Vision*

523 The experiments performed use a camera placed on top of the maze in order to track the
524 position of the robot. The position is used for: a) Logging the path performed by the robot, b)
525 Calculating the activation of each place cell, and c) Navigating to a given coordinate.

526 Tracking is done by placing a color marker on top of the robot (as seen in Figure 2B), and
527 using the Robocup Small Size League Vision software (SSL-Vision)[2] to find the position of
528 the robot. SSL-Vision makes position detections available via google protocol buffers sent as
529 multicast messages.

530 *Spatial Cognition System (SCS)*

531 Node SCS is in charge of controlling experiments, executing the model and controlling the
532 robot. The execution of the model varies between trials that train the reservoir and trials that
533 assess the reservoir, and the details of these interactions is schematically illustrated in Figure
534 7.

[1] https://medium.com/@rosbots/ready-to-use-image-raspbian-stretch-ros-opencv-324d6f8dcd96
[2] https://github.com/RoboCup-SSL/ssl-vision/wiki

*** Insert Figure 7 ***

2.3.2 Training and Assessment

*Training Episode using pre-recorded paths :*

On training trials, the robot recreates pre-recorded trajectories synthesizing those from rat experiments, as in panels B-D Figure 1. On each execution cycle the robot moves from its current position to the next position in the trajectory using the information received from SSL-Vision. To do so, the robot first turns in place until it faces towards the next position, and then it moves forward until its distance from the target position is below a given threshold (3 cm). While moving forward, orientation readjustments are made by the robot to ensure that it always faces the target location. Once the target location is reached, the new place cell activation pattern is computed and used to train the reservoir along with the received reward. This process is repeated until the full pre-recorded trajectory is recreated by the robot.

To assess the noise between the target coordinates and the coordinates produced by the robot, a test was performed where the robot was assigned to go to 100 random points chosen from a uniform distribution over a circle of radius 1. The error was measured as the distance between the points. Results showed that the maximum error was of 2.96cm with mean, median and std of 0.88cm, 0.90cm and 0.42cm respectively.

*Training Episode using robot local camera for feeder-taxic behavior*

During a training episode using the camera for feeder-taxic behavior, the robot is assigned to visit a sequence of feeders in a specific order using a local camera to guide navigation. To do so, the robot uses the camera to search and identify feeders in the maze. Feeders are represented using PVC cylinders of about 9cm in diameter, which are covered with different color markings to allow identification as illustrated in Figure 8.

To identify feeders in an image, we use OpenCV to find blobs of color that are classified according to their color patterns. The result of the identification process provides a list of detections where each detection consists of the detected feeder's id and its centroid in the image. The centroid's coordinates are scaled to the range [-1 ,1] both for the x and y axis, where negative values represent pixels to the left and lower halves of the image and vice versa. The identification process is illustrated in Figure 8.

*** Insert Figure 8 ***

When the robot is assigned to go to a given feeder, first the robot checks whether there is an obstacle (previous feeder) in front of it using a forward-facing distance sensor. If there is (Figure 8 D), the robot moves backward until it has enough space to turn in place (at least 11cm from its front). Then (Figure 8E), the robot turns in place in counterclockwise manner until the next feeder's x coordinate is detected within the range [-0.5,0.5]. At that point (Figure 8F), the robot continues turning in place but using proportional control to control the angular speed of the robot using the feeder's x coordinate as the process variable with a setpoint of 0 and using lower and upper absolute bounds for the output speed. This process continues until the feeder's x coordinate is within the range [-0.2, 0.2], at which point (Figure 8G) the robot starts moving forward using the same mechanism (but with different constants) to control the angular speed, and using proportional control on the distance of the front sensor to control the linear speed of the robot until the distance to the feeder is smaller than 8cm, point (Figure 8H) at which the robot stops and considers it has reached the target feeder.

*Assessment:*

The interaction between robot and reservoir model is detailed in Figure 7. Assessment trials are similar to training trials, except that the next robot position is provided by the reservoir instead of using a pre-recorded trajectory. In order to initialize the reservoir, the first 10 positions of the trajectory are fixed. Then the reservoir generates the next position, and is put into a wait mode, where the current state is frozen. The position is used to drive the robot. After reaching this position, the place cell activation pattern is computed and passed to the reservoir to compute the next target position, with a single execution of Eqn. 11. The process is repeated until the reservoir indicates that the sequence is finished.

## 3. Results

A total of 5 experiments were performed assessing whether simulation results could be reproduced using a real robot. As in the simulations, each experiment was split into two phases. Phase 1 consisted of at least one training trial, where the robot replicated a pre-

recorded path on each trial. During each trial, the place field activation patterns were recorded and later used to train the reservoir. Phase 2 consisted of a single assessment trial where the trained reservoir was used to control the motion of the robot as explained above. All training and assessment trials together totaled over 160 separate runs of the robot. Experiment 1 used one prerecorded path to provide training data for the replay model, and then tested the ability to learn and reproduce that sequence. This was a validation experiment. Experiment 2 used three prerecorded paths to train the replay model (corresponding to the three non-efficient trajectories in Figure 1B-D). The reward modulated replay was then used to train the reservoir, and the trained reservoir was then used in the sensory-motor loop with the robot. Experiment 3 extended Experiment 1 by using the visually-guided feeder-taxic behavior that allowed the robot to autonomously follow a pre-specified sequence of feeders to generate the training data. Experiment 4 used the same visually guided behavior to run between the feeders ABCED, EBCDA, BACDE in order to provide the training data. Experiment 5 used a simplified version of this, with the sequences ABC, BCD, CDE. Parameters for the simulation are provided in Table 1.

### 3.1 Experiment 1 – Sequence reconstruction

Phase 1 of Experiment 1 consisted of one training trial on the trajectory linking feeders ABCDE. The trajectory was provided to the robot. Similarly to the simulations, the objective of this experiment was to assess whether the reservoir would still be able to reconstruct the sequence presented in Phase 1, even under the presence of noise introduced by the robot. Ten replications of the experiment were performed with different reservoir instances. Results are shown in Figure 9. The superposition of the ten robot executions of the pre-recorded trajectory is illustrated in Figure 9 panel A. The superposition of the ten robot reproductions of this trajectory based on learning in the reservoir is illustrated in panel B. In all cases the reservoir was capable of learning and replicating the path performed on Phase 1.

*** Insert Figure 9 ***

### 3.2 Experiment 2 – Sequence optimization in TSP

In Experiment 2, the objective was to assess whether the reservoir would be able to combine and optimize multiple trajectories performed by the robot. In this case, Phase 1 consisted of 3 trials, where each trial contained a subsequence of the optimal path. Here, ten replications of the experiment were performed. The three pre-recorded trial sequences that were used to drive the robot are illustrated in Figure 1 panels B-D. Again, each of these contained part of the efficient path (illustrated in Figure 1 panel A), but they also contained inefficient components (illustrated in blue in the trajectories in Figure 1). Thus, the challenge of the model is to use the reward-modulated replay in order to learn and extract the efficient components and synthesis of the efficient sequence.

Thus, the three trajectories were run, allowing the collection of the path and reward data that feeds into the replay module (Figure 4A). There, the replay algorithm propagates the reward backwards from each rewarded feeder. These reward values are then used as probabilities in selecting snippets for replay to be used to train the reservoir model. The result is that the snippets used to train the model are drawn from locations close to the rewards. This produces a tendency to favor the shortest sequences linking the five rewarded targets (see Figure 5 for an illustration of this effect).

As observed in Figure 9F, the reservoir was able to extract the optimal subsequences and combine them into an optimal sequence. These results demonstrate that the PFC-HIPP reservoir-replay model can indeed successfully be integrated in the real-time sensory-motor loop as required for physical robot control.

### 3.3 Experiment 3 - Visually Guided Sequence Reconstruction

Experiment 1 used a pre-established trajectory to guide the robot in the training phase. Here we use a visually guided navigation capability in order to drive the robot along the ABCDE trajectory in the training phase. Thus, the experience that the robot uses to learn from is self-generated, based on the taxic behavior toward the targets.

We performed 10 replications of the experiment using different random number seeds to initialize the connection weights in 10 instances of the reservoir. As seen in Figure 10, the superposition of 10 executed trajectories yields quite low variability, and the visually guided navigation is quite robust. Slightly more variability is observed in Figure 10 B illustrating the superposition of the 10 executions. Note in Figure 10A that there is very low variability or noise in the visually guided sequences executed by the robot.

*** Insert Figure 10 ***

### 3.4 Experiment 4 - Visually Guided Sequence Consolidation in TSP

Experiment 2 used 3 pre-established trajectories to guide the robot in the training phase. Each of those trajectories contained part of the efficient trajectory linking feeders ABCDE. Here we use visually guided sequences, illustrated in Figure 10 C-E, each of which contains part of the efficient trajectory linking feeders ABCDE. The objective is to determine whether the reservoir instances, in 10 separate executions of the experiment, can link these subsequences together to generate the efficient trajectory ABCED.

Comparison of these sequences that are used for training, with those issued from pre-recorded sequences in Figure 9A-C reveals an interesting difference. In Figure 9A-C, the sequences are more looping, and in particular the inefficient parts are long and wondering (as observed in the rat (de Jong et al 2011)). In the sequences generated by the robot visual guidance, illustrated in Figure 10, the trajectories between feeders are quite straight. In Figure 10F, showing the superposition of the 10 repetitions of the assessment trials, we see that the model has indeed extracted the efficient subsequence ABCD, but gets stuck at D. This indicates that after arriving at D, the activation within the reservoir state generates an output in the reservoir readout that does not correspond to a real location. We observe this behavior only in these more difficult situations involving consolidation of multiple potentially conflicting trajectories.

### 3.5 Experiment 5 - Visually Guided Sequence Consolidation in TSP

The objective of this experiment is to demonstrate that the HIPP-PFC model consolidates a complete sequence from training on individual trajectories, generated by the visually-guided feeder-taxic behavior, each of which contains only part of the desired whole. The experimental procedure was identical to that in Experiment 4, with the exception that the feeder sequences that were provided to the visually-guided feeder-taxic navigation system were ABC, BCD and CDE illustrated in Figure 11A-C.

*** Insert Figure 11 ***

Figure 11D illustrates the superposition of 10 assessment trials after training on snippets generated from visually-guided feeder-taxic sequences. These assessment trials demonstrate that the robot can use trajectories generated by visual navigation to train the HIPP-PFC model which can integrate the efficient subsequences and generate the complete efficient trajectory.

## 4. Discussion:

In this study, experiments were performed with a physical robot to test a novel hippocampus-reservoir model (Cazin et al. 2019) under real-time constraints similar to that of rats with 5 baited cups. By real-time we mean that the model generates its response, the robot physically moves, the location input is updated to provide a new input to the model, and the model generates it next response. During Phase 1 first trials, the robot follows pre-recorded paths. We can consider that this corresponds to the rat using simple local heuristics for foraging During the traversal, place cells are activated, and when a reward is detected (provided by the control program that will give a reward signal when the robot approaches a baited target), this place and reward information is input to the replay model that propagates reward value along the spatial trajectories, implementing a reinforcement learning process. After the robot has visited all of the rewarded targets, it enters the inter-trial period. During this period the hippocampus model will predominantly replay snippets that were along the part of the trajectories that received the propagated reward. This replay is provided as a place cell activation sequence to the PFC model. With data from multiple trials, via the reward propagation in hippocampus model, the replay of snippets that occur during an efficient traversal from one rewarded target to another will be increased with respect to traversals that strayed far from rewards. The inputs to the PFC model thus predominantly consist of snippets that form part of the global efficient sequence to be synthesized from past experience. We have shown that, whereas the hippocampus model can learn local regularities (in terms of creating snippets with replay modulated by reward), it does not bind them into a coherent whole. The PFC model performs this global linkage of replayed snippets to produce a complete beginning to end sequence.

An interesting feature of this replay is the time warping. While rats tend to navigate at speeds on the order of 0.5m/s, their replay rates are on the order of 15-20 times faster (7.6 - 10.8 m/s). This requires that the PFC can receive training input from hippocampus at the compresses or speeded rate, and then during future navigation, generate its outputs 15-20 times slower. Interestingly, such effects can be achieved by changing the time constant of the leaky integrator neurons (Jaeger et al 2007), suggesting that the PFC model can learn at one rate and replay 15-20 times slower, as required in the animal.

Perhaps most interestingly, this research serves as an example of the value of physical embodiment. In Cazin et al (2019), we noted that when reservoir output (in a place cell population code) is fed back as input during autonomous sequence generation, small errors can accumulate and lead to divergence from the desired trajectory. To attenuate this problem we introduced denoising procedure as a spatial filter, which decoded the space cell output into the corresponding location and then properly recoded that location which was then reinjected into the reservoir in the autonomous generation procedure. Interestingly, what we observe with the robot is that the physical embodiment actually contributes to this denoising procedure. That is, the system takes the place-cell coded output and uses this to generate the command to move to the coded location. The place-cell code for this location is then generated and used as input to the reservoir. Even if we use the overhead camera to determine the location of the robot to generate the new place cell code, what is important is that the noisy location code in PFC is denoised by generating a noise free physical movement by the robot The embodiment thus contributes to the denoising, as it is the physical movement of the robot to a particular location that is used to generate the proper coding of that location as input in the next processing cycle.

The current research focuses on awake replay that occurs during rest between successive trials. During learning, the robot follows a trajectory not generated by the model. This could be pre-recorded (as in Experiments 1 and 2), or could come from a simpler foraging system (as in Experiments 3-5). After one or more such trials, during a presumed short rest period, replay propagates the rewards, and generates snippet based on these propagated rewards. These snippets are used to train the model, and in the following trail the robot uses the newly synthesized TSP sequence. The system thus exploits reverse replay for formation of the reward model, and forward replay for training the reservoir. In previous work (Cazin et al 2019), we also examined how reverse replay during training of the reservoir could allow the

system to exploit its past experience but in the reverse direction as discussed in (Gupta et al 2010).

In the real TSP setting, after each attempt in a given configuration of the arena, the rat is returned to a waiting area where replay can occur. Awake replay can typically occur during such rest periods between trials. Interestingly, awake reply predicts upcoming navigation behavior and tends to over-represent the goal location (Pfeiffer & Foster 2013), consistent with our model. We approximate this by modeling replay as occurring during the rest period after several runs through the baited arena, with replay biased towards locations near the rewards. In contrast, during a given run, while the rat is consuming a reward, reverse replay occurs, starting from the rewarded location (Carr et al 2011). This is related to our reverse replay that is used to propagate the reward. Interestingly, allowing reverse replay for this reward propagation to take place during the pause at each rewarded site would provide for a more realistic model.

The use of replay in reinforcement learning (RL) (Cazé et al 2018) and more generally in machine learning (Andrychowicz et al 2017) indicates its power as providing a method to generate training data and internal models to improve learning. Different forms of replay can be used in to learn the value function in model free RL, and in learning the world model in model based RL. Our model uses reward propagation to form a reward gradient that is used to generate data for training a reservoir sequence learning model. We evaluate the model using the TSP problem which is rather unique in the RL sequence learning literature. We provide one exposure to each of a small number of sequences (here 3), each of which contains a subsequence that is part of the optimal path. The system then learns based on a single exposure to each of these three sequences, and is then allowed a single trial to determine if through learning the system can extract the efficient subsequences to generate the optimal sequence. Typical RL systems tested in rat navigation contexts learn by exploration and require 20-50 laps through the maze in order to learn (Cazé et al 2018). Our model is able to learn with one trial of each of the training sequences by an innovative use of replay first to perform reward propagation, and then to train the model on data that is biased by this propagated reward. This reward propagation to bias replay for learning is a computationally simple form of reinforcement learning for reservoir computing that favors the efficient subsequences resulting in synthesis of the novel efficient sequence.

One of the limitations of our work is that robot position was provided by an external top-down viewing camera system. More realistic localization in navigation has been provided in related work by (Tejera et al 2018), building on work initially demonstrated in place cell models for robot navigation (Burgess et al 1997). Future research includes further understanding of how different navigation and learning mechanisms interact. The system we investigated in our work requires previous experience from which to extract the most efficient components. Thus, other, likely simpler, mechanisms could be at work to plan more local navigation strategies, while more integrative strategies could be provided by the type of mechanisms suggested in this work. Importantly, we demonstrate here how the biasing of snippet replay by reward proximity allows a reservoir model of PFC to generate efficient navigation sequences, demonstrated in physical robot implementation.

# Figures

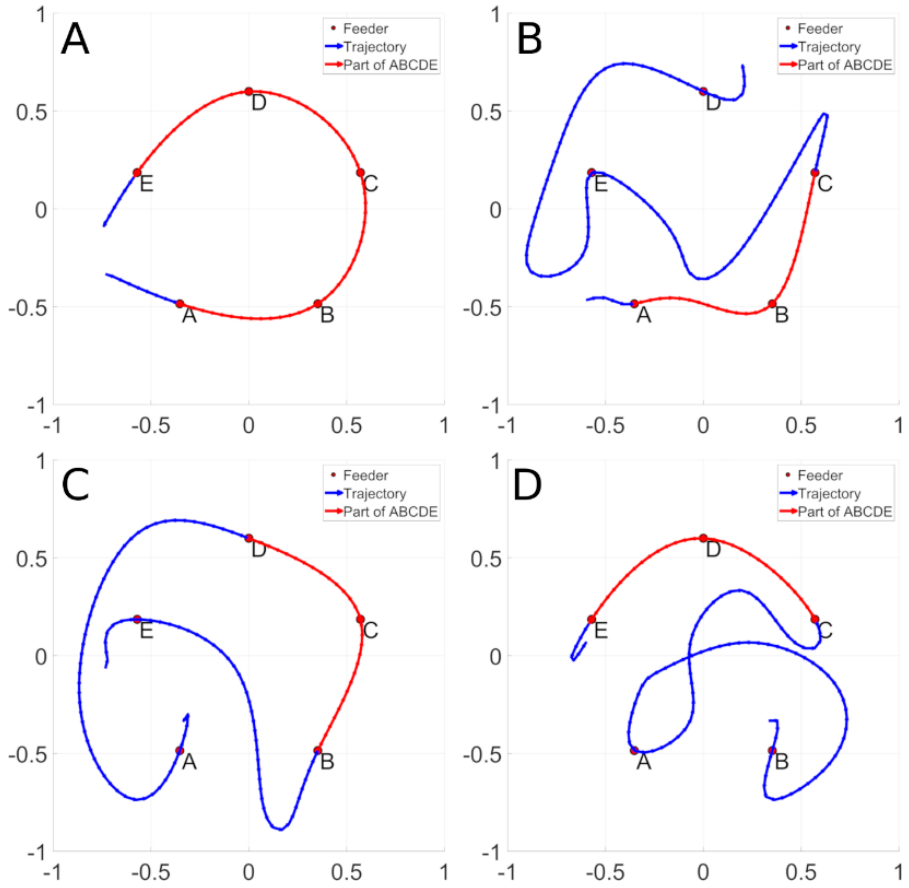


Figure 1. Schematic of the TSP task. A. The desired efficient trajectory. B-D. Three inefficient trajectories that contain parts of the efficient trajectory (in red), and inefficient components (in blue). After exposure to B-D, the rat or model should be able to generate A. (From Cazin et al. 2019 with permission).

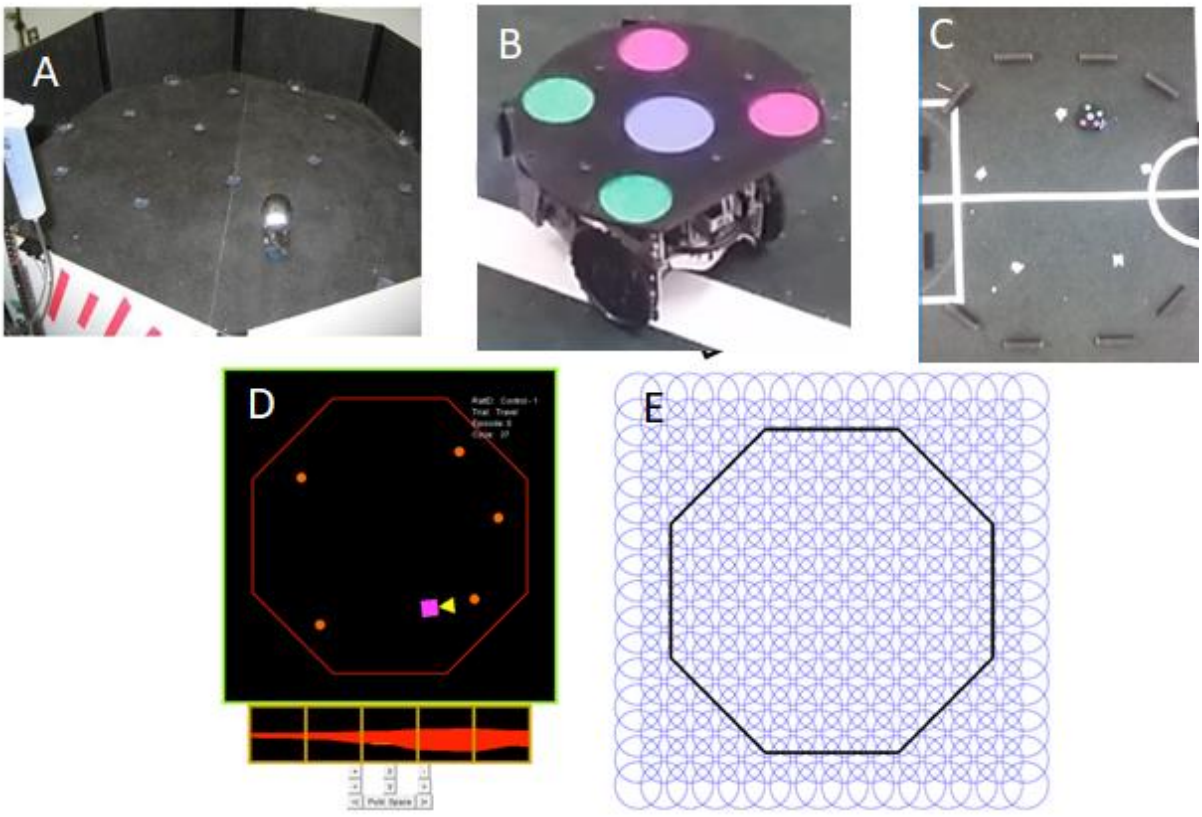


Figure 2: Rodent and robot experiments. A. Behavioral apparatus where a rat (no hyperdrive) is shown wearing a reflective jacket for tracking purposes on the open-field with 21 cups (Watkins de Jong et al. 2011). B. Physical robot that is used as a rat robot. C. Deployment of the rat robot in the baited arena. D. Illustration of SCS robot simulator in an evirnment of baited target positions. E. 16x16 place cell tessellation of the physical space of the SCS simulator and robot.

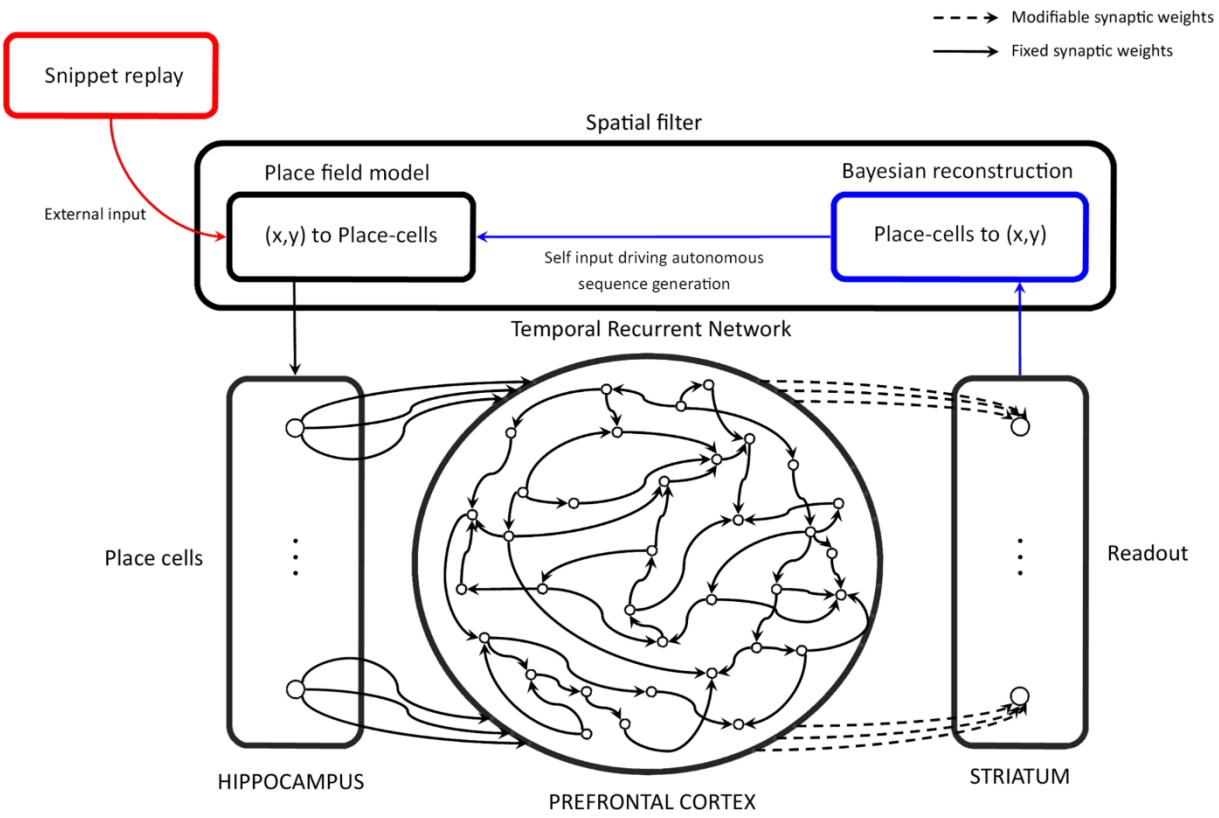


Figure 3. Hippocampus-PFC model architecture. Prefrontal cortex is modeled as a 1000 unit leaky integrator reservoir. Input is provided as the current location from Hippocampus in a 256 element place cell activation vector. Likewise, modifiable readout connections project to a place cell code of the next location coded in the readout (striatum). During sequence generation, this place cell output is recoded into a de-noised place cell code in the Spatial Filter (equivalent to driving a simulated or real robot), with the resulting next step provided as input to the PFC-reservoir, in an auto-generation mode. See text for details on training. From Cazin et al (2019) with permission.

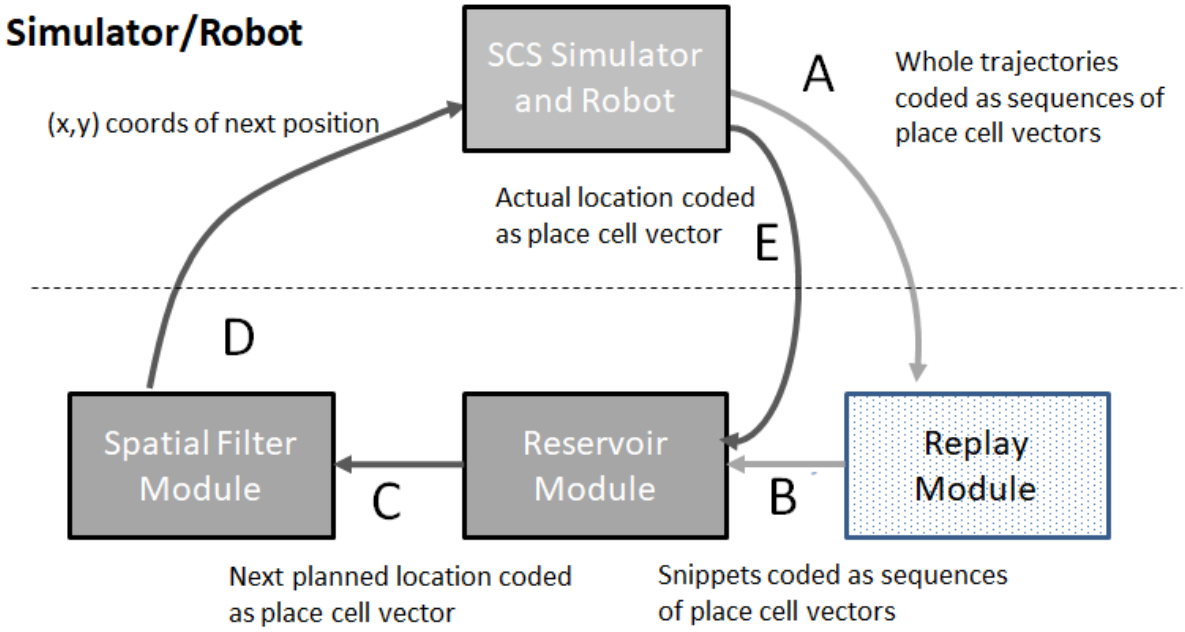


Figure 4. Integration of SCS Simulator and Robot with the Reservoir sequence learning model. A. After the SCS generates one or more navigation trajectories to train the Reservoir, these trajectories are sent to the Replay Module. The Replay Module learns the reward propagation model, implementing a spatial reward gradient. B. Snippets generated by replay using the reward gradient are used to train the Reservoir model. The trained model is then executed. C. At each time step it sends the next predicted location to the Spatial Filter as a place cell activation vector. D. The Spatial Filter decodes the x,y coordinates of that place cell population and transmits these coordinates to the SCS Simulator or Robot. E. The Simulator/Robot generates the corresponding motion to that new position, and transmits the location coded as a place cell activation vector to the Reservoir. This completes the sensorimotor loop, as this new position input drives the Reservoir to generate the prediction of the next movement in the sequence.

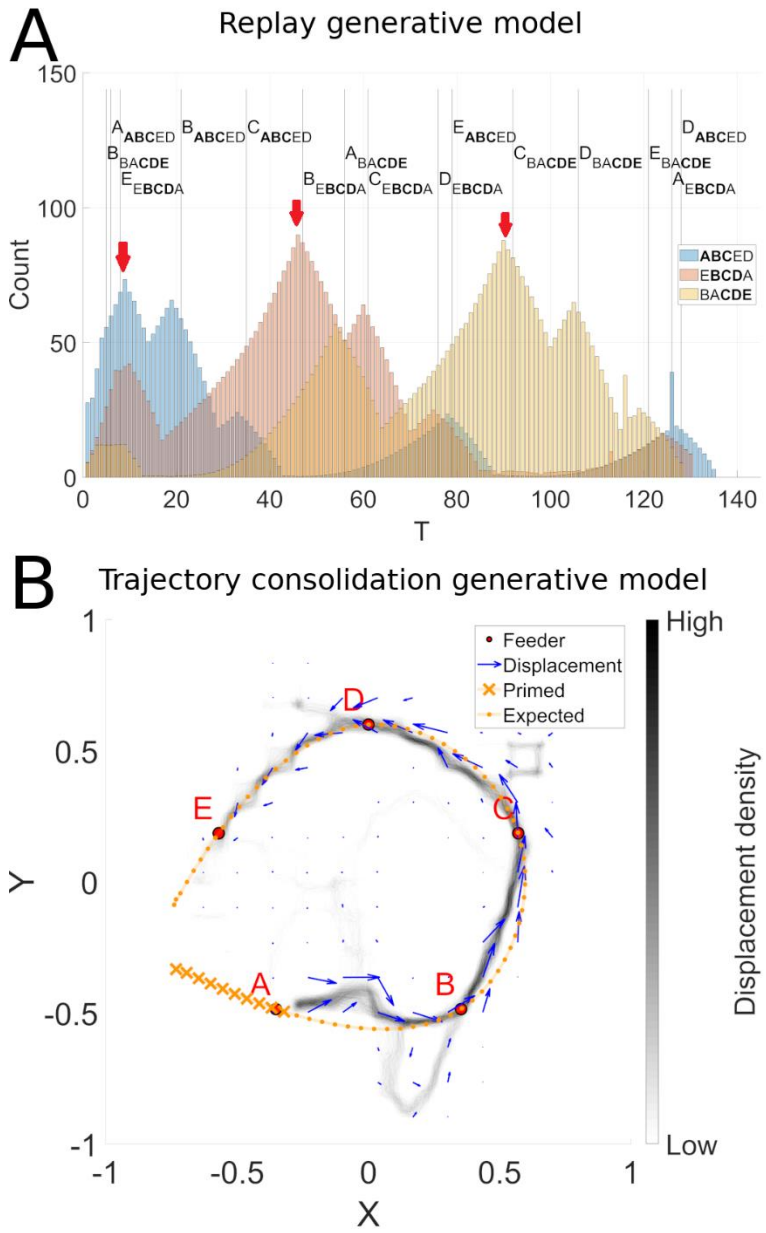


Figure 5. Efficient Sequence Synthesis using the SCS robot simulator. A. Distribution of snippets drawn from the sequences illustrated in Figure 1 B, C and D. Red arrows indicate the peak replay probabilities corresponding to the subsequences ABC, BCD and CDE that constitute the efficient sequences. Globally we observe snippet selection favors snippets from the beginning of sequence ABCED (blue), the middle of EBCDA (yellow), and the end of sequence BACDE (pink), which corresponds exactly to the efficient subsequences (ABC, BDC, and CDE) of these three sequences. This distribution of snippets is used to train the model. The results of the training are illustrated in panel B. Here we see a 2D histogram of sequences generated by the model in the ABCDE recombination experiment. Scs simulator is used instead of a virtual rat plugin, 1000 trajectories (1000 different rats, 1 random walk per rat) are used.

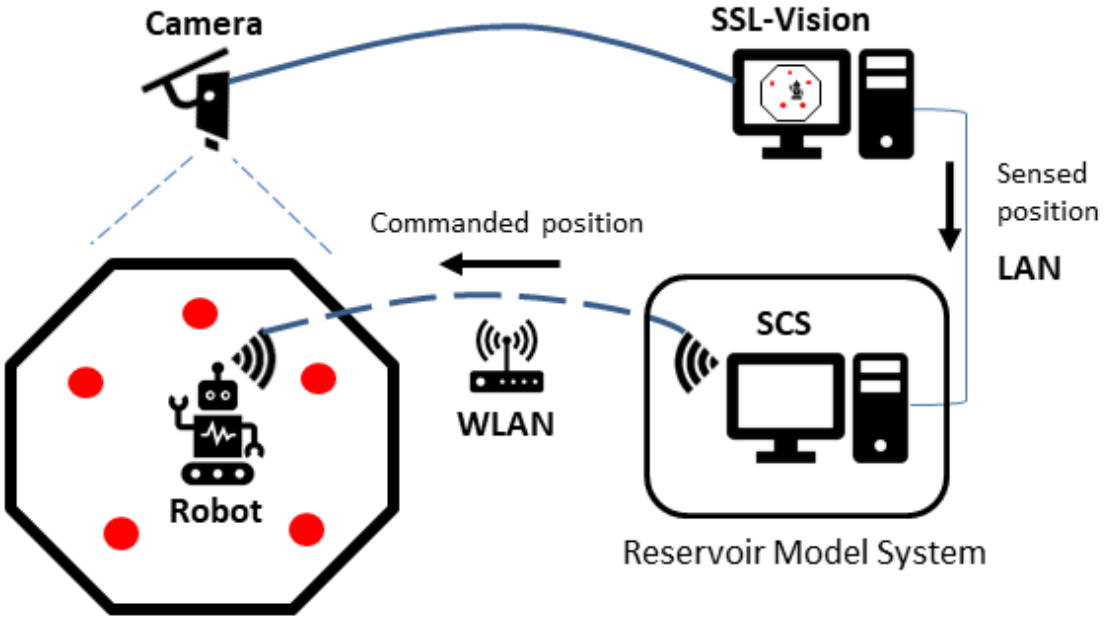


Figure 6. Communication infrastructure for integrated robot experiments. Robot location perceived by Camera connected to computer that executes SSL-Vision software to track the position of the robot. The position is made available to the computer running SCS by means of multicast messages sent over LAN. SCS runs the reservoir model and sends commands to the robot via WIFI using ROS (Robot Operating System).

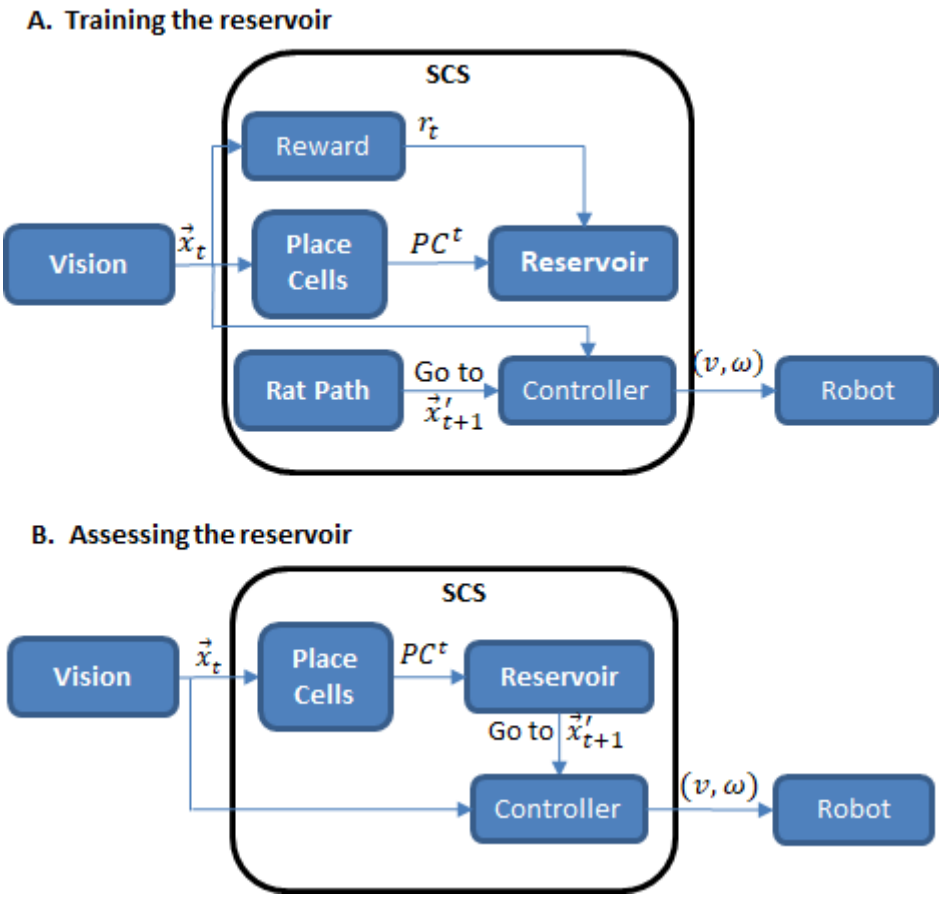


Figure 7. Summary of interaction between robot and reservoir for training and assessment. A. During the execution of pre-recorded trajectories, Rat Path sends locations to the controller, and in parallel, the sensory feedback from vision allows creation of the place cell trajectories that correspond to the physical path taken by the robot. This is to be used for generating the reward-modulated replay model that generates snippets to train the reservoir. B. During assessment, the trained reservoir is inserted in the sensory-motor loop. Vison generates place cell codes that drive the reservoir. The reservoir generates output as place cell activity that is used to generate the command to the robot controller. The robot moves, updates its position and the vision system perceives this new position and the loop continues.

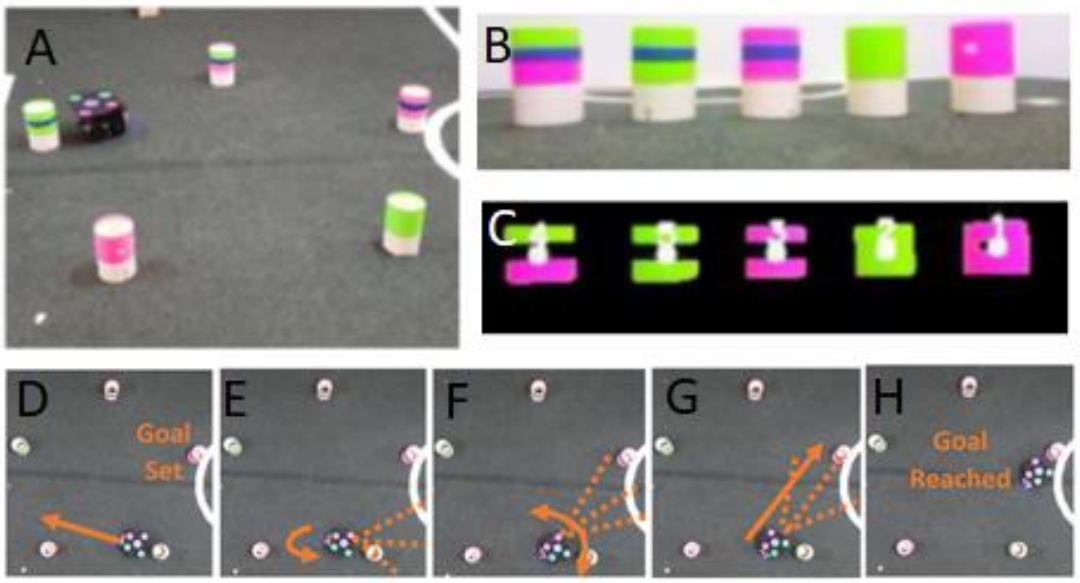


Figure 8. Visually guided navigation. Feeder localization is performed by finding the centroids of the color blobs using OpenCV A) Disposition of the feeders in space. B) Feeder color details. C) CVS Output detection. D) New goal is set. Robot moves backwards to allow turning in place. E) Goal not in sight, turn in place until goal feeder is found. F) Goal was found, center it on image using proportional control. G)Centered on goal, advance while keeping goal centered until close enough. Use proportional control to control both linear and angular speed. H) Goal reached. Robot stops and waits new feeder to be chosen.

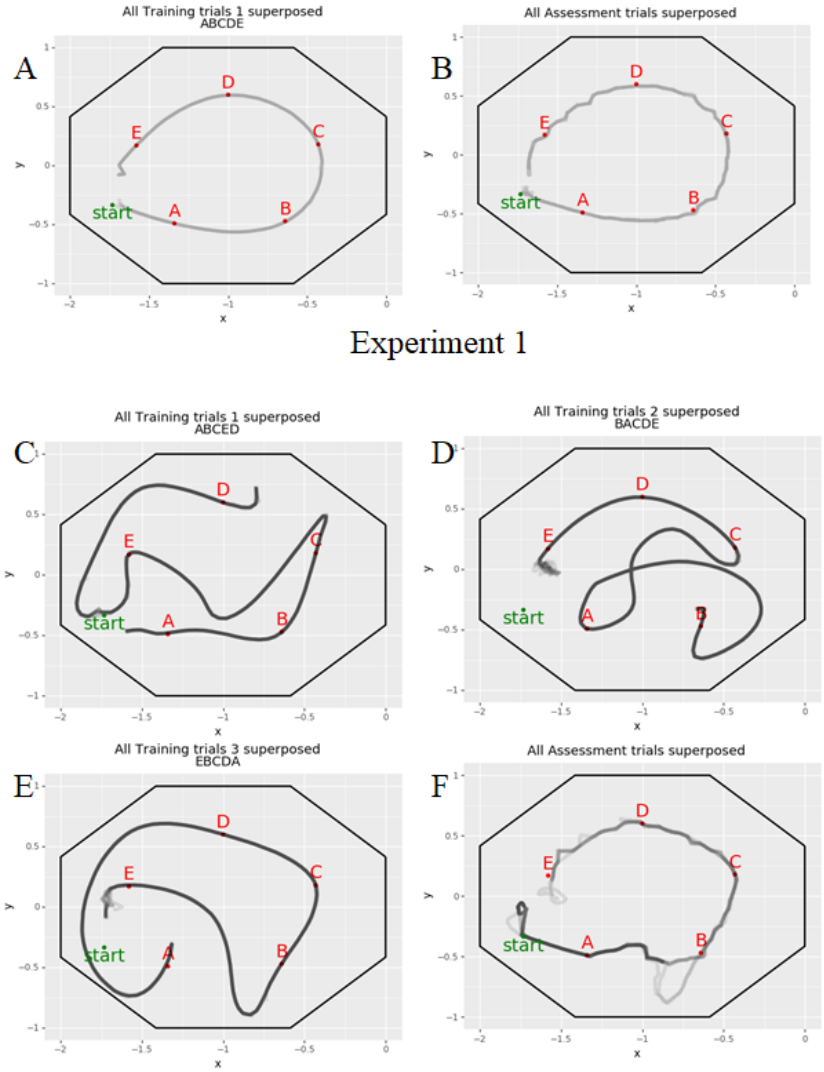


Figure 9. Robot execution results with training trajectories given to robot. Experiment 1. Panel A indicates the superposition of the 10 robot executions of the pre-recorded trajectory. This was used to generate a place cell trajectory that was used by the replay model to train the reservoir. Panel B illustrates the trajectory superposition of the trajectory that was learned by the reservoir and executed in a real-time sensorimotor loop with the robot over 10 separate execution. Experiment 2. Panels C-E represent the superposition of the 10 robot executions for each of the three prerecorded inefficient sequences linking rewarded targets A,B,C,D and E. Panel F illustrates the superposition of 10 executions of the robot when the integration of training trials in panels C-E is used to train the reservoir. The result is the discovery of the efficient global sequence, which was never seen in its entirety during training. The superposition in Panel F illustrates some variability in this integrative processing.

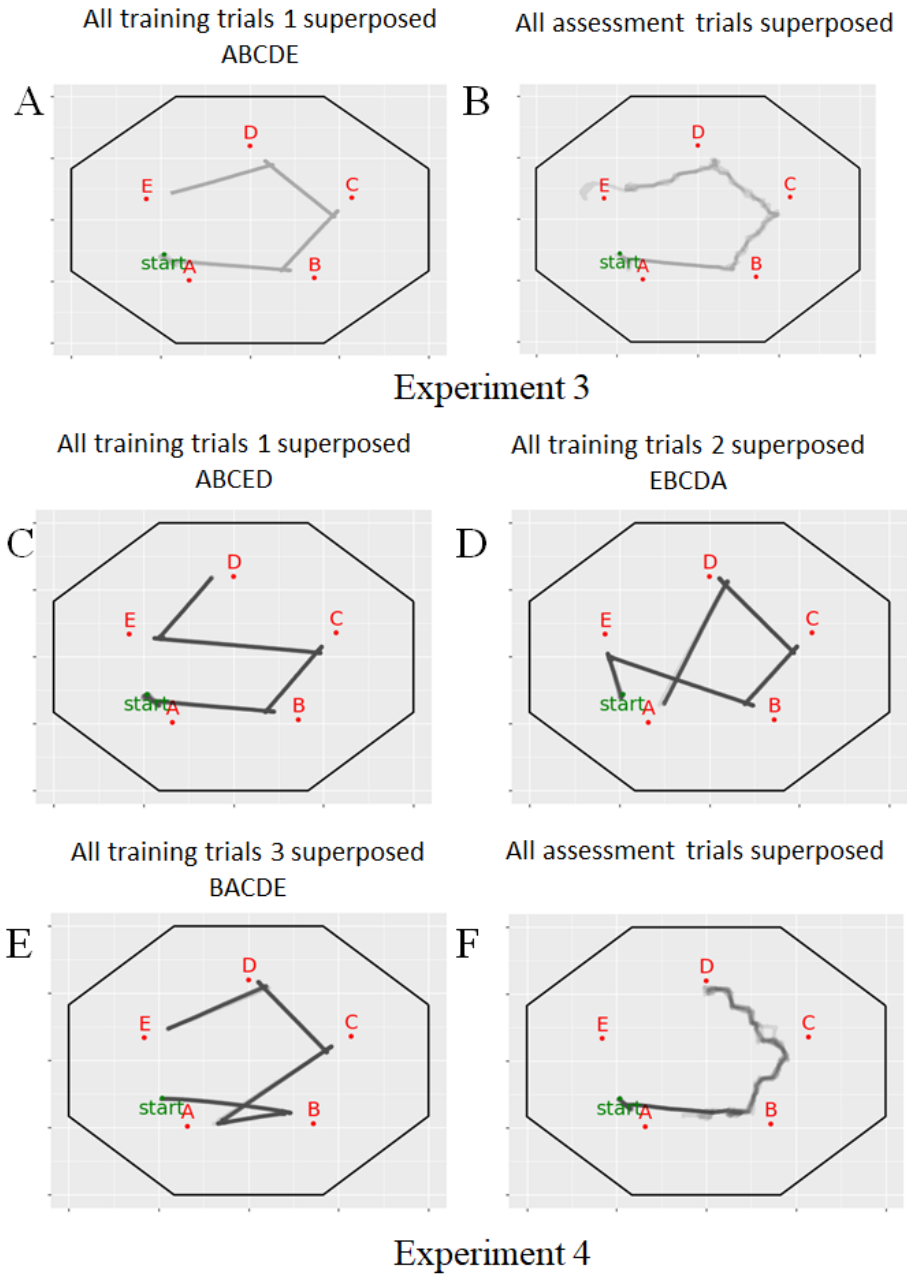


Figure 10. Robot execution results with training trajectories given provided by visual-taxic navigation. Experiment 3. Panel A indicates the superposition of the 10 robot executions of the visually guided trajectory. This was used to generate a place cell trajectory that was used by the replay model to train the reservoir. Panel B illustrates the trajectory superposition of the trajectory that was learned by the reservoir and executed in a real-time sensorimotor loop with the robot over 10 separate execution. Experiment 4. Panels C-E represent the superposition of the 10 robot executions for each of the three visually guided partial sequences linking rewarded targets A,B,C,D and E. Panel F illustrates the superposition of 10 executions of the robot when the integration of training trials in panels C-E is used to train the reservoir. The result is the discovery of the efficient global sequence, which was never seen in its entirety during training. The superposition in Panel F illustrates some variability in this integrative processing, and the inability to finish the sequence trajectory from feeder D to E.

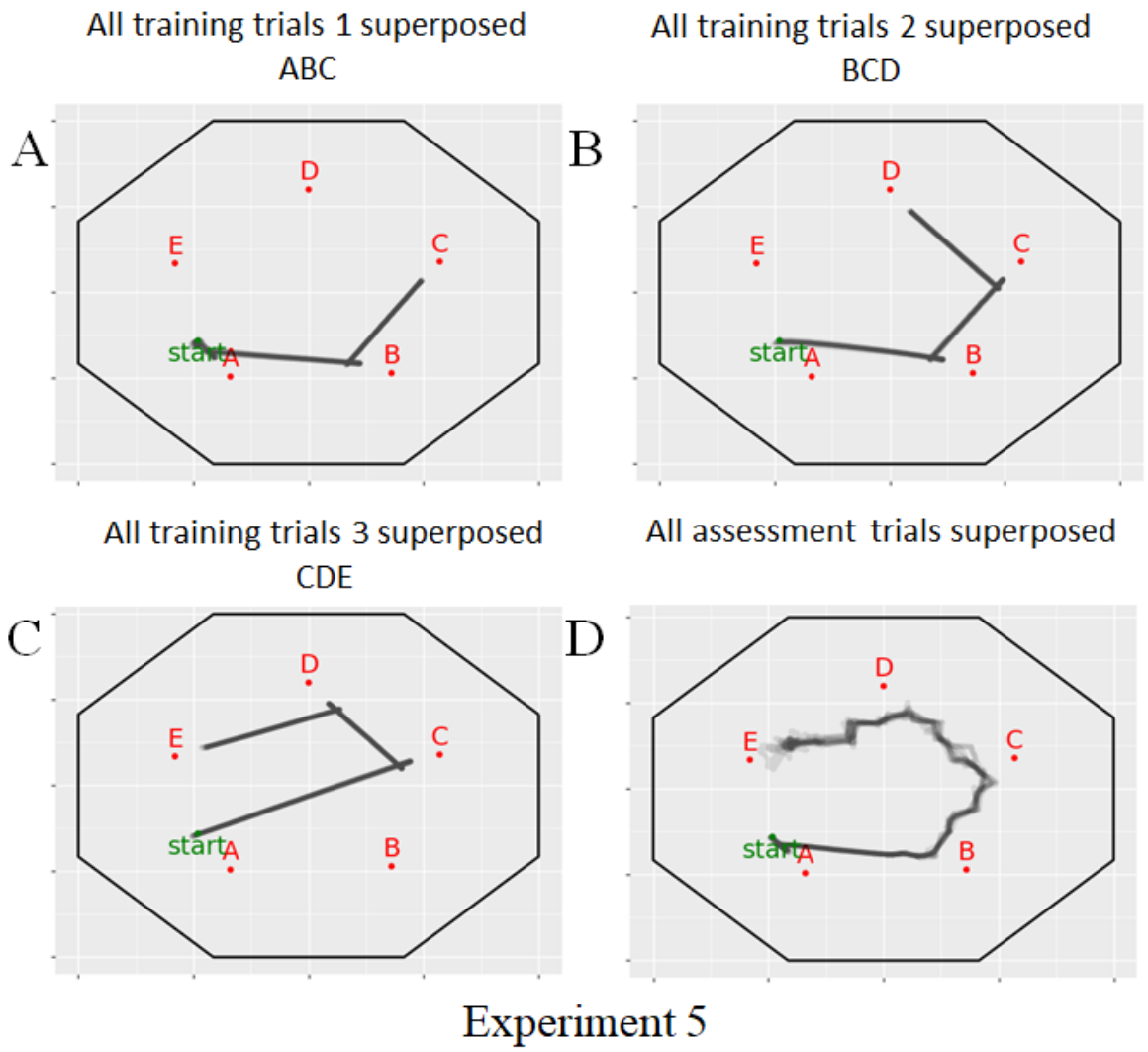


Figure 11. Experiment 4. Robot execution results with training trajectories given provided by visual-taxic navigation. Panels A-C represent the superposition of the 10 robot executions for each of the three visually guided partial sequences linking rewarded targets A,B,C,D and E. Panel D illustrates the superposition of 10 executions of the robot when the integration of training trials in panels A-C is used to train the reservoir. The result is the discovery of the efficient global sequence, which was never seen in its entirety during training. The superposition in Panel D illustrates some variability in this successful integrative processing.

Table 1. Parameters used for robot experiments. Leak rate h was established empirically.

| Replay | Reservoir |
|---|---|
| $\beta_{learn}$ = 1 learning reverse replay rate | N = 1024 Reservoir size |
| $\beta_{generate}$ = 0 generate reverse replay rate | α = 9.76e-06 learning rate |
| $T$ = 1000 replay time budget | γ = 0.95 discount constant |
| Snippet size = 10 | $h$ = 0.785991 leak rate |
| K = 256 number of place cells | Mini-batch $b$ = 32 |
| | Sampling rate – integration time step = 0.05s |

Appendix: Code for the HIP-PFC model simulator is available at : https://github.com/NicolasCAZIN/TRN. Code for the animat SCS simulator integrated with the model is available at : https://github.com/biorobaw/scs

References:


Ambrose RE, Pfeiffer BE, Foster DJ. 2016. Reverse replay of hippocampal place cells is uniquely modulated by changing reward. *Neuron* 91: 1124-36

Andrychowicz M, Wolski F, Ray A, Schneider J, Fong R, et al. *Advances in neural information processing systems2017*: 5048-58.

Arleo A, Gerstner W. 2000. Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity. *Biological cybernetics* 83: 287-99

Barrera A, Cáceres A, Weitzenfeld A, Ramirez-Amaya V. 2011. Comparative experimental studies on spatial memory and learning in rats and robots. *Journal of Intelligent & Robotic Systems* 63: 361-97

Barrera A, Tejera G, Llofriu M, Weitzenfeld A. 2015. Learning Spatial Localization: From Rat Studies to Computational Models of the Hippocampus. *Spatial Cognition \& Computation* 15: 27-59

Barrera A, Weitzenfeld A. 2008. Biologically-inspired robot spatial cognition based on rat neurophysiological studies. *Autonomous Robots* 25: 147-69

Bendor D, Wilson MA. 2012. Biasing the content of hippocampal replay during sleep. *Nat Neurosci* 15: 1439-44

Brown MA, Sharp PE. 1995. Simulation of spatial learning in the Morris water maze by a neural network model of the hippocampal formation and nucleus accumbens. *Hippocampus* 5: 171-88

Burgess N, Donnett JG, Jeffery KJ, O–keefe J. 1997. Robotic and neuronal simulation of the hippocampus and rat navigation. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 352: 1535-43

Burgess N, Recce M, O'Keefe J. 1994. A model of hippocampal function. *Neural networks* 7: 1065-81

Buzsáki G. 1989. Two-stage model of memory trace formation: a role for "noisy" brain states. *Neuroscience* 31: 551-70

Caluwaerts K, Staffa M, N'Guyen S, Grand C, Dollé L, et al. 2012. A biologically inspired meta-control navigation system for the psikharpax rat robot. *Bioinspiration & biomimetics* 7: 025009

Carr MF, Jadhav SP, Frank LM. 2011. Hippocampal replay in the awake state: a potential substrate for memory consolidation and retrieval. *Nature neuroscience* 14: 147-53

Cazé R, Khamassi M, Aubin L, Girard B. 2018. Hippocampal replays under the scrutiny of reinforcement learning models. *Journal of neurophysiology* 120: 2877-96

Cazin N, Llofriu Alonso M, Scleidorovich Chiodi P, Pelc T, Harland B, et al. 2019. Reservoir computing model of prefrontal cortex creates novel combinations of previous navigation sequences from hippocampal place-cell replay with spatial reward propagation. *PLoS computational biology* 15: e1006624

Davidson TJ, Kloosterman F, Wilson MA. 2009. Hippocampal replay of extended experience. *Neuron* 63: 497-507

de Jong LW, Gereke B, Martin GM, Fellous J-M. 2011. The traveling salesrat: insights into the dynamics of efficient spatial navigation in the rodent. *Journal of Neural Engineering* 8: 065010

De Lavilléon G, Lacroix MM, Rondi-Reig L, Benchenane K. 2015. Explicit memory creation during sleep demonstrates a causal role of place cells in navigation. *Nature neuroscience* 18: 493

Diba K, Buzsaki G. 2007. Forward and reverse hippocampal place-cell sequences during ripples. *Nature neuroscience* 10: 1241

Dollé L, Sheynikhovich D, Girard B, Chavarriaga R, Guillot A. 2010. Path planning versus cue responding: a bio-inspired model of switching between navigation strategies. *Biological cybernetics* 103: 299-317

Dominey PF. 1995. Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning. *Biol Cybern* 73: 265-74

Dominey PF. 1998a. Influences of temporal organization on sequence learning and transfer: Comments on Stadler (1995) and Curran and Keele (1993). *Journal of Experimental Psychology: Learning, Memory, and Cognition,* 24: 14

Dominey PF. 1998b. A shared system for learning serial and temporal structure of sensori-motor sequences? Evidence from simulation and human experiments. *Brain Res Cogn Brain Res* 6: 163-72

Dominey PF, Arbib MA, Joseph JP. 1995. A Model of Corticostriatal Plasticity for Learning Oculomotor Associations and Sequences *J Cogn Neurosci* 7: 25

Dominey PF, Inui T, Hoen M. 2009. Neural network processing of natural language: II. Towards a unified model of corticostriatal function in learning sentence comprehension and non-linguistic sequencing. *Brain & Language* 109: 80-92

Dominey PF, Ramus F. 2000. Neural network processing of natural language: I. Sensitivity to serial, temporal and abstract structure of language in the infant. *Language and Cognitive Processes* 15: 40

Euston DR, Tatsuno M, McNaughton BL. 2007. Fast-forward playback of recent memory sequences in prefrontal cortex during sleep. *Science* 318: 1147-50

Foster DJ, Wilson MA. 2006. Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature* 440: 680-83

Gaussier P, Banquet J, Sargolini F, Giovannangeli C, Save E, Poucet B. 2007. A model of grid cells involving extra hippocampal path integration, and the hippocampal loop. *Journal of integrative neuroscience* 6: 447-76

Gaussier P, Revel A, Banquet J-P, Babeau V. 2002. From view cells and place cells to cognitive map learning: processing stages of the hippocampal system. *Biological cybernetics* 86: 15-28

Guazzelli A, Bota M, Corbacho FJ, Arbib MA. 1998. Affordances. motivations, and the world graph theory. *Adaptive Behavior* 6: 435-71

1036 Gupta AS, van der Meer MA, Touretzky DS, Redish AD. 2010. Hippocampal replay is not a simple
1037 function of experience. *Neuron* 65: 695-705
1038 Hasselmo ME. 2008. Temporally structured replay of neural activity in a model of entorhinal cortex,
1039 hippocampus and postsubiculum. *Eur J Neurosci* 28: 1301-15
1040 Hinaut X, Dominey PF. 2013. Real-time parallel processing of grammatical structure in the fronto-
1041 striatal system: a recurrent network simulation study using reservoir computing. *PloS one* 8:
1042 1-18
1043 Hoffman KL, McNaughton BL. 2002. Coordinated reactivation of distributed memory traces in
1044 primate neocortex. *Science* 297: 2070-3
1045 Jaeger H, Haas H. 2004. Harnessing nonlinearity: predicting chaotic systems and saving energy in
1046 wireless communication. *Science* 304: 78-80
1047 Jaeger H, Lukosevicius M, Popovici D, Siewert U. 2007. Optimization and applications of echo state
1048 networks with leaky-integrator neurons. *Neural Netw* 20: 335-52
1049 Ji D, Wilson MA. 2007. Coordinated memory replay in the visual cortex and hippocampus during
1050 sleep. *Nat Neurosci* 10: 100-7
1051 Johnson A, Redish AD. 2005. Hippocampal replay contributes to within session learning in a temporal
1052 difference reinforcement learning model. *Neural Netw* 18: 1163-71
1053 Lansink CS, Goltstein PM, Lankelma JV, Joosten RN, McNaughton BL, Pennartz CM. 2008. Preferential
1054 reactivation of motivationally relevant information in the ventral striatum. *J Neurosci* 28:
1055 6372-82
1056 Llofriu M, Tejera G, Contreras M, Pelc T, Fellous J-M, Weitzenfeld A. 2015. Goal-oriented robot
1057 navigation learning using a multi-scale space representation. *Neural Networks* 72: 62-74
1058 Lukosevicius M. 2012. A practical guide to applying echo state networks In *Neural networks: tricks of*
1059 *the trade*, pp. 659-86: Springer
1060 Lukosevicius M, Jaeger H. 2009. Reservoir computing approaches to recurrent neural network
1061 training. *Computer Science Review* 3: 22
1062 Maass W, Natschlager T, Markram H. 2002. Real-time computing without stable states: a new
1063 framework for neural computation based on perturbations. *Neural Comput* 14: 2531-60
1064 McClelland JL, McNaughton BL, O'Reilly RC. 1995. Why there are complementary learning systems in
1065 the hippocampus and neocortex: insights from the successes and failures of connectionist
1066 models of learning and memory. *Psychol Rev* 102: 419-57
1067 Moser EI, Moser M-B, McNaughton BL. 2017. Spatial representation in the hippocampal formation: a
1068 history. *Nature neuroscience* 20: 1448
1069 Nadel L, Moscovitch M. 1997. Memory consolidation, retrograde amnesia and the hippocampal
1070 complex. *Current opinion in neurobiology* 7: 217-27
1071 Nikolic D, Hausler S, Singer W, Maass W. 2009. Distributed fading memory for stimulus properties in
1072 the primary visual cortex. *PLoS Biol* 7: e1000260
1073 Peyrache A, Khamassi M, Benchenane K, Wiener SI, Battaglia FP. 2009. Replay of rule-learning related
1074 neural patterns in the prefrontal cortex during sleep. *Nat Neurosci* 12: 919-26
1075 Pfeifer R, Lungarella M, Iida F. 2007. Self-organization, embodiment, and biologically inspired
1076 robotics. *science* 318: 1088-93
1077 Pfeiffer BE, Foster DJ. 2013. Hippocampal place-cell sequences depict future paths to remembered
1078 goals. *Nature* 497: 74
1079 Redish AD, Touretzky DS. 1997. Cognitive maps beyond the hippocampus. *Hippocampus* 7: 15-35
1080 Rigotti M, Barak O, Warden MR, Wang X-J, Daw ND, et al. 2013. The importance of mixed selectivity
1081 in complex cognitive tasks. *Nature*
1082 Ruder S. 2016. An overview of gradient descent optimization algorithms. *arXiv preprint*
1083 *arXiv:1609.04747*
1084 Singer AC, Frank LM. 2009. Rewarded Outcomes Enhance Reactivation of Experience in the
1085 Hippocampus. *Neuron* 64: 910-21

Tejera G, Llofriu M, Barrera A, Weitzenfeld A. 2018. Bio-Inspired Robotics: A Spatial Cognition Model integrating Place Cells, Grid Cells and Head Direction Cells. *Journal of Intelligent & Robotic Systems* 91: 85-99

Widrow B, Hoff ME. 1960. Adaptive switching circuits, STANFORD UNIV CA STANFORD ELECTRONICS LABS

Wilson MA, McNaughton BL. 1994. Reactivation of hippocampal ensemble memories during sleep. *science* 265: 676-79

Wylie TR. 2013. *The discrete Fréchet distance with applications*. Montana State University-Bozeman, College of Engineering